Efficient Content Sharing over Content Centric Networking

Sho Harada
Waseda University
Tokyo, Japan
shoharada1990@akane.waseda.jp

Yong-Jin Park
Waseda University
Tokyo, Japan
yjp@ieee.org

***Abstract*—Content Centric Networking (CCN) is a clean slate network architecture optimized for today’s and expected future’s demands for the Internet. It is a promising paradigm for the Future Internet architecture among new clean slate network architectures. It realizes the reduction of the load on servers, efficient mobility support, fast content distribution and retrieval, and high security. However, there is room for improvement on CCN when we share contents in a community or we distribute contents as a service provider. In this paper, we propose a method that enables us to share contents more efficiently in a community. Our content sharing model will reduce an extra processing and the network load by 20% on average.**

Keywords—Future Internet; Content Centric Networking; Named Data Networking;

I. INTRODUCTION

In recent years, the number of Internet users is explosively increasing. In the past, we used Internet for exchanging messages. However, the purpose changed and the majority of today’s users use the Internet for retrieving contents. To meet today's demands, researchers have been extending the functionalities of Internet protocols. However, in this scenario, we need to continue extending the functionality if the scale of the Internet keeps increasing. One promising solution is to make a clean slate architecture that is designed for large-scale Internet and optimized for today's demands.

In the past five years, Information Centric Networking (ICN) [1] [2] is becoming popular. In ICN, we do not depend on location information such as IP address to request or retrieve contents, but Unified Resource Identifiers (URIs) that indicate resources. It enables us to request contents directly and we do not need to be aware of their exact locations. As ICN architectures, there are many promising architectures. PURSUIT [3], which is the succeeding project of PSIRP [4], has been researched mainly in Europe. It realizes the optimization of network by utilizing centralized management.

Content Centric Networking (CCN) [5] [6] is the most promising one in ICN architectures. CCN enables us to reduce the burden on servers, efficient mobility support, and fast content retrieval by dispersion of network load. However, there is room for improvement. Our proposal realizes cutting out the extra communication and reducing network load when we share contents in a dynamic community.

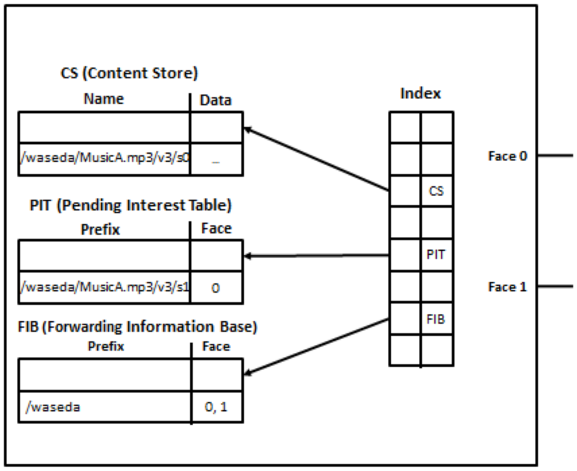


Figure 1. Structure of CCN Router

The remainder of this paper is organized as follows. Section Ⅱ describes the basic CCN architecture and Voice over Content Centric Networking [7] as a related work. In Section Ⅲ, we explain the problem of the basic CCN, our proposed community model, the key distribution method, and the content sharing model. In Section Ⅳ, we show a performance evaluation. Finally, we conclude our paper in Section Ⅴ.

II. CONTENT CENTRIC NETWORKING

A. Basic CCN Architecture

In CCN, we use two kinds of packets. Interest Packets are used to request content by name. Data Packets are used to deliver requested content to the requester in response to an Interest Packet. By using these packets, we can retrieve contents without knowing where they are. In CCN, we use an exclusive router called CCN Router. Fig. 1 shows the structure of CCN Router. CCN Router has three kinds of tables. Content Store (CS) is used to store contents, enables CCN Router to cache contents. When a CCN Router receives an Interest Packet and has the requested content in its CS, it will send the cached content to the requester instead of the content producer. Therefore, users can retrieve contents fast while the load on content producer is reduced. In addition, it supports mobility. When a user moves and fails to retrieve a Data Packet, the user just simply requests the content again.

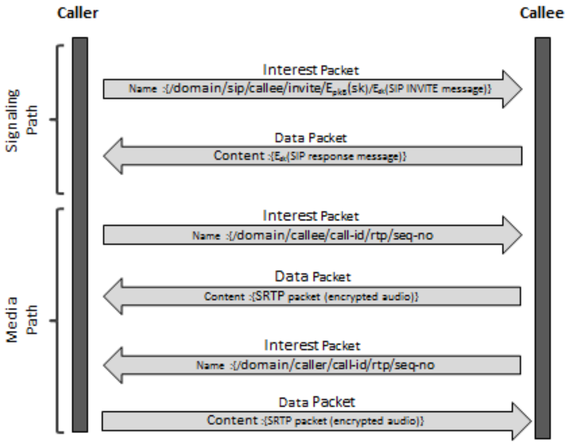

Caller
Callee
Signaling Path
Media Path
Interest Packet
Name :{/domain/sip/callee/invite/E_{pkB}(sk)/E_{sk}(SIP INVITE message)}
Data Packet
Content :{E_{sk}(SIP response message)}
Interest Packet
Name :{/domain/callee/call-id/rtp/seq-no
Data Packet
Content :{SRTP packet (encrypted audio)}
Interest Packet
Name :{/domain/caller/call-id/rtp/seq-no
Data Packet
Content :{SRTP packet (encrypted audio)}

Figure 2. Communication Model in VoCCN

In this case, the user can retrieve the content from the adjacent CCN Router, which has cached the content in its CS. Of course, we can use traditional technologies that are used in IP network for mobile support. Pending Interest Table (PIT) saves the face where an Interest Packet carried on and is used to forward Data Packet to the original requester. Forwarding Information Base (FIB) is similar to a routing table and used to forward Interest Packet to the content producer. The detail of FIB and the routing is written in [8].

In CCN, users can retrieve contents by name and do not care about the content locations. In addition, adequate security measure is taken in contents. Therefore, it is difficult for malicious users to attack a specific server nor to tamper contents.

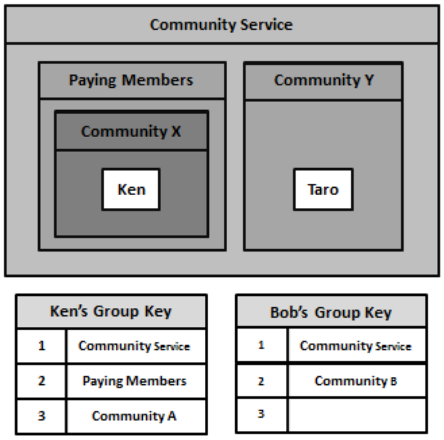
B. Related Work

Voice over Content Centric Networking (VoCCN) is a real-time, conversational, telephony application over CCN and simpler, more secure and more scalable than VoIP [9]. In VoCCN, data flows directly from producer to consumer. In addition, CCN architecture enables multipoint routing.

Fig. 2 shows the processes to start a call over CCN. At first, a caller sends an Interest Packet, which includes *"invite message"* and the symmetric key encrypted by the callee's public key combined with its content name. Then, the callee who received the caller's Interest Packet will send a response message encrypted by the caller's symmetric key as a Data Packet. By this method, the caller and the callee share a symmetric key and can communicate each other securely. In our proposal, secure key transmission is very important because our proposal is designed on the premise of it.

III. EFFICIENT CONTENT SHARING MODEL

In this Section, we explain the insufficient of basic CCN and our proposed content sharing model. Our content sharing model will reduce the load on servers. In addition, the average load on network will be cut by 20%.



Ken's Group Key	
1	Community Service
2	Paying Members
3	Community A

Bob's Group Key	
1	Community Service
2	Community B
3	

Figure 3. Proposed Community Model

A. The Problem of Basic CCN

In basic CCN, when a producer sends out content, the producer will encrypt the content with a key. Then, the requester will decrypt the content with a key that the requester received from the producer in advance. In this time, the encrypted content will be cached in every CCN Routers. Therefore, users can retrieve the content from these CCN Routers and decrypt it if they have the key. However, when the scope of disclosure is updated, all cached contents have to be deleted and the producer needs to re-encrypt content when they are requested again. It causes extra network load and processing since in a dynamic community or monthly subscription service, the scope of disclosure changes frequently. Therefore, we propose a method that enables CCN Routers to keep cached contents even when a community member list is changed and the scope of disclosure is updated. It will cut the extra network load and processing.

B. Community Model

We introduce a community model to realize our efficient content sharing model. Fig. 3 shows the structure of a community. A user belongs to a community that is a part of a big community. In a community, group members share a Group Key that is managed by the group leader or the service provider. When a user distributes contents, the user will encrypt the contents with a Group Key. Based on the scope of disclosure, the user can choose a Group Key. When a list of group members is updated, its Group Key will also be updated.

For example, when a producer wants to send out content only for the members of *"Community X"*, the producer will encrypt the content with the Group Key of *"Community X"*. However, when the producer sends out the content for all members of *"Community Service"*, the producer will encrypt the content with the Group Key of *"Community Service"*. Group Key is managed by the group leader or the service provider.

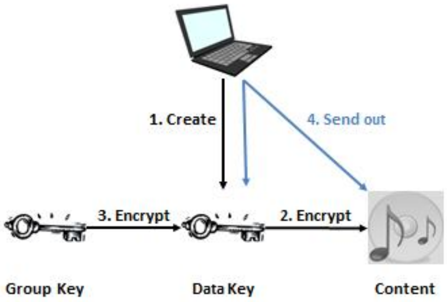


Figure 4. Content Encryption

In this case, the Group Keys of *"Community Service"* and *"Paying Members"* will be managed by the service provider. The Group Keys of *"Community X"* and *"Community Y"* will be managed by the group leaders.

This method enables us to control the scope of content disclosure by the community. In addition, we can control the scope of disclosure only by encrypting contents with key. This model is suitable for CCN architecture because authentication by server is not needed to retrieve contents.

C. Key Distribution

In our model, it is very important to exchange or distribute keys smoothly and securely. Therefore, we propose a key distribution method that applies, simplifies, and optimizes the key transmission of VoCCN for our proposed model.

To distribute a key, client sends Interest Packet by using the Public Key of Group Leader (PKG) and the Symmetric Key of the Client (SKC). The content name of Interest Packet will be like *"Community Service/Paying Members/ Community X/PKG(SKC)/SKC(authentication message and key request message"*. PKG(SKC) means that SKC is encrypted with PKG. In response to this Interest Packet, the group leader will send its group key encrypted with SKC as a Data Packet. Once the group leader needs any information about the client, the group leader will send Interest Packet encrypted by SKC to authenticate the client before sending Group Key.

D. Efficient Content Sharing Model

Fig. 4 shows our content encryption model. To distribute content, the producer creates a Data Key. The content name of a Data Key will be defined like *"Waseda/Network Community/MusicA.mp3/DataKey"*. After creating a Data Key, the producer will encrypt the content with this key.

The Data Key, which was used to encrypt the content, will also be encrypted with a Group Key. The producer will keep several Group Keys. Therefore, the producer will choose one of them to encrypt the content.

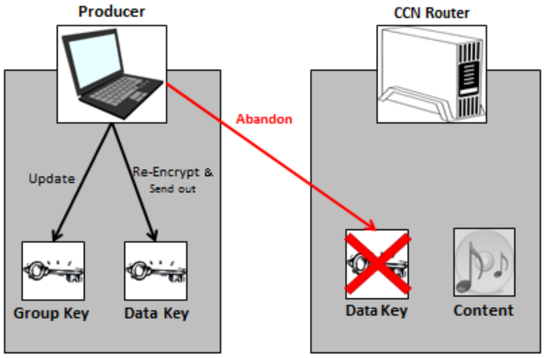


Figure 5. Updating the Scope

In our model, the content name of a Group Key will be defined like *"Waseda/NetworkCommunity/GroupKey/v0"*. Different from the name of Data Key, Group Key has the number of the version in its name. When a scope of disclosure is updated, the Group Key has to be updated. All users need to know it and distinguish the new Group Key. However, the old Data Key must be abandoned and replaced by the new Data Key. Therefore, the name of Data Key must be unique. When this content producer distributes the content, the content and the encrypted Data Key will be sent out at the same time.

In basic CCN, if the list of a community member is updated, the manager of the Group Key needs to update the Group Key and abandon the encrypted content. In our model, the manager will update the Group Key and abandon the encrypted Data Key instead of the content itself (Fig. 5).

Therefore, we do not have to update encrypted content and can continue to use cached contents. It will reduce the re-encryption cost and network load between the producer and the CCN router that caches the content because in many cases, the size of Data Key is much smaller than that of content itself.

IV. EVALUATION

Our proposal will reduce the network load. We defined *Efficiency* as the network load in our proposal divided by the network load in basic CCN. The network load between consumers and routers that cache contents is almost same. However, the load between a producer and the routers will be greatly reduced. The average *Efficiency* of updating the scope will be the following calculation formula.

$$\text{Efficiency} = \frac{0.8\text{Dc} + 2.6\text{I} + 1.8\text{Dk}}{1.0\text{Dc} + 2.0\text{I} + 1.0\text{Dk}} \qquad (1)$$

I in this formula is the data size of an Interest Packet. *Dc* is the size of an encrypted content. *Dk* is the size of a key. The load by *Dk* and *I* increases. However, we can reduce the load by content from 1.0*Dc* to 0.8*Dc*. In many cases, *Dc* is far bigger.

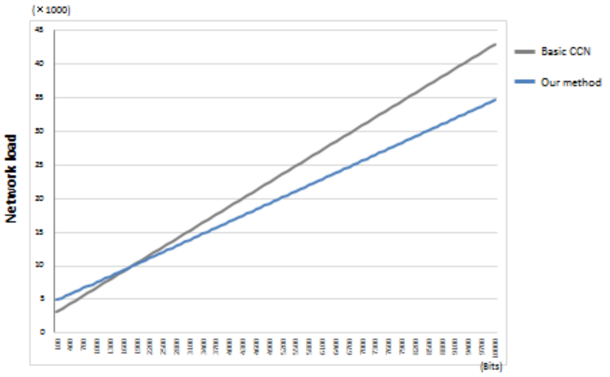


Figure 6. Comparison of Network Load

We compared the network load between our model and basic CCN, as shown in Fig. 6. When the content size is more than 2000 bits, the efficiency of our method is better than basic CCN. The bigger the content size is, the more efficient our proposal will be. However, when the content size is small, the traditional method is better. Therefore, the producer needs to choose which method to use. When we use the evaluation result of [5] as an argument, the calculated efficiency is 80%. In other words, our proposed model can reduce the network load by 20% on average.

In our model, the caching policy is based on on-path caching, which is a basic caching policy in CCN. However, many caching policies are already proposed [10,11,12]. If we combine our model with a appropriate caching policy, the efficiency will improve dramatically. In addition, our proposal is independent of security standards. Therefore, users can use any traditional security methods.

V. CONCLUSION AND FUTURE WORK

Content Centric Networking is a promising architecture in Future Internet research. It enables us to retrieve contents efficiently and reduces the network load. In addition, CCN realizes efficient mobility supports, high security, and fast content retrieval. CCN Routers have storage to cache contents, while acting as intermediate nodes. We can share contents efficiently by using caching function. In basic CCN, we encrypt contents to distribute them to particular group. However, when the scope of disclosure is updated, we have to abandon the cached contents. In this case, we have to re-encrypt the contents and re-send them. In other words, we cannot use caching function effectively. In this paper, we proposed a community model, a key distribution method, and content sharing model. By using the community model and the key distribution model, our proposed content sharing model enables us to share contents efficiently in a dynamic community. In our content sharing model, content producers create Data Key, which is abandoned instead of the content itself. It reduces the network load and extra processions. Fig.6 shows the efficiency of our proposed method. We can reduce the network load by 20% on average because the size of Data Key is much smaller than that of the content itself and we are able to take advantage of CCN caching function.

REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, and D. Kutscher, “A Survey of Information-Centric Networking”, IEEE Communications Magazine, Vol. 50, Issue. 7, 2012, pp. 26-36.

[2] M. F. Bari, R. Boutaba, and B. Mathieu, "A Survey of Naming and Routing in Information-Centric Networks", IEEE Communications Magazine, Vol. 50, Issue. 12, 2012, pp. 44-53.

[3] N. Fotiou, G.C. Polyzos, and D. Trossen, "Illustrating a publish-subscribe Internet architecture", Journal on Telecommunication Systems, Springer, 2011, pp. 233-245.

[4] N. Fotiou, P. Nikander, D. Trossen, and G.C. Polyzos, "Developing Information Networking Further: From PSIRP to PURSUIT", BROADNETS’10, 2010, pp. 52-58.

[5] V. Jacobson, et al., “Networking Named Content”, CoNEXT ’09, 2009, pp. 1-12.

[6] L. Zhang, et al., "Named Data Networking (NDN) Project", NDN Technical Report NDN-0001, 2010..

[7] V. Jacobson, et al., “VoCCN: Voice-over Content-Centric Nwtworks”, ACM ReArch’09, 2009, pp. 1-6.

[8] L. Wang, et al., "OSPFN: An OSPF Based Routing Protocol for Named Data Networking", NDN Technical Report NDN-0003, 2012.

[9] B. Goode, "Voice over Internet protocol (VoIP)", Proceedings of the IEEE, Vol. 90, Issue. 9, 2002, pp. 1495-1517.

[10] Z. Ming, M. Xu, and D. Wang, “Age-based cooperative caching in Information-Centric Networks”, INFOCOM WKSHPS’12, 2012, pp. 268-283.

[11] I. Psaras, W.K. Chai, G. Pavlou, “Probabilistic in-network caching for information-centric networks”, ICN’12, 2012, pp. 55-60.

[12] K. Cho, et at., “WAVE: Popularity-based and collaborative in-network caching for content-oriented networks”, INFOCOM WKSHPS’12, 2012, pp. 316-321.