Towards Early Deployable Content-Centric Networking Enhanced by using IPv6

Shingo Ata*, Hiroshi Kitamura†, and Masayuki Murata‡
*Graduate School of Engineering Osaka City University, Japan
Email: ata@info.eng.osaka-cu.ac.jp
†NEC Corporation / University of Electro-Communications, Japan
‡Graduate School of Information Science and Technology, Osaka University, Japan

Content-Centric Networking (CCN) is promising as a future communication paradigm to exchange contents without specifying address of nodes. However, deployment of CCN from clean-slate would take a long time since all routing facilities are needed to be replaced. For early deployment of CCN it is feasible to use the current Internet infrastructure as much as possible, and achieve minimum modifications of applications and/or end nodes. Based on such background we propose a novel approach which enables to deploy CCN rapidly with extremely small modifications compared to other approaches. Our solution is based on the use of IPv6 network as a lower layer routing scheme. To realize we design some important components such as Name Mapper and Address Mapper. We also discuss about a deployment scenario for early adaptation of CCN in IPv6 networks.

I. Introduction

Content-Centric Networking (CCN) is promising as a future communication paradigm to exchange contents without specifying address of nodes. Traditionally a communication in the Internet is based on *who or where*, i.e., a client specifies the address of the server which the client intends to communicate to. This is because in the beginning of the Internet, end users clearly understand roles of nodes, and can specify the exact address of the node which has the service or content that users want to get. However, many of communications in recent years are based on *what*, i.e., in many cases end users don't care about the location (address) of node to get the service or content. There is a gap between the current communication form in the Internet, and its style of usage. Instead, a search engine is widely used to find the appropriate node which provides the content.

Content-Centric Networking is expected to fill the gap; the communication can be established based on *what*, where end users don't need to care about exact addresses of nodes to communicate. The concept of CCN is expected as a new communication paradigm in the Future Internet. From this perspective, many architectures to realize the content centric networking are proposed in recent years. DONA [1] was one of the first clean-slate CCN proposals. Content-Centric Networking (CCN) [2] and NDN [3] propose a name-based routing mechanisms to forward packets based on interest and data. PURSUIT [4] and PSIRP [5] are publish/subscribe architecture for information centric networking.

However, the most of approaches taken in these proposals are from clean-slate, i.e., architectures are designed from scratch. Of course clean-slate approach is important to drastically solve problems that the current Internet has, however, it requires the replacement of network components such as routers, protocol stacks, and related systems (e.g., DNS) as well. This is a problem to deploy the CCN widely. We agree that the clean-slate approach is ideally better, but it is feasible to consider the coexistence of the new paradigm with the current IP infrastructure. From that point, to use the current Internet infrastructure as much as possible, and achieve minimum modifications of applications and/or end nodes would be realistic.

Also, security is an important issue in the current Internet, and should be considered more in CCN for protecting digital rights. Though many of above projects have a mechanism to protect the content, most of them are signature-based encryption. Of course the encryption is effective to protect the content by itself, a lightweight mechanism to enhance the security is expected as a built-in functionality of the network protocol.

This work is partally supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) by Ministry of Internal Affairs and Communications of Japan.

978-3-901882-50-0

Based on above mentioned backgrounds, in this paper we propose a novel communication architecture to realize the Content-Centric Networking by the enhanced use of IPv6 network. To realize, we introduce the mapping scheme which associate a content and an IPv6 address one by one. The end users only specify the name of content though, the actual routing is performed based on the current IPv6 routing. In addition, we use the Unified Multiplex communication architecture [6] which enables to use different *one-time* IPv6 addresses [7] to different contents, to enhance the security.

This paper is organized as following. We first describe the outline of our proposed CCN over IPv6 network in Section II. We then explain the details of mapping mechanisms in Section III. The use of the Unified Multiplex to enhance the security is discussed in Section IV. The overview of experiment and deployment are shown in Section V. Finally we conclude our paper with future topics in Section VI.

II. Content Centric Networking over IPv6: An overview

A. Criteria of Requirement

Before describing our proposed architecture of Content Centric Networking by using IPv6, we first enumerate basic requirements of CCN for early deployment.

- Use the existing facilities of the Internet as much as possible, and realize CCN with a small (almost none of) modifications of userland programs in end nodes, intermediate routers, and/or relative services.
- Support a gradual migration to CCN. We should consider a situation where both the current IP and CCN coexist in the same network. Also, a scenario for smooth migration to the CCN is necessary.
- Support dynamic placement of contents and mobility. One of the major advantages of the CCN is content caching for effective data transfer. Supporting mobility in the network protocol would decouple contents from temporal and spatial limitations.
- Support security for protecting contents. As already described, security is one of key issues to protect digital rights, privacy, and safety.

Based on above criteria, we next describe the overview of our proposed CCN architecture, and explain how our proposed architecture solves these requirements.

B. Content Name, Content Address, and Responder Address

Content Name is a human-friendly address which specifies the content. Typically something like file names are used for content names. To identify the content correctly, Content Name must satisfy the uniqueness (i.e., different contents have different content names), persistency, and scalability [8]. Currently, there are number of naming schemes used to identify contents. For example, NDN project uses a slash separated hierarchical naming structure (e.g., /picture/river.jpg), a similar expression is standardized by URN (Uniform Resource Name) [9] and DOI (Digital Object Identifier), which is used to permanently identify the digital documents [10].

Unlike these naming expressions which use / (slash) for separator of hierarchical structure, we use Reverse Domain Name instead which uses . (dot) for the separator. Reverse Domain Name is a notation to represent the resource by using FQDN (Fully Qualified Domain Name) like expression. The difference from FQDN is the direction of aggregation. In FQDN, the right end is the top level domain while it is the left end in Reverse Domain Name. Reverse Domain Name is firstly used in Java, and also used in some operating systems. The reason why we use the Reverse Domain Name for content names is to avoid modification of applications. Current Internet applications typically specifies FQDN of the server to establish a communication. If we use the Reverse Domain Name as the content name, the application recognizes that a FQDN is specified and pass a query packet to DNS server to resolve the FQDN. Otherwise, for example if we use URN like format, the application may fail due to a syntax error of the parameter.

In this architecture, we newly introduce two types of addresses in addition to *Content Name*. *Content Address* is IPv6 address to distinguish *Content Names*. Basically, *Content Address* should have an uniqueness according to *Content Name*, that is, a different *Content Name* has a different *Content Address* one by one. For protection of content, a single *Content Name* may have multiple *Content Addresses* which are used as one-time address for content retrieval. *Content Address* is a location-free address which is permanent regardless the location of the content server. To forward the packet properly in the existing IPv6 network, the actual address of the node is necessary. For this purpose *Responder Address* is used to specify the IPv6 address of the node having the content.

To obtain *Content Address* or *Responder Address*, two types of mapping scheme is introduced. *Name Mapping* and *Address Mapping* are used to obtain *Content Address* and *Responder Address* from *Content Name* and *Content Address*, respectively.

Figure 1 shows the summary of address types and

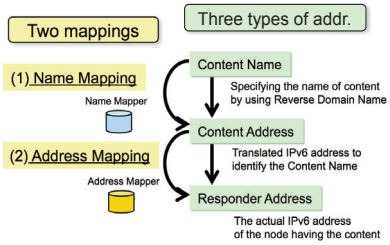


Fig. 1. Address Types and Mappings

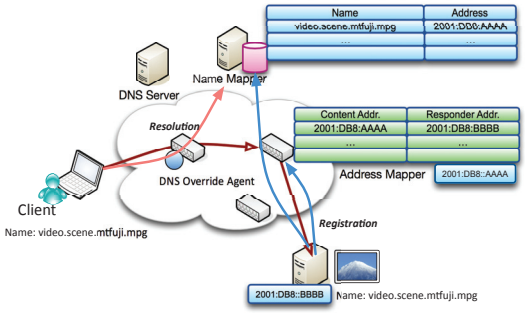


Fig. 2. Overview of Proposed CCN Architecture

mappings.

C. Overview of CCN over IPv6

Figure 2 shows the overview of our proposed CCN architecture by using IPv6. In addition to the current IPv6 network, two mappers (*Name* and *Address*) are deployed in this figure. Firstly, the server having the content `video.scene.mtfuji.mpg` perform a registration process. In the registration, two registration messages are sent. The one is to register the association between *Content Name* and *Content Address* in the Name Mapper. The other is to update *Responder Address* for *Content Address* in the Address Mapper.

Note here that *Content Address* is used for two purposes, *Content Address* is not only used to identify the content, but also is IPv6 address of the Address Mapper. This is a key for rendezvous of contents.

A client intend to get a content specified by `video.scene.mtfuji.mpg` the application simply sends a DNS query as done in specifying FQDN. A DNS override agent, located between the client and DNS server, hooks the query packet and forwards to the Name Mapper if the name in the query is *Content Name*. Then the Name Mapper receives the query and lookup the table to find the *Content Address* for the *Content Name*. If found, the DNS reply message is sent to the client whose resolved AAAA address is the *Content Address* (`2001:DB8::AAAA`). The client then send a packet to retrieve the content by specifying `2001:DB8::AAAA` as the destination address. The packet is forwarded to the *Address Mapper* according to a regular packet routing in IPv6. The *Address Mapper* then check the table to get the *Responder Address* of the content, and then the packet is forwarded to the server. To transfer the packet to the server, a couple of ways can be used such as tunneling, IP capsulation, or address translation. Finally, the server sends packets for the content back to the client.

In this case, all packets are communicated via the Address Mapper, which leads inefficiency of communication. A kind of route optimization described in Mobile IPv6 can be used to communicate directly between the server and the client.

III. Mapping Scheme

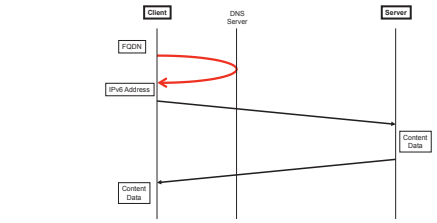
Figure 3 compares the communication sequence between the current IPv6 and proposed CCN. In the typical IP (in both v4 and v6), the application is initiated by specifying the FQDN of the destination node. According to the call of function `getaddrinfo`, the kernel library then sends a DNS query to the DNS server to resolve the IP address of the destination node. After resolving, the client then establish a connection by specifying the IP address of the destination.

On the other hand, in the proposed CCN architecture, the behavior of the client is exactly the same as the regular IPv6 case, though the user specifies the *Content Name* instead of FQDN. The client does not identify whether the specified name is *Content Name* or not. The identification is triggered by DNS override agent.

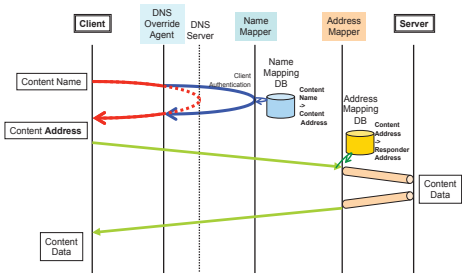
IV. Security Enhancement by Using Unified Multiplex

In this section we first explain briefly about the Unified Multiplex communication architecture [6]. Here, we refer the existing communication architecture as *Legacy Architecture* to distinguish from Unified Architecture.

Figure 4 compares communication styles in both Unified and Legacy architectures. In the Legacy architecture, three TCP connections use the same IP addresses (A and B), while the port numbers at Client B are different. These connections are distinguished at both nodes by the port numbers of connections. On the other hand, in the Unified architecture, different ephemeral addresses (E1, E2, and E3) for Client B are assigned to every connection, and different specific service addresses for Server A are also assigned.

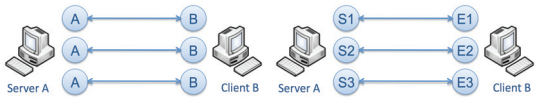


(a) IPv6



(b) CCN

Fig. 3. Communication Sequence



(a) Legacy (b) Unified

Fig. 4. Legacy/Unified Communication

The key features of the Unified Multiplex Communication Architecture are followings:

- We make use of the extensive address space which IPv6 has positively and assign different address for service or session.
- We prevent an unnecessary access from the outside by shortening an address life time in minimum requirement by assigning a different address to every session and disposing the assigned address after the session.
- By getting rid of the correlation among addresses, it is impossible to connect the service by using the randomly generated IPv6 address. It makes end users easier to deploy the server at the global network without any security considerations.

Even if the address of the server has been known to a third party, the address cannot be used anymore to connect the server in future because the address is only valid for a single session. We call it as *One-time* use of IP addresses.

The motivation behind the use of one-time IP address for CCN is to protect the content from unknown users. A single content address generated and assigned to the content is used only for the user who know the content address. From the wide range of the addressing space of IPv6, other users cannot infer the assigned Content Address.

V. Deployment

We currently have implemented and deployed both DNS override agent and the Name Mapper on FreeBSD 5.3R kernel. Our preliminary experiments show that we can retrieve the file by only executing `wget http:video.mtfuji.mpg`. No specific IP address is needed to download the file.

VI. Conclusion

We have proposed a new CCN architecture by enhancing existing IPv6 network. We hope our approach would be helpful for early deployment of CCN. Also, one-time address inspired by Unified Multiplex would improve the protection of contents. Since we only have a preliminary experiment, more detailed functionalilies should be designed and implemented in future.

References

[1] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 181–192, Aug. 2007.

[2] 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 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, (New York, NY, USA), pp. 1–12, ACM, 2009.

[3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (NDN) project," tech. rep., PARC, NDN-0001, October 2010.

[4] D. Trossen and G. Parisis, "Designing and realizing an Information-Centric Internet," *IEEE Communications Magazine*, vol. 50, pp. 60–67, July 2012.

[5] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 26–33, Apr. 2010.

[6] H. Kitamura, S. Ata, and M. Murata, "Communication architecture evolution enabled by introducing specific IP address for each session - unified multiplex communication architecture -," *submitted for publication*, Sept. 2010.

[7] S. Ata, H. Kitamura, and M. Murata, "Architectural design of unified multiplex communications for one-time use of IP addresses," in *Proceedings of 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–4, February 2011.

[8] L. Masinter and K. Sollins, "Requirements for uniform resource names," *RFC 1737*, December 1994.

[9] T. Berners-Lee, R. T. Fielding, and R. T. Fielding, "Uniform resource identifier (URI): Generic syntax," *RFC 3986*, 2005.

[10] H. V. de Sompel, T. Hammond, E. Neylon, and S. L. Weibel, "The "info" URI scheme," *RFC 4452*, 2006.