Virtual Content-Centric Networking

Keiichiro Tsukamoto[a], Masato Ohtani[a], Yuki Koizumi[a], Hiroyuki Ohsaki[b],
Kunio Hato[c], Junichi Murayama[d]
[a] Graduate School of Information Science and Technology Osaka University, Japan
Email: {k-tukamt,m-ohtani,ykoizumi}@ist.osaka-u.ac.jp
[b] Department of Informatics, School of Science and Technology, Kwansei Gakuin University, Japan
Email: ohsaki@kwansei.ac.jp
[c] NTT Secure Platform Laboratories NTT Corporation, Japan
Email: hato.kunio@lab.ntt.co.jp
[d] Department of Communication and Network Engineering,
School of Information and Telecommunication Engineering, Tokai University, Japan
Email: murayama@m.ieice.org

Abstract— **Data-centric networking has recently been gaining attention. A representative design for data-centric networking is Content-Centric Networking (CCN), which routes packets based on content identifiers. CCN is basically designed to be open because ease of data reuse is one of the greatest advantages of data-centric networking. However, for real-world networking, completely open data-centric networking is not sufficient; it is necessary to allow for private communication within a group of users. In this paper, we propose Virtual Content-Centric Networking (VCCN), which realizes private communication within a group of users through CCN router virtualization. We present four building blocks of VCCN: extension of the content identifier, CCN router virtualization, packet transport between virtualized CCN routers, and Social Network Services cooperative user/group identification. We have implemented VCCN's basic features by extending the CCNx software and have conducted a preliminary performance evaluation of our VCCN implementation.**

Index Terms— **Content-Centric Networking, Router Virtualization, Group-Based Communication, Social Network Services Cooperative User/Group Identification**

I. 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 [1]–[4].

A representative design for data-centric networking is Content-Centric Networking (CCN) [5], [6], 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.

Manuscript received June 26, 2013; revised September 13, 2013; accepted January16,2014.
This research is partly supported by Grant-in-Aid for Scientific Research (B) (25280030).

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 [6], 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 [6], [7].

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 paper, we focus 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.

We propose 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 [6].

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 networks, 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 paper are the following. First, we present a general and practical network architecture (VCCN) for constructing virtual private networks on

doi:10.4304/jnw.9.4.807-815

a content-centric network by CCN router virtualization. Second, we show that VCCN is scalable with respect to content request rate and the number of VCCN networks, through a preliminary performance evaluation of our VCCN implementation.

The organization of this paper is as follows. Section II contains a summary of related work. In Section III we give an overview of VCCN and its four building blocks. In Section IV, we describe our VCCN implementation and the results of a preliminary performance evaluation. In Section V, we discuss open research issues in VCCN network construction. Finally, in Section VI we give our conclusions and indicate future work.

II. Related Works

One attempt to realize group-based communication on data-centric networks is the Virtual Private Community (VPC) service [8], [9]. VPC is a CCN-based service architecture designed to share content among users who belong to the same network domain or to external domains. 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 the construction of virtual private community for a group of users, 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 paper was inspired by the Virtual Data-Oriented Network Architecture (VDONA) [10]. 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, significantly different from VDONA in terms of how a router is virtualized and how packet transport between virtualized routers is accomplished.

III. VCCN (Virtual Content-Centric Networking)

A. VCCN 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 network is shown in Fig. 1. Users are allowed to send Interest packets to VCCN networks that they belong to, and they can receive Data packets only from those networks. An Interest packet is routed within the VCCN network 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 network 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 network to which every Interest/Data packet belongs.

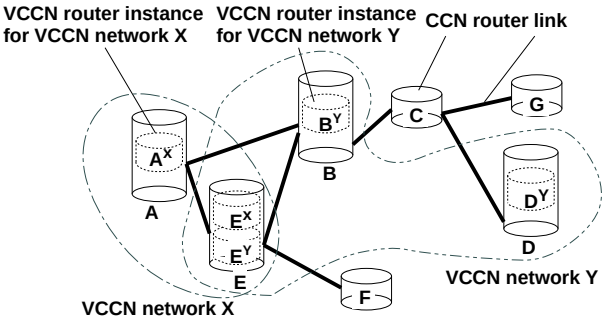

Figure 1. Example of a VCCN network built on a CCN network; two logically independent VCCN networks X and Y are built on the network of seven CCN routers, A through G.

- **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 running VCCN router instances) which are not adjacent in the CCN 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 networks. The last building block, SNS cooperative user/group identification, prevents injection of unauthorized traffic into a VCCN network by an outsider.

In the following, we describe these building blocks in more detail.

B. Extension of the Content Identifier

Content identifiers in CCN are extended to enable a virtualized CCN router to identify the VCCN network 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. 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 network 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 network.

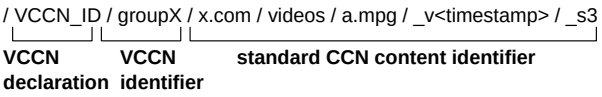

Figure 2. Example of an extended content identifier; the first two components are used as the VCCN declaration and the VCCN identifier.

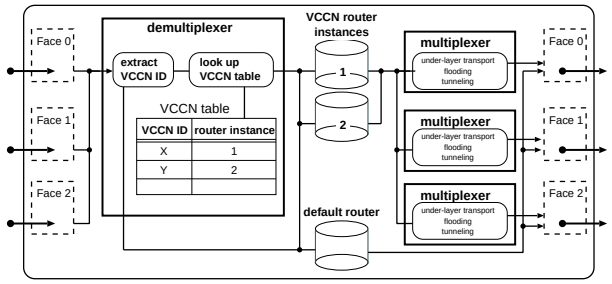

Figure 3. A virtualized CCN router; it is composed of a demultiplexer, VCCN router instances, and multiplexers.

C. 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) [6]. A CCN router can be equipped with multiple FIBs, CSs, and PITs and one of each of these tables is assigned to each VCCN network. 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. 3). We explain 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 network) 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.

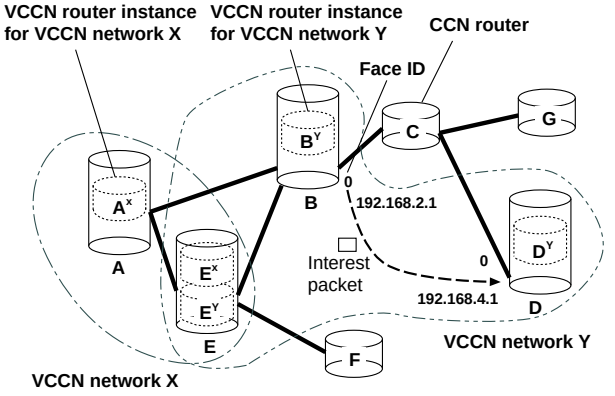

Figure 4. Packet transport in a lower layer; if a lower layer protocol supports communication between an arbitrary pair of nodes (e.g., IP, UDP, TCP, and broadcast communication), any pair of CCN routers can communicate using the lower layer protocol.

D. 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 CCN 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. 4). CCN can operate on variety of lower layer protocols such as IP, UDP, TCP, broadcast communication, Ethernet, and P2P [11].
  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. 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 CCN network. So flooding should not be permitted, especially when VCCN networks 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

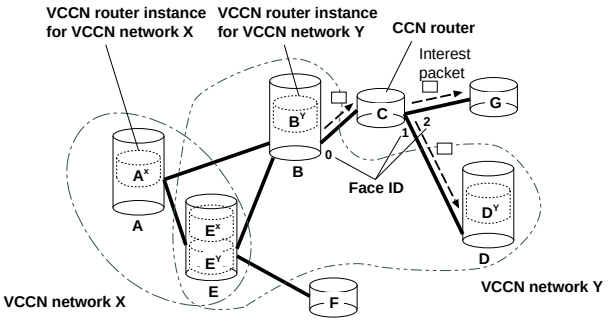

Figure 5. Flooding in the CCN layer; in CCN, flooding can be realized simply by duplicating Interest packets and sending them through all faces of every CCN router.

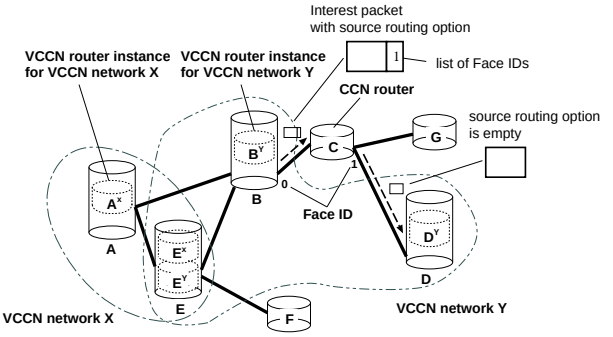

Figure 6. Tunneling in CCN layer; as in the source routing option of IP, the CCN router forwards the packet to the face listed at the head of the source route.

and inefficiency caused by the flooding in CCN layer is not acceptable, tunneling in CCN layer can transport packets between virtualized CCN routers (Fig. 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 [12]. However, tunneling in the CCN layer is still realizable with source routing [13].

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* [6]. 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 [13], 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.

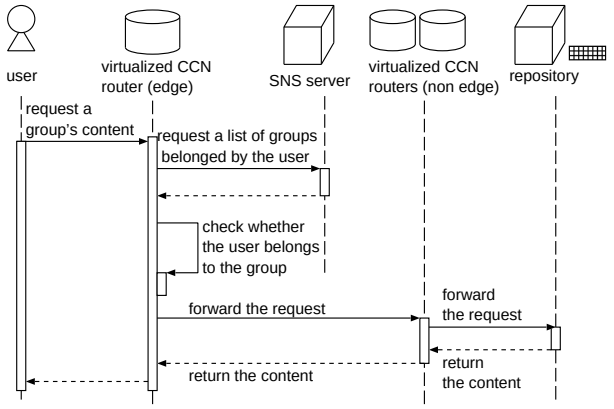

Figure 7. Sequence diagram for requesting content in VCCN.

E. 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. We believe 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 [14].

In SNS cooperative user/group identification, virtualized CCN routers at the edge of a VCCN network identify whether a user is allowed to access that VCCN network. Access to a VCCN network 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 network is also a proxy for an SNS authentication service (see Fig. 7). When users want to access a VCCN network, they communicate with a router at the edge of the VCCN network 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 network. 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 network, every router at the edge of the VCCN network is a proxy to a SNS authentication service, too (see Fig. 8). A repository communicates with a router at the edge of the VCCN network before content registration. The repository sends its identification information (e.g., repository name and password in an SNS). As with access to a VCCN network,

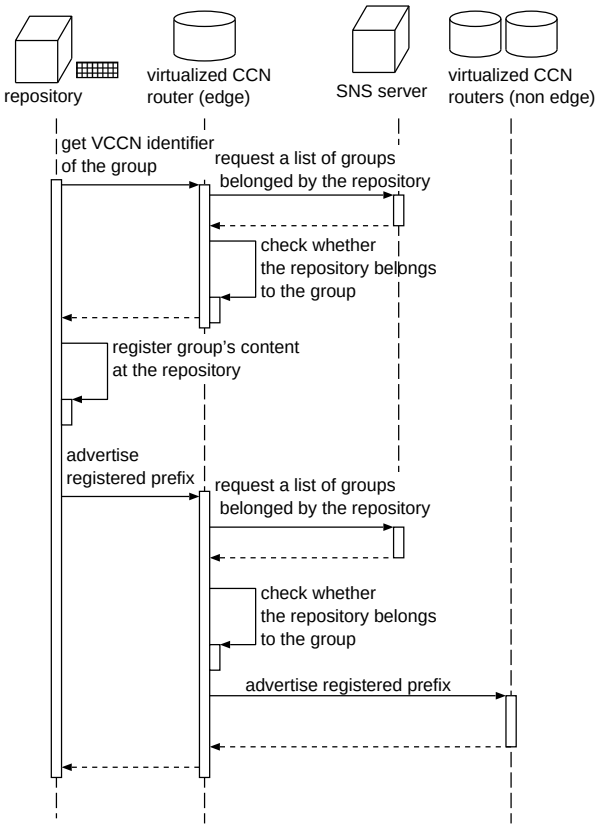

Figure 8. Sequence diagram when registering a content in VCCN.

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 network. 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 [6] in order to advertise the prefix of the registered content to VCCN routers on the VCCN network. In a *Register* operation, the repository forwards its identification information and an Interest packet to advertise the prefix to the router at the edge of the VCCN network. The repository is allowed to advertise the registered prefixes to routers in the VCCN network only if both identification checks are successful again.

Although such cross-layer cooperation may debase the performance of a VCCN network, 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.

IV. Implementation and Evaluation

A. VCCN Implementation

We implemented VCCN's basic features by extending the CCNx software [5], 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 network.

Our VCCN implementation realizes traffic separation for VCCN networks in the following way. An edge router of a VCCN network 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 network: specifically, every FIB entry is tagged with a VCCN identifier. For simplicity, the CS and PIT are shared among all VCCN networks. Packet transport between virtualized CCN routers is realized with a lower layer protocol (UDP).

We prevent the injection of unauthorized traffic from a user using Facebook's authentication mechanism. When a user/repository accesses a VCCN network, 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 [15] 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.9). In the implemented identification, when a user/repository accesses to a VCCN network, the user/repository passes an access token and a group name to the edge router of the VCCN network. 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 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. 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

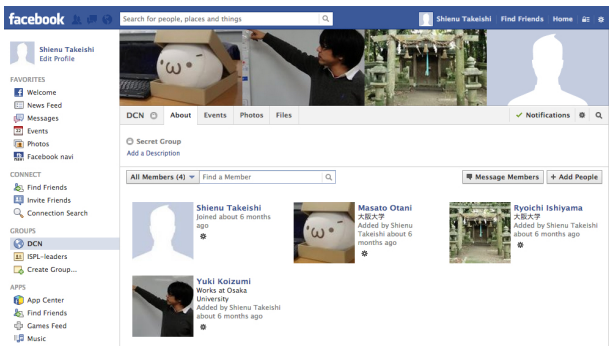

Figure 9. Example of creating a group; four members are registered with the data-centric networking group on Facebook and every registered member can take part in group-based communication.

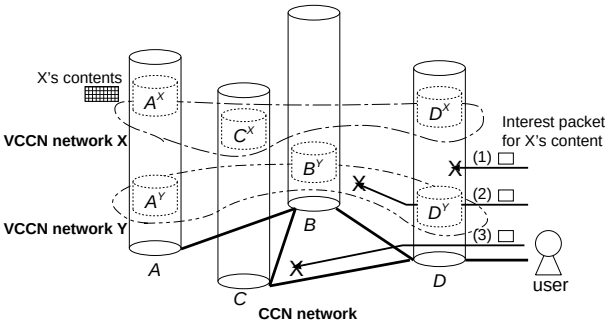

Figure 10. Processes for discarding three types of packet: (1) an Interest packet for X's content that a user who does not belong to any group requests through VCCN; (2) an Interest packet for X's content that a user belonging to Y requests through VCCN; and (3) an Interest packet for X's content that a user belonging to X requests through CCN.

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 network 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 CCN network.

B. Performance Evaluation of the VCCN Implementation

We 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, we evaluate the scalability of virtualized CCN routers in a CCN network.

For the first experiment, we used the network topology shown in Fig. 11—four CCN routers are connected, and two VCCN networks 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

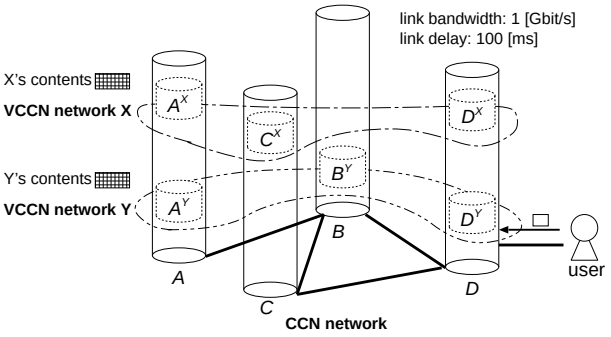

Figure 11. Network topology used in the CNNx/VCCN comparison; four CCN routers are connected and two VCCN networks, X and Y, are created.

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 network 3,000 times. Note that the average hop count from the source (CCN router A) to the user (CCN router D) is 1.5 (i.e., one hop in VCCN network Y and two hops in VCCN network 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. We measured the content delivery delay disregarding the delays caused by identification processing.

Figure 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 we do not claim that VCCN is more efficient than CCN. Instead, we just addressed the question of whether the introduction of VCCN has a positive or negative impact on CCN performance. We are planning to conduct more detailed experiments.

Secondly, since it is expected that the performance of VCCN networks will be debased by CCN router virtualization, we evaluated the scalability of virtualized CCN routers in our VCCN implementation. We consider

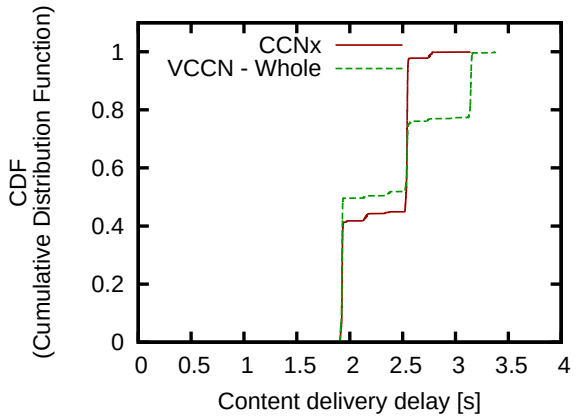

Figure 12. CDFs for content delivery delay when content is requested through a CCN network and the VCCN networks.

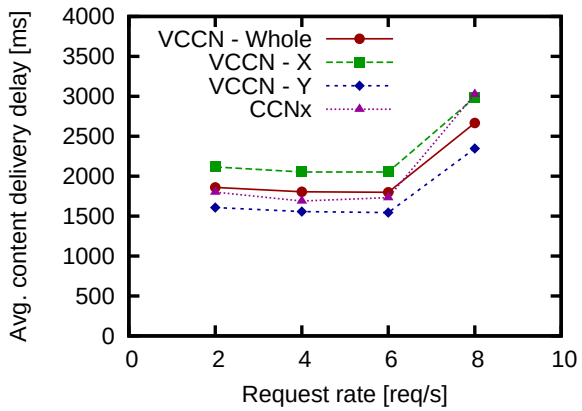

Figure 13. Average content delivery delays against content request rate.

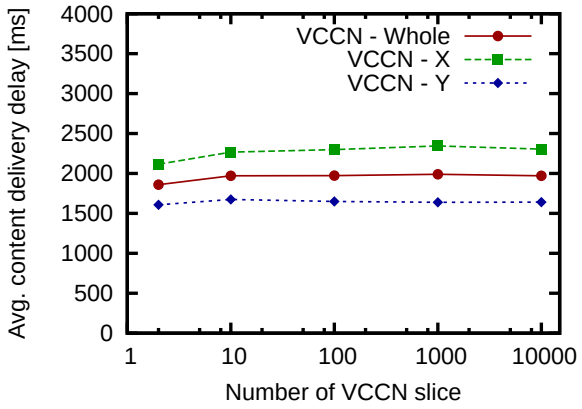

Figure 14. Average content delivery delays against the number of VCCN networks in a CCN network.

two types of scalability: scalability with respect to request rate and scalability with respect to the number of VCCN networks. 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. 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 network because the average content delivery delays in our VCCN implementation and in CCNx both increase at a rate of 8 [request/s]. Therefore, a VCCN has the scalability of a virtualized CCN router with respect to content request rate.

Next we considered how average content delivery delays vary with the number of VCCN networks in a CCN network. When increasing the number of VCCN networks, the number of content items in the CCN network is fixed and equal numbers of topologies X and Y for the VCCN networks are constructed. For example, when the number of VCCN networks 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. 14 indicates that the performance of virtualized CCN routers is not debased even if the number of VCCN networks is substantially increased. Moreover, the fact that the average content delivery delays are maintained does not depend on the network topology. Therefore, VCCN also has the scalability of a virtualized CCN router with respect to the number of VCCN networks that can be accommodated.

V. Open Issues

In this section, we discuss open research issues of VCCN network construction based on knowledge acquired by designing, implementing and evaluating VCCN networks.

A. CCN Router Resource Management

One important issue for virtualizing a CCN 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 and robustness of a network.

In CCN network virtualization, we 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 [16], [17]. 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 [16]. 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 network, that user can obstruct another VCCN network by interest flooding [18].

To prevent resource occupation of a CCN network and improve overall network performance, we should design a method to allocate the resources of a physical CCN router to each group. Some related methods have been proposed [17], [19] and our research group is planning to investigate analytically the effect of CS allocation methods and content request patterns in VCCN networks on the average content delivery time of each separate VCCN network and the entire network.

B. VCCN Network 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 [20]–[22]. Since this mapping is an NP-hard problem, heuristics for providing efficient performance were proposed in these studies.

In CCN network virtualization, existing virtual network mapping methods may not be applicable because these methods do not take data reuse into account. Mapping VCCN networks influences the effect of caching as well as performance and traffic. For example, in the experiment of Section IV, 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 network. It is desirable to study this problem, taking the effect of caching into account.

C. 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 networks and a substrate CCN 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 specify 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 networks are constructed on a CCN 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 network 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 network 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, we 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 CCN network going down.

VI. Conclusions

In this paper, we have 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 networks, which are composed of multiple VCCN router instances.

We have 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 networks. We have also discussed open research issues in VCCN network construction based on knowledge acquired by designing, implementing and evaluating VCCN networks.

In the future we will consider who names and manages VCCN identifiers and how such tasks should be done. We will also consider where VCCN router instances should be created or removed when a group is changed.

References

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

[2] 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)*, Dec. 2008, pp. 1–6.

[3] 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, no. 3, pp. 121–127, Mar. 2011.

[4] 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.

[5] "Project CCNx," http://www.ccnx.org/.

[6] 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 2009)*, Dec. 2009, pp. 1–12.

[7] L. Zhang *et al.*, "Named Data Networking (NDN) project," http://www.named-data.net/ndn-proj.pdf, Palo Alto Research Center, Tech. Rep. NDN-0001, Oct. 2010.
[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)*, Jan. 2011, pp. 843–844.
[9] D. Y. Kim and J. Lee, "CCN-based virtual private community for extended home media service," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 532–540, May 2011.
[10] 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.
[11] W. A. Simpson, "The Point-to-Point Protocol (PPP)," *Request for Comments (RFC) 1661*, July 1994.
[12] ——, "IP in IP tunneling," *Request for Comments (RFC) 1853*, Oct. 1995.
[13] J. Postel, "Internet protocol," *Request for Comments (RFC) 791*, Sept. 1981.
[14] 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, no. 12, pp. 1926–1938, Aug. 2010.
[15] "Facebook Graph API," http://developers.facebook.com/docs/reference/api/.
[16] K. Ohsugi, K. Tsukamoto, and H. Ohsaki, "A study on the effect of ccn router virtualization on content delivery time (in Japanese)," *the 2012 IEICE Society Conference(B-7-4)*, p. 83, Sept. 2012.
[17] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Evaluating per-application storage management in content-centric networks," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 750–757, Apr. 2013.
[18] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named-data networking," http://arxiv.org/abs/1208.0952, Tech. Rep., 2012.
[19] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *Proceedings of 10th IEEE International Conference on Communications (ICC '11)*, June 2011, pp. 1–6.
[20] G. P. Alkmim, D. M. Batista, and N. L. da Fonseca, "Mapping virtual networks onto substrate networks," *Internet Services and Applications*, vol. 4, no. 3, pp. 1–15, Jan. 2013.
[21] 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)*, Dec. 2008, p. 15.
[22] J. Lu and J. Turner, "Efficient mapping of virtual networks onto a shared substrate," WUCSE-2006-35, Tech. Rep., 2006.

Keiichiro Tsukamoto received Bachelor of Information Science and Master of Information Science degrees from Osaka University in 2009 and 2011, respectively. He has been a graduate student of the Graduate School of Information Science and Technology, Osaka University since April 2011. His research interests are in the area of Web content mining and Content-Centric Networking. He is a member of the IEICE.

Masato Ohtani received Bachelor of Information Science and Master of Information Science degrees from Osaka University in 2011 and 2013, respectively.

Yuki Koizumi received the M.E. and D.E. degrees in Information Science from Osaka University, Japan, in 2006 and 2009, respectively. He is currently an Assistant Professor at the Graduate School of Information Science and Technology, Osaka University, Japan. His research interest includes traffic engineering in photonic networks and biologically inspired networking. He is a member of IEEE and IEICE.

Hiroyuki Ohsaki received the M. E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 1995. He also received the Ph. D. degree from Osaka University, Osaka, Japan, in 1997. He is currently a professor at Department of Informatics, School of Science and Technology, Kwansei Gakuin University, Japan. His research work is in the area of design, modeling, and control of large-scale communication networks. He is a member of IEEE, IEICE, and IPSJ.

Kunio Hato received B.E. and M.E. degrees from Tokyo Institute of Technology in 1997 and 1999, respectively. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 1999, he has been engaged in research and development of IP VPNs, Wide Area Ethernet, network security systems and intercloud computing systems. He is now a senior research engineer in the Secure Communication project of NTT Information Sharing Platform Laboratories. He is a member of IEICE.

Junichi Murayama received B.E. and M.E. degrees in electronics and communication engineering from Waseda University in 1989 and 1991, respectively. He also received Ph.D. degree in information science and technology from Osaka University in 2011. From 1991 to 2013, he had worked for Nippon Telegraph and Telephone Corporation (NTT). He is currently a professor at Department of Communication and Network Engineering, School of Information and Telecommunication Engineering, Tokai University, Japan. He has been engaged in research and development of ATM networks, IP VPNs, optical IP networks, network security systems and intercloud computing systems. He is a member of the Institute of Electronics, Information and Communication Engineers(IEICE) and the Institute of Electrical Engineers of Japan(IEEJ).