Retrieving Information with Autonomously-Flying Routers in Information-Centric Network

Taku Kitagawa
Graduate School of Information Science and Technology
Osaka University, Japan
1–5 Yamadaoka, Suita, Osaka, 565–0871
Email: t-kitagawa@ist.osaka-u.ac.jp

Shingo Ata
Graduate School of Engineering
Osaka City University, Japan
3–3–138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka 558-8585
Email: ata@info.eng.osaka-cu.ac.jp

Masayuki Murata
Graduate School of Information Science and Technology
Osaka University, Japan
1–5 Yamadaoka, Suita, Osaka, 565–0871
Email: murata@ist.osaka-u.ac.jp

***Abstract*—Information-Centric Networking (ICN) enables retrieval of content with various controls using flexibility of a content name, which is treated as a routing information. In existing works, however, data processings on end nodes or routers are mainly considered. Therefore, controls of physical behaviors, especially actual movement of devices, are not considered so much. we consider to incorporate physical movement of routers into routing control in ICN, and we propose a novel architecture of ICN that makes it possible to retrieve unretrievable contents between physically disjoint networks, by only specifying its name, which is realized by a Content-Centric Networking (CCN) router mounting on an Unmanned Air Vehicle (UAV). Moreover, we design and develop a Flying Router (FR), which combines a CCN router and a UAV, and we propose a detail design of communication architecture in which multiple FRs are able to process routing cooperatively among disjoint networks. In addition, we examine feasibility of the proposed architecture by carrying out a basic experiment using a prototype of an FR.**

I. Introduction

Information-Centric Networking (ICN) is focused as a novel Internet architecture realizing content-oriented communication [1], [2], [3]. ICN controls routing by defining a name of content as a destination address of a packet. Routing control by a content name relaxes tightly coupling between content and its holder (e.g., content server), therefore ICN improves independence and mobility of content, and communication performance by content caching and multicasting.

Recently, moreover, research works that retrieve content flexibly by incorporating control information (e.g., signaling) into its name, which improves flexibility as a routing identifier, are studied such as Voice over CCN (VoCCN) [4]. ICN supports retrieving content flexibly with seamless data processes.

In the existing works, however, data processing on end or intermediate routers are mainly considered, such as video transcoding, encryption, data filtering, and so on. Hence, controls of physical behavior including actual movement of devices are not considered so much. In this paper, we consider to incorporate physical movement of routers into routing control in ICN. If routing control with physical movement of routers becomes possible by only specifying a content name, information sharing between physically disjoint networks becomes possible.

We first introduce related works in Section II, and network environment with autonomously-moving routers is proposed in Section III. We then describe a detailed design of the proposed architecture in Section IV. An implementation of a prototype of the autonomously-moving router is presented in Section V. Also, a basic experiment about retrieving content between disjoint networks with the prototype is described in Section VI. Finally we give a conclusion with future works in Section VII.

II. Related Works

ICN uses a content name as a routing identifier. Existing works use content names as not only routing information but control information because of extent of name domain and simplicity of giving meaning.

VoCCN [4] realizes a voice talking system like VoIP over Content-Centric Networking (CCN), which is advocated by Palo Alto Research Center (PARC). CCN is widely referred as an implementation of ICN. VoCCN enables setting up a voice channel by expressing SIP signaling in a content name. It results in putting a signaling path and a media path together into one path simply. Moreover, DASH over CCN [5], which applies CCN to media streaming, can change bitrate, resolution and other quality of video in units of a segment by issuing an Interest message every segment. It results in improving convenience by change quality of content flexibly.

As mentioned above, CCN makes flexible control possible on network layer, which is difficult to realize in only existing IP layer. In these works, however, data processings on end nodes or routers are mainly considered, and there is no research work that extends routing control of routers to realize its physical movement to the best of our knowledge. Thereby, an autonomously-moving router proposed in this paper indicates possibility of novel control of CCN, and it can be an efficient information-distribution infrastructure taking advantages of CCN especially in physically disjoint networks.

On the other hand, research works on Delay Tolerant Networking (DTN) are currently carried out actively as an information-distribution infrastructure in physically disjoint networks. In an example [6], an Unmanned Air Vehicle (UAV) crawls to find disjoint networks that are decoupled by disaster and the UAV makes possible to share data between disjoint

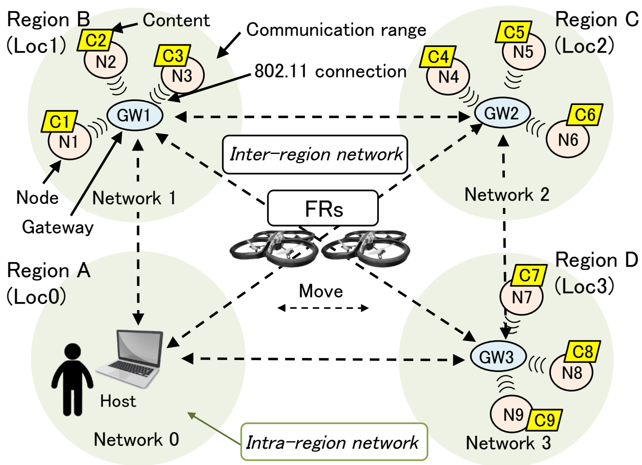

Fig. 1. Target Networks

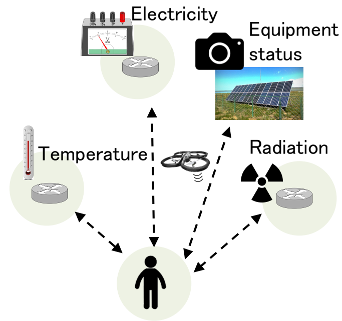

Fig. 2. Exchange of Sensor Data

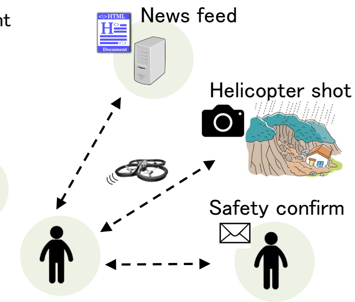

Fig. 3. Delivery of Disaster Info

networks with a DTN protocol. However, DTN is commonly a passive communication protocol, which communicates with nodes if available. Therefore, DTN does not control routing actively and it is not a fully optimized routing control. ICN with an autonomously-moving router is able to control routing actively at routers and it optimizes delivery of content by taking the advantages of ICN (e.g. using content caches, content names and so on). It results in improving communication performance (e.g. arrival time of messages and high prevalence of messages in decoupling networks).

III. CCN Architecture with Flying Routers

We focus on CCN as implementation of ICN and we propose a CCN Flying Router (FR), which is a CCN router mounting on a UAV as an autonomously-moving CCN router. Adopting a UAV as moving body of a router makes it possible to provide connectivity for disjoint networks without influence on terrain. This section defines detail of disjoint networks and describes application examples. Moreover, we describe differences in features between IP FR and CCN FR, which adopts IP or CCN as network architecture of FR, to advocate advantages of CCN FR.

A. Target Networks

Network environment which we focus on is a situation that independent networks are distributed disjointly as shown in Fig. 1. Such situation can be found in the cases where (1) a portion of network is damaged due to e.g., a disaster, or (2) a sensing infrastructure within very vast area such as a farm.

Each independent network consists of a cluster of nodes with a gateway (GW) or a single host (like Network 0). Nodes connect with the gateway with wireless communication like 802.11. We define two types of network, one is called an *intra-region network*, which is an independent network described above, and *inter-region network*, which is a geographic network for connecting independent networks. In this paper we consider to use cooperative multiple FRs to construct the *inter-region network*. Note here that the *inter-region network* does not always have a connectivity but a kind of moving entity supports to deliver messages. The cooperation of multiple FRs results in shortening moving distance (pr reducing battery consumption) of FRs, and shortening time of retrieving content by content caches. The more multiple FRs are used, the more availability and/or efficiency can be achieved.

B. Application Examples

We consider exchange of sensor data or disaster information among widely distributed disjoint networks, as application. In a disjoint sensor network, sensors are distributed within a vast area, which are not connected directly, and collection of sensor data is performed by the physical movement of an entity (i.e., FRs), as shown in Fig. 2. By using ICN, it is possible to retrieve data by only specifying a type of a sensor (e.g. temperature in Fig. 2) in its content name, or to retrieve and dynamically generate content which is suitable for its name (e.g. a picture of solar panel status) by mounting sensors (e.g. camera) on a UAV. These features are similar in the case of a delivery of disaster information as shown in Fig. 3. Disaster-information delivery is performed in a decoupling network where natural disaster occurs, and makes it possible to retrieve news regarding disaster or safety confirmation for someone by only specifying its property. In a disaster case, frequently-requested content such as news regarding disaster is expected to be stored as a content cache, thereby it is possible to shorten moving distance of FRs greatly.

C. Differences between IP and CCN on FRs

Of course an FR is also implementable for IP, however, applying FR to CCN has following benefits.

- A content name can also be used to manage behavior of FR itself in addition to message routing. It makes possible to control FRs flexibly while forwarding messages.
- A content cache can shorten moving distance of FRs. It also results in shortening time of retrieving content and reducing battery consumption.
- Dynamic generation of content can be performed by Interest (e.g., a helicopter shot at an arbitrary location).

It is difficult for IP FR to realize these features, because we need to specify the exact IP address of FR to control and manage FRs. On the other hand, IP FR can support push type

TABLE I
FRs' FIB (Fig. 1)

Name	I/F
C1, C2, C3	GW1
C4, C5, C6	GW2
C7, C8, C9	GW3

TABLE II
FRs' Interface Table (Fig. 1)

I/F	Move to
Host	Loc0
GW1	Loc1
GW2	Loc2
GW3	Loc3

communication simply because of its communication principles unlike receiver-driven communication in CCN. However, support of push type communication in CCN is also studied (e.g., [7], [8]) that can be applied to our architecture, or we briefly describe push type communication in the proposed architecture by using Pub/Sub model in Subsection IV.

IV. Architectural Design

This section describes design of a proposed architecture where CCN communications within disjoint networks are realized by physical movement of FRs. We first define three major modes of FRs, i.e., waiting mode, crawling mode and delivering mode, which are important to form a cooperative behavior of FRs. We also design a method for making division of roles among multiple FRs by signaling protocols.

A. Routing Information with Physical Movement

We first design routing information for processing name base routing in an *inter-region network*. We define an *interface* as a name of node, and we propose an Interface Table which associates the name and its geographical location. In CCN, Forwarding Information Base (FIB) and Pending Interest Table (PIT) have interfaces as a next hop for the routing prefix. For example, we consider the case shown in Fig. 1.

Tables I and II show the FIB and Interface Table of FR, respectively. An FR defers message forwarding temporarily when the location of the *interface* presented in FIB is far from the current location, and the FR moves first to a geographical location of the gateway associated with the *interface*, finally the FR resumes the message forwarding after the FR arrives in the coverage area of the gateway (the detail is described in Subsection IV-D). The proposed network is able to deliver content seamlessly in the same procedure as normal CCN by extending the process of physical movement of an FR. In addition, *interfaces* are virtual information which is generated dynamically on FRs. This brings benefits that FRs can deal with a change of the number of *intra-region networks*. Moreover, it is possible to add movement control to CCNx [9], as an implementation of CCN used in this paper, without any modification in its specification.

B. Waiting Mode

An FR with waiting mode (waiting FR) waits at Rendezvous Point (RP), which is a central position of an *inter-region network* as shown in Fig. 4. Basically, the location of the RP is calculated from a centroid of a polygon constructed by connecting gateways of disjoint networks, in which the length of the side of polygon can be a weighted cost by taking both

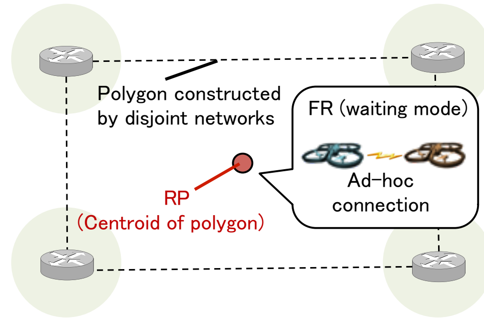

Fig. 4. Waiting Mode

physical distance and access frequency into consideration, i.e., the length may be shortened if the node is frequently accessed. The position of RP is updated by a fixed interval, T_{RP}.

Around RP, waiting FRs are connecting with each other by ad-hoc mode. Waiting FRs exchange their routing information (FIB, PIT, and Interface Table) each other and update them. Also, some statistics such of the number of packets forwarded per interface are exchanged to decide future strategy of physical movement. A method of collecting these information is explained in Subsections IV-C and IV-D.

C. Crawling Mode

An FR with crawling mode (crawling FR) collects routing information of *intra-region networks* and checks whether *intra-region networks* have messages that are necessary to be transferred through an *inter-region network*. The outline of crawling mode is shown in Fig. 5. When a crawling FR arrives in a disjoint network by connecting with a gateway of the *intra-region network*, the crawling FR and the gateway exchange routing information by advertising modification of FIB from the last exchange. If the gateway has messages that are necessary to be transferred, the crawling FR receives them, which will be delivered to RP at the end of crawling.

A crawling FR craws to cover all regions of the predefined area (i.e., *inter-region network*), and collects all gateways of *intra-region networks*. Based on the history of gateways, a crawling FR collects updated information of routing, as well as actual messages to be forwarded from gateways. If a crawling FR finds a new *intra-region network* by connecting with an unknown gateway, the crawling FR collects routing information from the gateway, and add a new entry in its Interface Table.

At the end, i.e., after a crawling FR arrives at RP, it broadcasts the updated information to other waiting FRs, and decides one FR to forward the received message, based on the physical distance to the destination gateway and remained battery in FR. The selected FR performs transition to delivering mode. After forwarding messages, the crawling FR continues crawling along *intra-region networks*.

A crawling FR keeps connecting with a gateway of an *intra-region network* for time T_c or until the number of messages which the crawling FR receives achieves N_c. Also, FRs with crawling mode and delivering mode have access points SSIDs

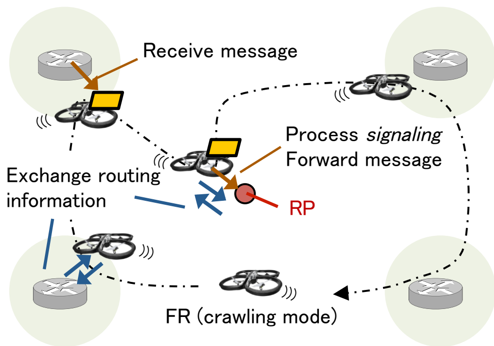

Fig. 5. Crawling Mode

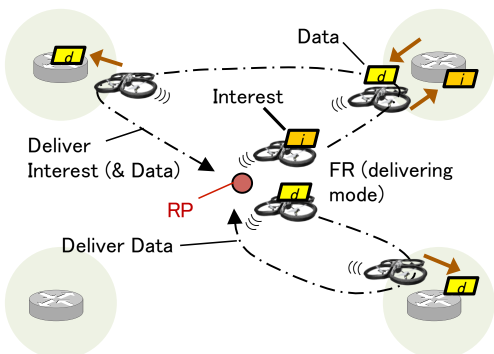

Fig. 6. Delivering Mode

of which are the same unanimously to connect with gateways of all *intra-region networks* seamlessly.

D. Delivering Mode

A waiting FR which receives messages from a crawling FR turns into an FR with delivering mode (delivering FR). A delivering FR processes content routing differently by a type of the messages it receives (Interest or Data) as shown in Fig. 6.

When the message type is Interest, the delivering FR first moves to the location of the destination gateway of the *intra-region network*, where the provider of the specified content is located, by looking up the Interface Table, and obtain its geometrical position. Then the delivering FR forwards the Interest message to the gateway and waits for the reply (i.e., Data packet).

After receiving Data message for the Interest message, the delivering FR then moves to another network. Note that if the delivering FR immediately leaves just after receiving a single message, the movement of FR becomes so frequent as the increase of the number of messages. Therefore we set two thresholds to let FRs keep a standby condition (i.e., waiting for any messages from other nodes) before leaving. We define the maximum time T_d and the maximum number of messages N_d for keeping standby condition. Namely, before leaving, the delivering FR waits for up to N_d messages or T_d seconds to receive any messages to be delivered.

After leaving, the delivering FR moves differently according to status of Data messages. If the FR has a capability to deliver

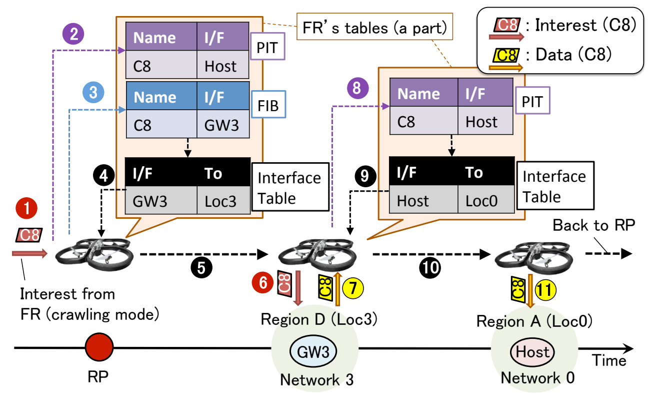

Fig. 7. Flow of Retrieving Content C8 from Host (Fig. 1)

the Data message (i.e., FR has enough battery to move to the destination), the FR directly moves to the location of the *intra-region network* to deliver the Data. Otherwise (or FR has no message to deliver), the FR goes back to RP.

When the message type is Data, the delivering FR just forwards the Data message to the gateway of an *intra-region network* where the content consumer exists by looking in PIT and Interface Table, and the delivering FR goes back to RP.

E. Basic Sequence of Message Delivery

Fig. 7 shows an example that Host in Region A retrieves content `C8` in Fig. 1. First, a crawling FR receives an Interest message for `C8` from Host, then the crawling FR updates its PIT by adding (C8, Host) as the entry. Also, by sensing the geometrical position from GPS, Interface Table is updated by (Host, Loc0). After going back to RP, the crawling FR exchanges information with other waiting FRs, and one of FR is decided to deliver the Interest. The delivering FR then looks up the FIB and finds the *interface* for the content `C8` (GW3 in this case). Then the FR moves to the location of GW3 obtained from Interface Table, and forward the Interest message to GW3. After receiving the Data (i.e., content) message from GW3, the FR goes to the location of Host by looking PIT and Interface Table. Finally, the FR goes back to RP and transits to the waiting mode.

An FR also supports push type communication by Pub/Sub model. Push type communication are realized by specifying strings, which means request of subscribing for pushed content, in a content name of an Interest message presented in e.g., [7]. An entry of PIT about push type communication is shared among waiting FRs, and the entry remains in PIT even if its data message has been forwarded until the FR receives an Interest message which stops subscribing for pushed content.

F. Signaling Procedure for Selection of Delivering FRs

A crawling FR processes *signaling* to select one delivering FR from waiting FRs at RP.

Signaling is processed every message. A crawling FR can select one delivering FR from Waiting FRs by broadcasting a new Interest message for *signaling* to waiting FRs and

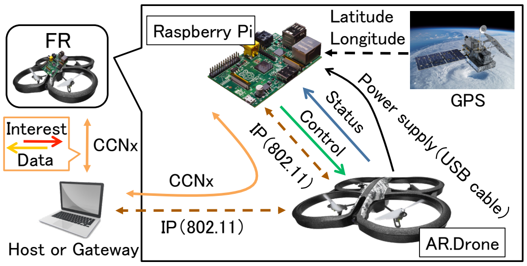

Fig. 8. Structure of FR

receiving the reply. A content name of the Interest message for *signaling* is specified by a name of the target content which is transferred and a special string `'?ask'` is appended to the original content name, which means a *signaling* message. When the original content name, which a consumer tries to retrieve a helicopter shot at a certain geographical location, for example, the content may be named such as `ccn:/domain/heli_shot/LAT,LON` and the *signaling* message is named to `ccn:/domain/heli_shot/LAT,LON?ask`.

A waiting FR receiving a *signaling* message sends back a reply message as Data if the waiting FR is capable to retrieve the content. the reply includes an identifier, specification of the waiting FR, and whether cache of the content is hit. Then, the crawling FR decides a delivering FR from waiting FRs which send back the reply. If there is no reply, the crawling FR turns into delivering mode and delivers a message by itself.

G. Specification of Gateways

A gateway of each *intra-region network* mainly collects routing information (e.g., collects a list of content entries) in its network and the gateway keeps holding messages that are necessary to be transferred through a *inter-region network* until a crawling FR comes into the communication area. A gateway collects routing information by being advertised from each node connected with the gateway.

V. Implementation

We developed a prototype of an FR and carried out a basic experiment using the prototype in disjoint networks actually. This section describes implementation of the prototype. The prototype realizes delivering mode, which forms the basis of an FR. In the implemented network, RP is pretermitted and direct movement between two regions is only considered because the experiment supposes to use one FR.

A. Structure of FR

The prototype of *controller* is configured by a Raspberry Pi B+ [10], mounted on the top of UAV, to process CCN routing and movement control of UAV. Also, an AR.Drone 2.0 Power Edition [11] is adopted as a UAV. AR.Drone is equipped with an access point (IEEE 802.11b), and the *controller* uses the access point to construct an *inter-region network*, i.e., to connect between an FR and each gateway of *intra-region networks*. Moreover, CCNx [9] is used as an implementation of the CCN architecture.

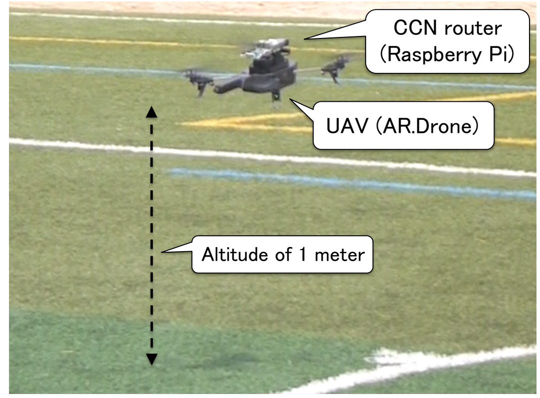

Fig. 9. Picture of FR

TABLE III
Software Specification of A Prototype

Raspbian	Debian base linux for Raspberry Pi (Ver. 2014-12-24-wheezy-raspbian)
Java	JDK 8u31
CCNx for Java	Ver. 0.8.2
ARDroneForP5 [12]	AR.Drone API for Processing (Ver. 2.0)

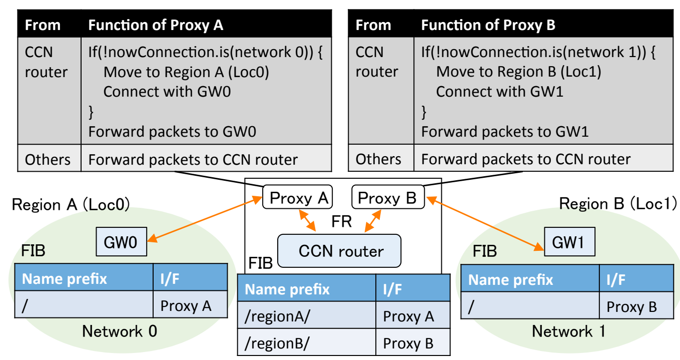

Fig. 10. Process Overview of Virtual Proxy Model

Fig. 8 shows specific structure and Fig. 9 shows a picture of the prototype. Moreover, Table III shows a list of used software in the prototype. The *controller* is supplied with power by a USB port of the AR.Drone and it processes communications of CCNx. The *controller* also transmits control signal to the AR.Drone and it receives status information which is necessary for autonomous mobile control from the AR.Drone. Moreover, a GPS receiver (GT-740FL) is used as self-localization of the AR.Drone.

B. Virtual Proxy for Movement Control

We propose *Virtual Proxy Model*, which realizes *interfaces* for movement control, which makes routing tables (FIB and PIT) associate with autonomous mobile control. Fig. 10 shows a process overview of *Virtual Proxy Model*.

In *Virtual Proxy Model*, virtual proxies are set up on an FR locally just as many as *intra-region networks* in disjoint networks. One proxy relays traffic from/to only one gateway of each *intra-region network* exclusively. Each proxy refers to

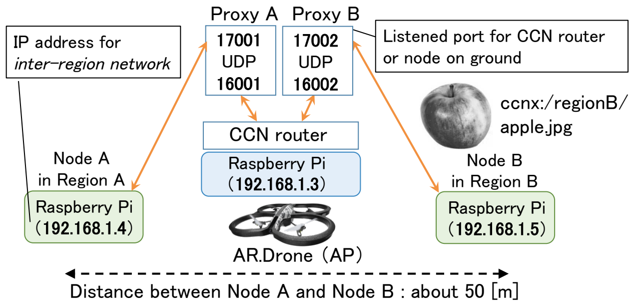

Fig. 11. Intra-Region Network Used in Our Experiment

only a source of a packet (i.e., where the packet is forwarded from). If the source is its own CCN router, the proxy forwards the packet to the gateway in an *intra-region network* which the proxy is in charge of. Otherwise, the proxy forwards the packet to its own CCN router. In case that the proxy forwards the packet to a gateway, the proxy makes the FR move to a location of the gateway if the FR is not connected with it.

An FR processes autonomous mobile control whose input is a goal position by combining self-localization and controlling body movement. The algorithm is simple to keep going straight on goal direction. An FR retrieves current location from a GPS and current direction from an AR.Drone.

CCNx uses pairs of IP address and port number as interfaces in FIB and PIT, thereby we implement *Virtual Proxy Model* using port number as an identifier of a proxy. Moreover, UDP is used for transport-layer protocol.

VI. Experiments

We confirm whether routing in an *inter-region network* is performed correctly with the prototype in disjoint networks.

A. Conditions

We carried out an experiment that retrieves unretrievable content between disjoint networks in vacant land on Suita campus of Osaka University to confirm whether content routing in an *inter-region network* is correctly performed. Specific conditions of the experiment are denoted below.

Fig. 11 shows an *intra-region network* used in the experiment. There are two regions, named Region A and Region B, which are unable to interconnect directly. Node A is in Region A and Node B is in Region B. In the experiment, we validate possibility of communications between Node A and Node B by using an FR and normal CCN methods. `ccnputfile` and `ccngetfile`, which are offered application by CCNx, are used to publish and retrieve content. Moreover image file (apple.jpg, 4 kB) is used as content. Two Raspberry Pis are used as Node A and Node B, and FIB entries of all nodes (including an FR) are set as same as ones of Fig. 10 manually.

B. Methods

In preparation, `ccnputfile` is executed on Node B to publish apple.jpg, named `ccnx:/regionB/apple.jpg`. After that, `ccngetfile` is executed on Node A to get content named `ccnx:/regionB/apple.jpg`. The constant numbers in delivering mode are set as $T_d = 1$ [s], $N_d = 1$. In the experiment, We confirms that apple.jpg is displayed on screen of Node A via routing in the *inter-region network*.

C. Results

We executed `ccngetfile` after confirming that Node A and Node B are unconnected by executing ping. As a result, the FR took off and moved to Region B, then moved back to Region A. Finally, screen of Node A showed apple.jpg. The results indicate that it was successful in routing in the *inter-region network* and retrieving unretrievable content between the disjoint networks with the FR.

VII. Conclusion

This paper focused on high flexibility of ICN routing and we consider to incorporate physical movement of routers into CCN routing. This paper proposed a Flying Router (FR), which combines a UAV and a CCN router. An FR makes it possible to retrieve unretrievable content between disjoint networks. Moreover, this paper indicates that there are more advantages of CCN FR than IP one. This paper also designs communication architecture where multiple FRs are able to process routing cooperatively. In addition, we carried out a basic experiment using a prototype of an FR in disjoint networks to validate feasibility of the proposed architecture.

In future, we will consider the proposed network furthermore, and we also implement crawling mode, waiting mode and *signaling* for decision of delivering FRs. We will also validate feasibility of retrieving content using multiple FRs in disjoint networks practically.

References

[1] V. Jacobson, D. K. Smetters, Thornton, and et al., “Networking named content,” in *Proceedings of 5th international conference on Emerging networking experiments and technologies*, Dec. 2009.

[2] N. Fotiou, P. Nikander, and et al., “Developing information networking further: From PSIRP to PURSUIT,” in *Broadband Communications, Networks, and Systems*. Springer Berlin Heidelberg, Oct. 2012.

[3] C. Dannewitz, “Netinf: An information-centric design for the future internet,” in *Proceedings of 3rd GI/ITG KuVS Workshop on The Future Internet*, May 2009.

[4] V. Jacobson, D. K. Smetters, N. H. Briggs, and et al., “VoCCN: Voice-over Content-Centric Networks,” in *Proceedings of the 2009 Workshop on Re-architecting the Internet*, Dec. 2009.

[5] Y. Liu, J. Geurts, J.-C. Point, and et al., “Dynamic adaptive streaming over CCN: a caching and overhead analysis,” in *Proceedings of 2013 IEEE International Conference on Communications*, Jun 2013.

[6] N. Uchida, N. Kawamura, T. Ishida, and et al., “Resilient Network with Autonomous Flight Wireless Nodes based on Delay Tolerant Networks,” *IT Convergence Practice*, vol. 2, no. 3, Sep 2014.

[7] K. Kim, S. Choi, S. Kim, and et al., “A push-enabling scheme for live streaming system in content-centric networking,” in *Proceedings of the 2013 workshop on Student workshop*, Dec. 2013.

[8] J. François, T. Cholez, and et al., “CCN Traffic Optimization for IoT,” in *The 4th International Conference on Network of the Future*, Oct. 2013.

[9] PARC, “CCNx,” Oct. 2014. [Online]. Available: http://www.ccnx.org/

[10] R. P. Foundation, “Raspberry Pi B+,” Oct. 2014. [Online]. Available: http://www.raspberrypi.org/

[11] Parrot, “AR.Drone 2.0 Power Edition,” Oct. 2014. [Online]. Available: http://ardrone2.parrot.com/

[12] E. Navi, “ARDroneForP5,” Oct. 2014. [Online]. Available: http://kougaku-navi.net/ARDroneForP5/