Master's Thesis

Title

Design and Implementation of Tracking System for Moving Objects by Realizing Dynamic Service in Information-Centric Networking

Supervisor

Professor Masayuki Murata

Author

Tatsuya Tanaka

February 9th, 2018

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Design and Implementation of Tracking System for Moving Objects
by Realizing Dynamic Service in Information-Centric Networking

Tatsuya Tanaka

Abstract

Information-centric Networking (ICN) is an approach to network architectures of the future. Unlike existing IP address based communication, ICN communicates using information (content) as an identifier. From its design concept, ICN attracts attention as the communication infrastructure for the Internet of Things (IoT) environment that requires various device cooperation and the mobility environment where an address frequently changes. Also, in network research, it is desirable to escape from dumb pipe, so the network needs to have service provision functions. By using "control", "management", and "data processing" in the network, ICN is expected not only as a simple data transfer but also an information distribution infrastructure with service provision function. However, most of the research on ICN focuses on accessing static content with a name, and the ICN architecture that provides dynamic services has not been sufficiently researched. In order to develop network functions that realizes dynamic services in ICN, we design and implement the tracking system for moving objects with ICN. In this system, the moving object is a car, and the image of the road camera shooting the car which passes a large number of cameras at high speed is distributed. By providing the tracking function of moving objects by the network, we realize the service that seamlessly switches road cameras and distributes the image. Also, as an extension of the system, focusing on in-network processing in ICN, we design a system that performs data processing such as splitting/merging images from multiple cameras in the network according to car position. We implement the designed system and conduct operation experiments using actual equipment. As a result, we confirmed that ICN can realize a dynamic video distribution service that tracks a moving object based on the designed system and ICN has the possibility as an information distribution infrastructure with dynamic service provision function. Also, for deploying the system in a real scale, we analyzed the issues in detail.

Keywords

ICN (Information-centric Networking)

NDN (Named Data Networking)

Tracking

ITS (Intelligent Transport Systems)

In-Network Processing

Dynamic Service

Contents

List of Figures

List of Tables

1 Introduction

IP, which is a location-oriented communication protocol, became widespread since it was originally intended to perform host-to-host communications when the Internet was first emerging, and is still being used deeply. Over time, however, network usage patterns have changed from a location-oriented style to a content-oriented one. Such services as the Web and video distribution, users are not interested in the nodes with which they communicate, but only in the content delivered by these services (web pages, movies, music, etc.). Therefore, due to the divergence between current network usage and network architecture, problems such as inefficiency of communication have arisen. Furthermore, in recent years, the amount of traffic on networks has increasing drastically due to the popularity of smartphones and IoT (Internet of Things) and enlargement in the capacities of contents, but ordinary IP protocols have limited mechanisms for handling efficiently. The Content Delivery network (CDN) as a technology used to construct an optimal network for delivering content such as Web applications and streaming media, but in cannot deal with traffic on an entire network because it is implemented at the application layer, incurs high cost, and it is not an essential solution to the problem of the network.

ICN (Information-Centric Networking) attracts attention as a new generation network replacing the IP network. ICN is a designed as content-oriented network architecture and has many advantages not found in IP because it eliminates the current dissociation between network usage and network architectures. For example, it is expected to solve existing IP problems such as increase in load of servers and relays nodes, multicast, mobility and so on integratedly. In particular, a control mechanism based on the name has gained a considerable amount of attention in recent years. Not only simple transport, but also flexible processing including control and management is possible, so ICN is expected to become the communication infrastructure of the IoT, which involves cooperation between and the configuration of various devices. Moreover, in recent network research, emphasis is placed on removing from dumb pipe, so the network itself is required to provide flexible services. ICN is expected as an information distribution infrastructure the realizes new services because of advantages such as control and management that can handle IoT, naming service for resources, ease of programming, and so on.

However, most of the research on ICN focuses on accessing static content with a name, and research on the dynamic service configuration and provision is no sufficiently done. For realization and technology deployment of ICN, it is necessary to study network architecture which realizes dynamic service configuration. Therefore, for the purpose of research on network architecture realizing dynamic service, we design and implement the Tracking system for moving objects that acquires information by tracking the car which is a high-speed object. As a specific application, we track the moving car on the network and switch the road camera shooting the car seamlessly deliver the video. Furthermore, as an extension of the system, focusing on in-network processing in ICN, we realize data processing such as splitting/merging images from multiple Cameras according to car position. We demonstrate that ICN can provide dynamic services by implementing using the real machines and performing basic experiments. Based on the experimental results, we analyze the effectiveness of the designed system and the issues for practical application in detail.

2 Related Work

2.1 The Emergence of ICN/NDN Technology

Many years have passed since the Internet became widespread, and IP is persistently used as a communication protocol even now. The Internet was originally designed for the use or acquisition or sharing of distant resources. For this reason, a communication model based on communication between two hosts has been developed, and, IP, which is a location-oriented communication protocol has become spread. However, nowadays, network usage patterns have changed from a location-oriented style to a content-oriented one focusing on the contents, and communication is carried out to obtain contents such as web pages, music, video files, etc. on the Internet. For this reason, there is a deviation between the network usage and the network architecture, and various problems have arisen. One of the most remarkable examples is the increase in processing loads on servers and relay nodes. In recent years, the amount of traffic on networks has increased drastically due to the spread of smartphones and the increase in the capacity of contents, but ordinary IP communication has limited mechanisms for handling these problems efficiently. Therefore, ICN attracts attention as a new network infrastructure that efficiently handles enormous contents.

ICN is designed as a content-oriented network architecture and is expected as a new future network because it eliminates the divergence in network usage and architecture. Unlike existing location-oriented architectures such as IP networks, routing is controlled by content name rather than node address in ICN. NDN (Named Data Networking) and Content-Centric Networking (CCN) [1] are well known as active ICN projects. In this research, we take up NDN of which implementation is published as open source.

In NDN, communication is then implemented by exchanging a request message with the name and content with the corresponding name. The content request message is referred to as Interest and the content message as Data. Interest is generated and communication is started from the content requester. Interest sent out on the network arrives at the interface called Face of the router which is a relay node. At this time, the router memorizes from which Face Interest has arrived. When Interest arrives at the router, a Face to be forwarded is determined based on the name of the content and is forwarded. When Interest arrives at the content holder, Data is returned. Data contains name and content data, it is carried in a way that Interest traces back, and reaches the content requester. In this way, unlike routing using the host and interface IP addresses on the

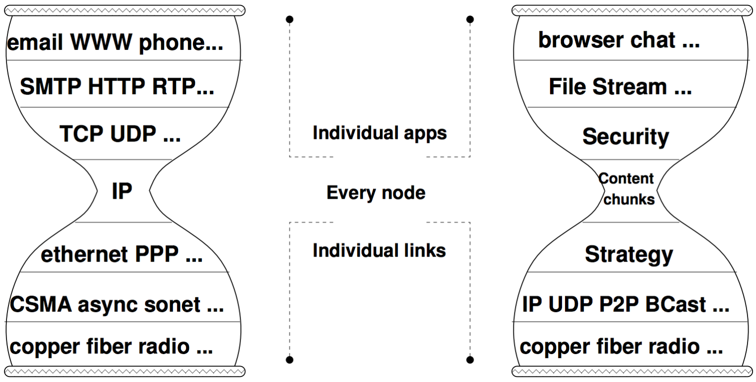


Figure 1: Protocol stacks of IP networking and NDN (cited from [1])

Internet, Interest is routed only by name, and Data is returned based on the hop information. In addition, as a mechanism for handling the contents efficiently, NDN caches the contents in relay nodes. While Data is cached. When the corresponding Interest arrives, the router can reduce the load on the network by sending the cached data back.

Figure 1 shows protocol stacks of IP networking and NDN (cited from [1]). NDN constitutes a thin-waist stack as in IP networking, but a strategy layer is also added to NDN, different from the protocol stack of IP networking. The strategy layer in NDN determines how to transfer Interest and Data. In IP networks, messages are transferred according to routing tables constructed by routing protocols; but in NDN, in addition to this routing processing, the strategy layer defines when and where to forward the message, and what kind of processing should intervene while forwarding messages.

The forwarder of NDN has the following five main tables to realize the above communication using Interest and Data:

FIB (Forwarding Information Base)

The FIB is a table that determines the forwarding destination Face of Interest received from the downstream. The node receiving Interest transfers Interest upstream with reference to the FIB.

PIT (Pending Interest Table)

The PIT is a table that stores information concerning Faces where Interest is received and sent. A node receiving Data returns it to the requester by referring to the PIT.

CS (Content Store)

The CS is a table that stores caches of Data. A node receiving Interest returns the Data to the requester when the corresponding entry is in the CS.

RIB (Routing Information Base)

The RIB is a table that stores routes. The RIB is updated manually or by routing protocols. Entries of the FIB are generated from those of the RIB.

Strategy Choice Table

The Strategy Choice Table is a table that stores strategies corresponding to each item of the content. A node receiving Interest or Data applies a strategy to content by referring to the Strategy Choice Table.

The names of contents are used for storing and referring to these tables.The PIT and the CS use complete content names, whereas the FIB, the RIB, and the Strategy Choice Table use the prefixes of content names. On the other hand, at the end nodes, to implement the exchange of Interest and Data, the prefix of the content name and the listener (application) are registered in advance to the Face managed by the end nodes. Then, when a message matching the prefix of the registered name arrives, the callback function defined in the listener is called for each type (Interest or Data), so that the application can receive the forwarded message.

What is further expected for ICN is improvement of system efficiency considering service and improvement of design efficiency of application software using the network. In the location-oriented network, the application development by programmers becomes complicated by the divergence in network usage and architecture. On the other hand, ICN's content-oriented network offers flexible content acquisition including seamless routing and node processing, so there are merits from programmers' perspective that application development is easier. It is meaningful for ICN to improve the ease of application development in the situation where IoT devices are increasing and creating systems required real-time processing of a large amount of data generated.

Therefore, in this research, we design and implement the tracking system for moving objects as an implementation model showing that ICN technology leads to improvement in ease of creating an application. In this system, a car passing multiple cameras, one after another at high speed is tracked, and a road camera which is taking pictures of the car is switched to acquire an image. When realizing such an application on an IP network, the client side requesting video needs the following processing.

1. Establishing communication with a car that is mobile node
2. Decision of camera to acquire from information on car position
3. Acquisition of the IP address of a road camera
4. Establishing communication with a road camera

Processing that must be performed on the client side is complicated because of location-oriented communication. Furthermore, it is difficult to switch camera in response to high-speed movement of the car. All the camera data and car position information are gathered in the cloud and the client can request the image shooting the car towards the cloud, but there are problems that it cannot cope with an increase in the amount of data traffic circulating in the network and real time processing. On the other hand, in ICN, by tracking the position of the car on the network, Interest is routed the appropriate road camera. Therefore, it can seamlessly realize the above processing and improve design efficiency. In this way, if ICN can provide a network infrastructure that facilitates the development of new applications and services, it is thought that the application field of ICN will be expanded further.

2.2 In-Network Processing enabling Flexible Information Acquisition

In recent years, IoT and M2M (Machine to Machine) technology has come up, and it is supposed that devices of orders of magnitude will be connected to the network in the future, and mutual communication will occur and a large amount of traffic will be generated. In the network mainly handling IoT environments, individual nodes are extremely restricted from the viewpoint of computational resources, because many nodes are composed of ultra compact and simple devices such as sensors and tags. Therefore, it is not easy to make each node take charge of intelligent processing. Generally, sensor data are collected and processed in the cloud, but application of edge computing is considered as concern about an increase in the amount of data traffic circulating in the network. Edge computing is expected to reduce the amount of data flowing into the network with the idea of processing data at the end of the network where sensors and others exist, that is, at the network edge. In addition, it is possible to reduce the response time, since users on the edge can obtain processed data without going through the cloud.

However, it is not practical in terms of cost to allocate computing resources such as servers to all edges of the network. Therefore, a method in which nodes in the network intervening between "edge" where objects exist and "core" where clouds exist process data in a distributed manner distributed data can be considered. Adopting this structure realizes in-network processing that reduces the number of nodes that provide processing functions and reduces the amount of data traffic flowing into the network.

In recent years, many studies have utilized in-network processing in ICN. In ICN, forwarding and in-network processing can be performed seamlessly by expressing contents to be controlled and objects to be controlled such as end nodes and relay nodes with a flexible name, and we can hence yield benefits not found in location-oriented communication architectures. For example, considering the case of specifying and controlling a certain device, in a location-oriented communication protocol such as IP, a conversion table that maps from the device name to the device address is required in the application layer. This is a disadvantage because the development of the application becomes complicated in spite of the demand for simply controlling devices. By utilizing in-network processing, ICN can provide programmability on the network and easily realize communication that can integrally execute flexible information acquisition, such as device cooperation in IoT and M2M or dynamically changing content according to specified parameters.

NDN-RTC [2] is an instance of the control of ICN nodes. NDN-RTC is an application that performs real-time video conference on NDN. By specifying the type of media such as camera and sound and the quality of video in the case of a camera image as the name of Interest, it is possible to change the settings such as the resolution and bit rate of the contents of video and audio on every chunk. An example of the name is shown in Table 1. In IP communication, when changing a setting like this, it is necessary to separately transmit an IP packet with a message indicating the setting. This can cause configuration lag. In ICN, however, it is possible to designate the content request and its quality at the same time in chunk units, so that it can seamlessly retrieve a stream corresponding to the name.

Moreover, applicable control in IoT environments has also been studied in recent years. In [3], Smart Home which is a representative scenario of IoT is designed with ICN. In the proposed NDOMUS (Named Data netwOrking for sMart home aUtomaton System), a namespace that supports sensing and execution by IoT equipment, managing equipment and setting operations is defined. As examples of such physical control, the lighting of the kitchen and sensing of the temperature

Table 1: An example of the control name in NDN-RTC

Name
/<root>/ndnrtc/user/<producer-id>/streams/ cam/mid/frames/delta/1/data/01

Table 2: An example of the name for for controlling lighting in NDOMUS

Name
/bobHouse/task/action/light/on/kitchen
/bobHouse/task/action/light/off/kitchen

of the bedroom were cited. Examples of the name representing control information are shown in Table 2. In the context of IoT environments, the effectiveness of ICN is drawing attention not only in data control and signaling but also in physical control, such as temperature adjustment of air conditioners, changing the angles of surveillance cameras, and so on.

Although the above is a study on physical control, NFN (Named-Function Networking) [4] is proposed as an architecture that realizes data processing in the network with ICN. In this architecture, CCN/NDN architecture is extended and a framework for router to provide data processing is proposed. By applying name-based searches to not only data but also processing functions, the user can acquire desired processed data by designating the name of desired data and processing function. The NFD management protocol [5] in NFD (NDN Forwarding Daemon) [6] which is the forwarding daemon of NDN can control the relay node. The NFD can control configurable components such as Face, RIB, Strategy Choice Table, etc. by various managers. NFD Management Protocol is a protocol supporting communication with various managers using Interest/Data.

In this research, in order to show that in-network processing in ICN is effective for providing flexible and dynamic service, we consider utilization of in-network processing for the system to be implemented. Specifically, in the communication between the client requesting video and the camera node, the relay node acquires the image from the multiple nodes, and implements the process of processing the images and returning the data. If the network side can change image according to the state of the car and ICN can provide a communication infrastructure capable of flexible information acquisition without being bound by IP address, the application field of ICN is expected to expand.

2.3 The Network required for ITS

In the application field of IoT, the research in the field of transportation has been actively conducted in recent years. A system for solving the problems such as traffic accidents, congestion and environmental measures carried out by people with transportation, is called ITS (Intelligent Transport Systems), and research and development have proceeded toward realization of ITS. Vehicle-to-vehicle communication and road-to-vehicle communication, especially for safe driving support, are expected as an indispensable technology for future ITS. However, since the connectivity of the vehicle network is intermittent, there are problems different from ordinary networks. In [7], the characteristics of the vehicle network are investigated, but in such a dynamic environment existing IP network does not work well was pointed out. On the other hand, ICN is suitable for a vehicle network where IP address is unnecessary and node configuration is frequently changed, and research for realizing ITS using ICN is being conducted. In the following, we introduce ITS research using ICN.

In recent years, applications using the VANET (Vehicular Ad-hoc NETwork) are being developed, but the network architecture is not fixed, there are problems in the IP network and ICN for VANET is considered. In [8], V-NDN applying NDN to VANET is designed and proposed. As an application scenario, “traffic information” and “road photograph” were acquired from vehicles and RSU (Road Side Unit), and simulation experiments were conducted. In [9], Navigo is proposed, which is implemented packet transfer mechanism based on location information.

In this research, of the IoT expected as ICN use case, we pay attention to the fact that research on ITS is prosperous, the target of tracking system for moving objects as the dynamic service is a vehicle. [10] proposes a networked vision system for tracking high-speed objects traveling across multiple cameras. An example of the application of the system is to continuously track the vehicles from the entrance to the exit while never losing the targets. Such a system should be useful for surveillance and analysis of traffic congestion and accident. It developed a prototype system that captures the moving object with more than 700 fps and that shares the small feature data only between a pair of adjacent cameras. An object moving across two cameras at 7000 mm/s 25 km/h was successfully tracked by the experimental system. However, the data circulating through the network in this system are about 64 bytes of data, such as the time stamp and the position of the tracking target, and it doesn’t taken into consideration until video distribution. In addition, it is only information sharing between cameras, and it is not able to provide a service for tracking moving objects as a network. Therefore, in this research, the network itself tracks mobile objects and designs an architecture that can be utilized not only for road camera video distribution but also for other ITS applications. We aim to provide a dynamic service that delivers video using the designed architecture, and design and implement it making full use of the advantages of ICN.

3 Overview of Tracking System for Moving Objects by Realizing Dynamic Service

The purpose of our research is to realize dynamic service in ICN by designing and implementing Tracking System for moving objects that tracks moving objects and acquires its information. In this thesis, the target of this system is the car and information is an image of a road camera. A road camera is determined and routed on the network side, according to the current position of the car, and an image is acquired. This chapter shows the overview of the designed Tracking System for moving objects and design policy.

3.1 Overview of System

In this research, we design and implement a system that tracks a car on the network and returns images cars traveling across the field of view of multiple road cameras. An application example of this system is one that continuously tracks the vehicles from the entrance to the exit while never losing the targets. Such a system is thought to be useful for road surveillance, analysis of congestion, investigation on the causes of accidents, etc. Figure 2 shows an overview of the Tracking System for moving objects to be implemented. The car (CarA) sends messages to the network at regular intervals (Send Register Message), so the network can track the position of the car. The consumer sends Interest requesting the road camera image of the current location of CarA (Send Interest). The network receives Interest and determines the road camera that is shooting the current location of CarA from the car information from received from Register Message and the specified car included in Interest. Upon receipt of Interest, the road camera returns the image as Data (Send Data).

In this research, the object of Data to be acquired is taken as the image of the road camera, but this is one of scenarios, and we aim to design an architecture that can acquire other contents relating to the car (for example, sensor data or routing information of the car) by the tracking system. That is, if the content name requested by the user is “road camera”, the camera image is returned by routing the corresponding road camera, if the content name is “location information”, the current location information acquired by tracking the car is returned. The goal is to provide such a network infrastructure that carries out from determination of Interest and return of Data.

The configuration nodes of this system are shown below.

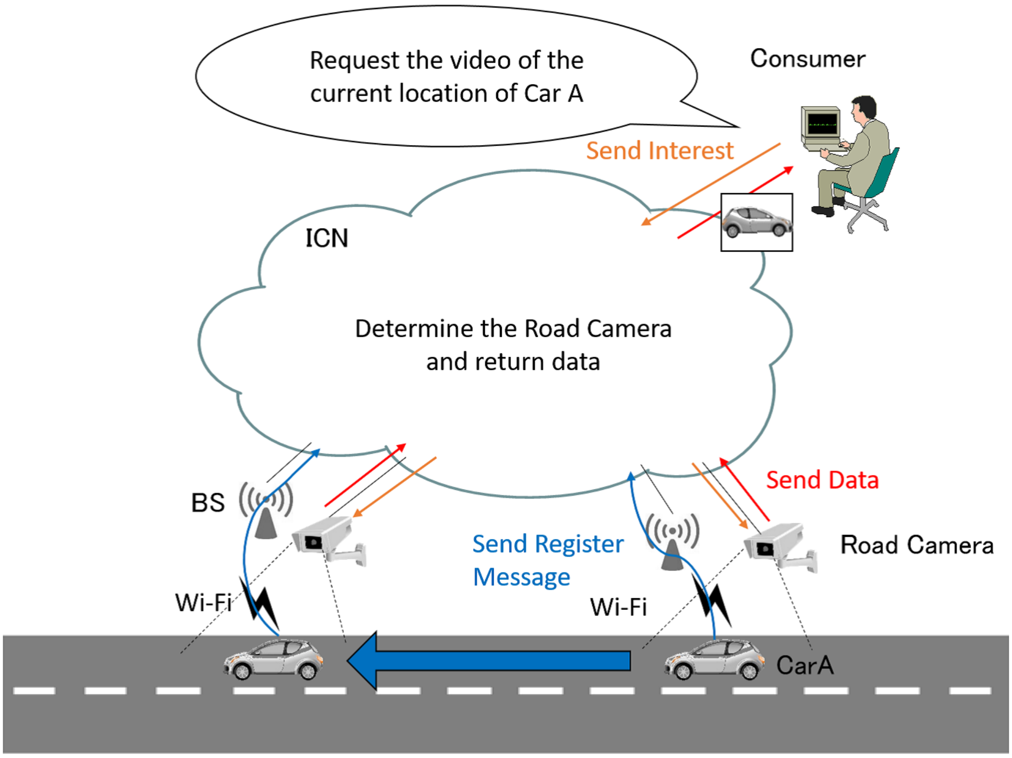


Figure 2: Overview of Tracking System

Car

This is a car to be tracked. The car performs network communication via a wireless access point set along a road network.

Road Camera

This is a camera set along the road network. The camera continues to capture the moving image data, and when there is a request for camera data from the network, the corresponding image is returned as Data. It is assumed that the camera is set so that the communication area of the access point coincides with the shooting range of the road camera, taking into consideration the shape of the road network.

Wireless Access Point

This is a base station that connects wireless communication from cars. Through the wireless access point, the car notifies the network of the location of the car.

Router

This is a relay node that processes packets forwarded.

Consumer

This is the user who acquires the tracking video of the car. What is assumed as a mode of use is the use of the driver as a drive recorder, police, emergency, monitoring, and the like.

3.2 Design Policy

In this thesis, we design Car-Tracking system using ICN. The advantage of ICN is that users do not have to worry about the location of contents, designate " information" that they want to acquire, and the network processes them to deliver content. It is desirable to realize dynamic service by taking advantage of this. Therefore, in this thesis, we focus on images of the road camera as one of the information of cars to be acquired, create the service which switch the camera set in the road side according to the movement of the car and continue to acquire video frames. And the network for executing the application in the form of extending the NDN. For implementation, we use NDN [11], one of the projects of ICN, ndn-cxx [12] which is NDN's C++ library, NFD [6] which is a forwarding daemon of NDN.

If we attempt to realize this system with address-based IP, it is difficult to respond quickly to cars moving at high speed and efficiently distribute data. This is because that when switching road cameras in accordance with the movement of the car, it is necessary to establish communication with the car, obtain the location of the car, determine the road camera to communicate with and designate the IP address of the road camera. Further, there is a disadvantage that application development becomes complicated. In the ICN, simply designating the car to be tracked by name enables processing on the network side according to the movement of the car. Therefore, it is possible to realize efficient data distribution corresponding to the mobile object.

Also, in recent years, a service enabler that provides advanced services in the network is required.many of the ICN's studies so far have been approached to accessing static content, but what is actually needed is the dynamic configuration/provision of services, and this realization is a research topic of ICN. In this system, the network tracks the moving car, and provides the dynamic service that seamlessly switches cameras and delivers the images according to the movement of the car. By realizing this system as a use case of ICN, we show that ICN is the multimedia communication network capable of realizing advanced dynamic services combined with control, management, etc. as well as transport.

In addition, when designing, it is also a design policy to pay attention to the following points:

In-Network Processing

In recent years, moves to utilize in-network processing in ICN are becoming popular. In ICN, flexible control and processing utilizing names can be incorporated into the network, and forwarding and in-network processing can be performed seamlessly, so there is an advantage that is not in the location-oriented architecture. For example, in image acquisition, setting the quality of contents in the name field of Interest makes it possible to change such settings as the resolution and the bit rate of the contents of video and audio on every chunk. Also, in other scenarios, it may be considered to execute processing such as collection, processing. and aggregation of sensor information on the moving path of the car. Therefore, in this thesis, we focus the fact that ICN can incorporate in-network processing, and design the infrastructure on which these controls can be realized. By incorporating in-network processing, it is possible to provide a more flexible service that was difficult to realize in conventional networks, and it is also possible to discuss the application of ICN to IoT and M2M communication in general.

Mobility support

Improvement of mobility support is mentioned as an advantage of ICN. In a location-oriented network such as IP, there are problems in terms of managing the geographical change of nodes, complicating the existing protocol. In contrast, ICN is apart from location-oriented, and it is said to be able to handle mobile nodes efficiently. In particular, NDN natively supports consumer mobility since it returns data by following the route of Interest sent by a consumer who is a content requester. On the other hand, as for producer mobility, it is necessary to have a mechanism to efficiently deliver Interest to mobile producer. Interest is forwarded with reference to the FIB, so the FIB must be updated each time the mobile producer moves, but there is a problem that the update frequency increases. Therefore, research on source mobility in ICN is actively conducted. [13] is a survey paper on the producer mobility support of NDN, mainly citing the following as a method of rendezvousing Interest and Data.

- MP (Mobile Producer) Chasing

  This is the way to find out where an MP is by using an RV (rendezvous). They identified the following two solutions:

  - Mapping-Based Solutions [14–19]

    The MP reports to the RV its PoAs (Point of Attachments) through which its data can be retrieved. If the MP has relatively high mobility, the mapping update costs will be high.

  - Tracing-Based Solutions [19–23]

    The MP creates a " breadcrumb trail " that can be followed by interests from the RV to reach the MP. There is additional overhead on routers to keep the trace state.

- Data Rendezvous

  This is the way to ensure that the mobile-produced data can be easily found.

  - Data Depot [24]

    The data produced by MPs are moved to a known stationary server. However a data depot needs to be enhanced with either a mapping or a tracing solution to fetch data from mobile producers in order to support applications that require support for retrieving data on-demand.

  - Data Spot [8]

    Data is produced within, and available from, a stationary region. This solution has limited application that can be used.

[25] proposes a multicast based seamless mobility as a method to deal with producer mobility. In this scheme, by multicasting Interest to a PoA that may be handed off by a mobile producer, Interest can be received even after the producer has handed off. When sending Interest to the mobile node, it also receives information of PoA that may be hand off destination of the mobile

node, thereby multicasting Interest to the hand off destination PoA in addition to normal Interest sending. As a result, if the mobile producer is not moving, it is possible to get Data with normal Interest, and if the mobile producer hand off, to get Data with multicast Interest.

In this way, the research for realizing producer mobility in ICN is actively conducted, but each method has advantages and disadvantages, and the optimal method has not been determined. In [26], in real applications. It is pointed out that further research is needed to devise suitable mobility support technology that takes into consideration the real time requirements and the cost of the hardware.

In the system of this research, the network needs to provide mobility support in order to recognize the car location for road camera decision. In this thesis, we design a mobility solution for mobiles by using multicast based mobility support from the above because this solution can reduce the overhead from the control and forwarding planes and accommodate the real-time requirements.

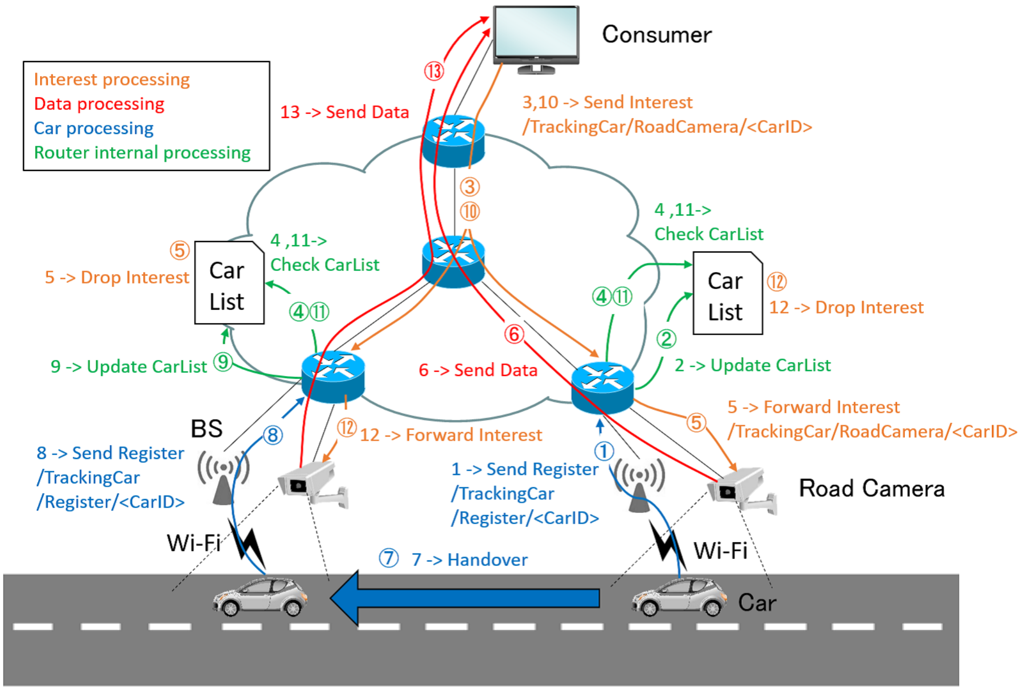


Figure 3: Communication procedure of Tracking System for moving objects

4 System Design

In this chapter, we describe the design of the tracking system for moving objects.

4.1 Communication Procedure

Communication procedure of tracking system in ICN is shown in Figure 3. The consumer sends an Interest containing the car ID in the name, and network forwards it to the multiple cameras. Also, the car notifies the position of the car to the network. The camera node that receives the Interest returns the image as the Data only when the car specified as the name of the Interest is in the shooting range. As a result, the tracking user can acquire images from the camera in which the designated car is photographed.

First, the process flow on the car side is shown below:

1. The car notifies the network of its own car ID via the neighbor access point sending the

Interest. (Send Register Message)

2. Register message is delivered to the camera node in charge of the area of the communication access point, and the camera node recognizes the car within the shooting range. (Update Car List)

3. The car repeats 1 at regular intervals.

The car need not recognize the existence of the tracking user, but only informs the network of its own car ID. The camera node knows the list of cars within the shooting range, and it is necessary to delete from the list after the car has moved. To do so, in this design, at the timing of movement, a message of a car deletion request is transmitted toward the previous camera node. As another method, the use of a timer is conceivable.

Next, the process flow on the consumer side is shown below:

1. The consumer sends the Interest containing the car ID of the car to be tracked in its name. (Send Interest)

2. The router multicasts the received Interest towards the upstream.

3. The camera that receives the Interest checks car list to see whether the car included in the name of Interest is within the shooting range. (Check Car List)

4. The only camera node with the car in Car List respond to the Interest and return images as the Data (Send Data). If there is no car in Car List, drop the Interest. (Drop Interest)

5. Transfer the Data referring to the PIT.

6. The consumer continues to acquire images by sending an Interest requesting the next frame.

In this thesis, we adopted the Multicast based mobility support proposed in [25] as a method to solve the mobility of cars. This is because of correspondence to the high mobility of cars. In the mapping based method described in the survey paper [13], the mapping server responds to mobility by holding the mapping between the mobile node and the current location of PoA, but in the view of the high mobility of cars, the cost of updating the mapping is so high. And, there is a problem of inefficiency of routing by accessing the mapping server. The tracking method described in [13] does not require PoA solution and can eliminate the demerit of the mapping based solution, but

Table 3: List of designed components

Component	tools	Role	Implemented by
Consumer	ndn-cxx	Generate and send Interest	User node
Producer	ndn-cxx	Generate and return Data	Camera node
Register	ndn-cxx	Send car register message	Car
Multicast Strategy	NFD	Multicast Interest	Router
TrackingCar Strategy	NFD	Determine appropriate camera	Camera node
CarList	NFD	Manage car information	Camera node

this also has a high maintenance cost of the trace state to cope with high speed movement like a car. On the other hand, in multicast based mobility support, there are problems such as the cost of copying an Interest and the existence of the dropped Interest. However, adaptability to cars moving at high speed is high, and it is suitable for acquiring images in real time. In addition, since the destination of the car can be predicted from the automobile network, it is also easy to expand such as specifying a range to be multicast.

4.2 Components

The communication described above can be realized by adding a unique data structure and strategies in addition to the consumer program and the producer program on the camera node, as well as the functions supported by the normal NDN. The data structures represent structures for storing data necessary for communication. Then, by the processing of our own defined strategy, we realize cooperation between network processing of tracking and NDN communication. The components that make up the system are shown in Table 3. The programs of Consumer and Producer that transmit and receive video are created by modifying chunks program which is a file transfer program prepared by ndn-tools [27]. The Register program is for the car to inform the network of its own car ID. Multicast Strategy which provided as the default strategy of NFD applies to routers. The TrackingCar Strategy is a strategy for recognizing the movement information of cars and performing processing for selecting the road camera. This new strategy applies to the NFD of the camera node. CarList is a newly defined data structure not found in normal NDN and prepared in NFD of camera nodes. This holds the ID of the car existing within the shooting range of the

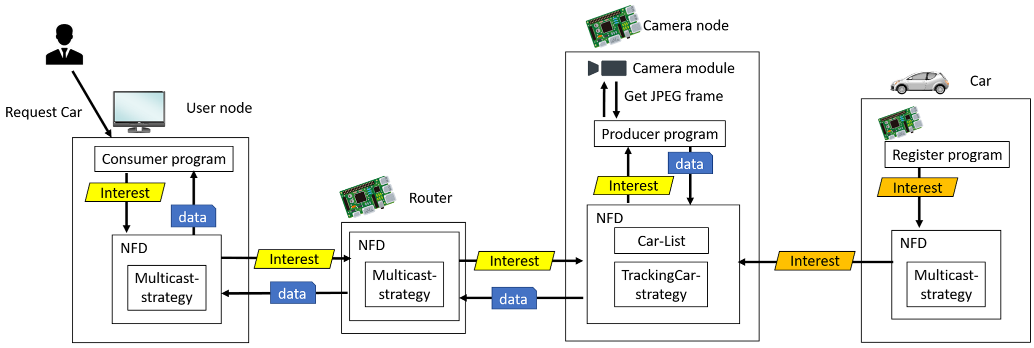


Figure 4: Component diagram of tracking system

camera and is prepared in NFD of camera nodes. This is accessed and updated by the function called from TrackingCar Strategy.

Figure 4 shows the component diagram of tracking system.

Detail of each component are described below.

4.2.1 Custom Table

In order to build a network for tracking cars, we added the following table to the node not found in NDN.

Car List

Car List stores the ID of car existing within the shooting range in each camera node. In the implementation, the ID is represented by the string type, but and ID that can uniquely identify the car, such as the number plate of the car or the registration ID of the application is used. This table is only accessed and updated from TrackingCar Strategy.

4.2.2 Custom Strategy

A strategy in NDN can be set for each content. A strategy in NDN can define further strategies to determine when and where to transfer Interest and Data, and what kind of processing to intervene in at the NDN node. It can perform various controls at the relay node by customizing the trigger method to be called when Interest or Data is received. In addition, NDN can freely determine strategies to apply by namespace using the Strategy Choice Table. It is also possible to realize various controls using the same route by changing only the namespace. In this system, as a strategy, TrackingCar Strategy is implemented to realize car tracking. With this strategy, it is possible to appropriately select the road camera and deliver the video. Also, by changing this strategy here, it is possible to call up other applications. In addition, we prepare the Multicast Strategy for multicast processing of the router.

TrackingCar Strategy

The TrackingCar Strategy is used to recognize the present location of the car and judge the Interest from the consumer according to the car condition. It is implemented in each camera node. In this implementation, it targets messages whose namespaces are "`/TrackingCar`" and provides processes shown in Table 4.

Table 4: The processes provided by TrackingCar Strategy

Name	Type	Process
`/TrackingCar/Register/{CarID}/...`	Interest	The processing of car registration is performed. It registers the CarID specified in the Interest name in CarList.
`/TrackingCar/RoadCamera/{CarID}/...`	Interest	The processing of the request of images of the road camera shooting the car is performed. It checks whether the CarID specified in the Interest name is registered in CarList, and transfers the Interest to application if registered. If not registered, drop the Interest. However, if the 1st segment of each frame was transferred, until the transfer of the frame is completed, all Interests that request the frame are transferred even when the car state changes.

Multicast Strategy

The Multicast Strategy is used to multicast Interest toward all upstream faces. It is implemented in the router. This strategy is prepared as the default strategy of NFD. In this implementation, it targets messages whose namespace is "`/TrackingCar`".

4.2.3 NDN Applications

The application program to be executed at each node will be described below.

Consumer Program

The Consumer program is used to request the tracking video. It is designed by modifying the file transfer program "chunks" prepared by NDN's essential tools.

The processing procedure is described below. The user creates and send the Interest by inputting the requested content name (ex: `/TrackingCar/RoadCamera/CarA/1`) from the command line. When the Producer program described later responds to the Interest, the Data of the first segment acquired by segmenting the JPEG frame is returned. When the consumer receives it, it confirms the number of segments of the contents from the metadata, and generated and send the Interest requesting the next segment. Upon receiving all the segments, reception of one JPEG frame is completed and it is stored in its own node.

Producer Program

The Producer program is used to create and distributed the video. It is designed by modifying the file transfer program "chunks" prepared by NDN essential tools.

The processing procedure is described below. On the camera node, after starting NFD, start the Producer program by specifying the content prefix name (`/TrackingCar/RoadCamera`). As a result, the FIB is updated so that Interest having this prefix can be received. When receiving the Interest (ex: `/TrackingCar/RoadCamera/CarA/1`), it acquires a JPEG frame from the camera. In the NDN, since the max packet size that can be sent in one packet transfer is determined, the JPEG frame is segmented for each size (4400 bytes) that can be sent in one packet,

Table 5: Name Structure

Name
/TrackingCar/<Information>/<CarID>/<Versioning&Segmentaion>

and the first segment is returned as the Data. At this time, the number of segments of one frame is added as metadata. Upon receiving the Interest requesting the next segment from the consumer. The corresponding segment is returned as the Data. When transmission to the last segment is completed, data transmission of one frame is completed.

Register Program

The Register program is used to send a registration message from cars. In the real environment, it is assumed that a message including the carID is transmitted to the nearest access point. In this thesis, we are running on the overlay network of NDN and it is the operation in the same Wi-Fi, so by including the camera of the current location in the Interest name, cars notify the current location to the network.

The processing procedure is described below. We execute the program specifying the Car ID (ex: CarA) of the argument. When entering the input wait state, and entering the camera ID (ex: CameraA) , Interest (`/TrackingCar/Register/CarA/CameraA`) is generated and transmitted. The Interest is forwarded to the camera nodes by the NFD, and the each CarList is updated by the strategy of the NFD of the camera nodes.

4.3 Name structure

In this thesis, the user requests the content "camera video shooting the current location of the car". This camera video is one of the scenarios, and it is desirable that it can be applied in a tracking system so as to be able to acquire other contents related to the car (for example, sensor data of the car, route information, etc.) . That is, based on the content name, the network judges the Interest and executes route control and data processing. In this way, we consider a name structure that can be applied to acquire information other than the road camera video.

The name structure of the Interest requested by the tracking user in this thesis is shown in the Table 5.

Table 6: Examples of car information

Information	Request Content
RoadCamera	Road camera image shooting a car
CarCamera	On-vehicle camera image
Route	Route information to a vehicle
Location	Position information of a car
Speed	Speed information of a car

TrackingCar

By using this prefix, the strategy for tracking system for cars is applied in NFD.

⟨Information⟩

⟨Information⟩ represents the type of contents to acquire. In this thesis, a user specifies RoadCamera here, so that he requests the camera image on the road camera shooting the car. As an application, examples of name that can be specified as Information is shown in Table 6.

⟨CarID⟩

⟨CarID⟩ represents the ID of the car to be tracked. What is used as the ID here can uniquely identify the car, such as the number plate or the number assigned to the car registered in the application.

⟨Versioning & Segmentation⟩

This is version information or segment number of data included in the content. For example, in the case of a camera image, it is necessary to describe a sequence number and a segment number for a certain frame.

In this thesis, the consumer specifies the frame number. When the producer receives the Interest, it adds the version information and the segment number to the name from the current time and return the Data. When the consumer receives the Data, it creates the Interest requesting the next segment by including the version information and the next segment number in the name.

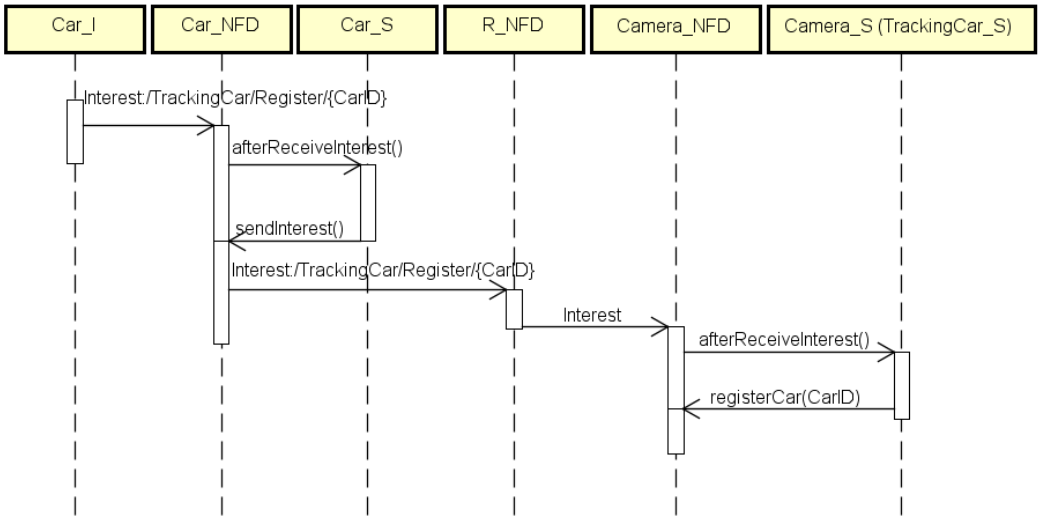


Figure 5: A sequence diagram of Registering Phase

4.4 Processing Sequence

In this section, we show the procedure for communicating of tracking system using our own components introduced in Section .

4.4.1 Registering a car in the network (Registering Phase)

In this system, in order for the network to recognize the position of the car, the car sends a message notifying the car ID to the network. Here we call it Registering Phase. Figure 5 shows a sequence diagram of Registering Phase. In this Figure, Car_NFD is the NFD of a car node, Car_S is the strategy of a car node, R_NFD is the NFD of a router, Camera_NFD is the NFD of a camera node, and Camera_S is the strategy of camera node.

Communication is started by creating an Interest (`/TrackingCar/Register/{CarID}`) whose name contains its car ID from the interface of the car node. This process is executed by the Register program described in Section 4.2. The NFD of the car node receives the created Interest and determines the forwarding destination via the strategy. In this case, the strategy set by default in NFD is used. The NFD of the router that received the Interest performs a forwarding process in a similar way. When the NFD of the camera node receives the Interest, it carries out car

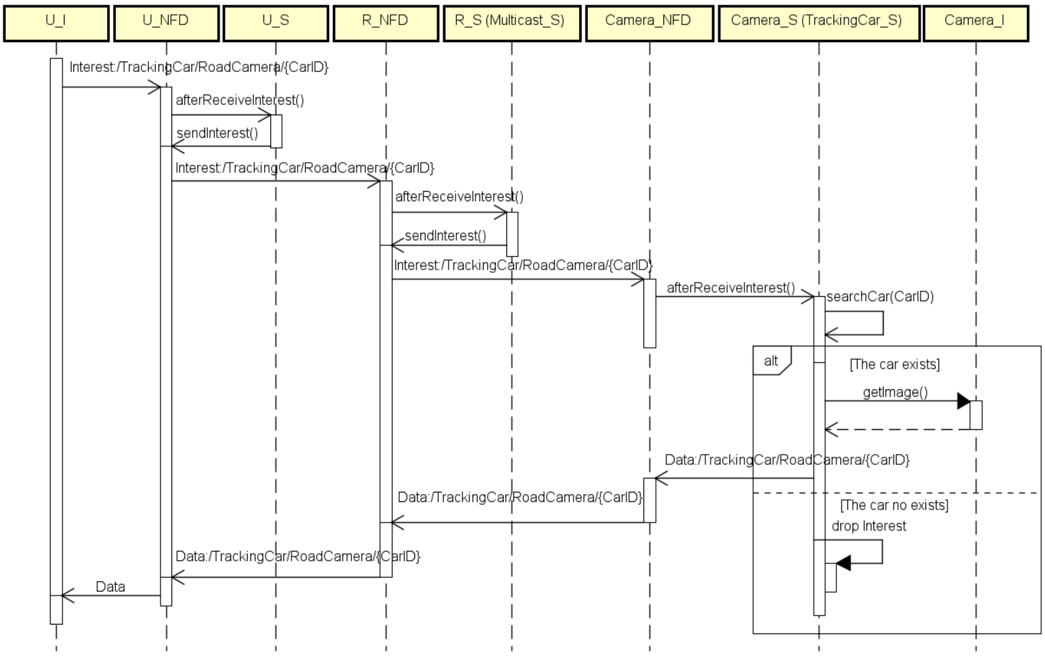


Figure 6: A sequence diagram of Delivering Phase

registration processing from the strategy of the camera node. By this registration processing, the car is registered in the Car List held by the camera node.

4.4.2 Requesting and Delivering Content (Delivering Phase)

The consumer sends the Interest and network forwarding it to the appropriate camera node and delivers the video from the camera to the consumer. Here we call it Delivering Phase. Figure 6 shows a sequence diagram of Delivering Phase. In this Figure, U_I is the interface of a user node, U_NFD is the NFD of a user node, U_S is the strategy of a user node and Camera_I is the interface of a camera node.

Communication is started by creating an Interest (`/TrackingCar/RoadCamera/{CarID}/{sequencenumber}`) including the car ID to be tracked from the interface of the user node. This process is executed by the Consumer program described in Section 4.2. The NFD of the user node receives the created Interest and determines the forwarding destination via the strategy. In this case, the strategy set by default in NFD is used. The NFD of the router that received the

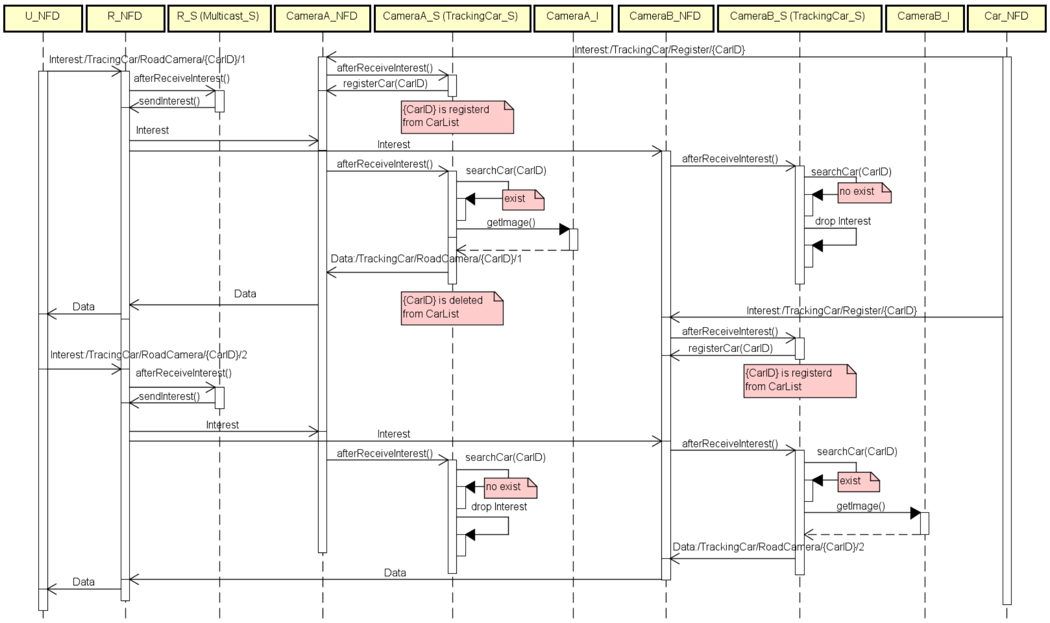


Figure 7: A sequence diagram of Handover Phase

Interest transfer it to the strategy. This strategy is Multicast Strategy, which decides the multicast forwarding destination, and the NFD multicasts the Interest. When the NFD of the camera node receives the Interest, it checks from the strategy of the camera node (TrackingCar Strategy) whether the car is within the shooting range of the camera. If there is a car, transfer it to the interface of the camera node, acquire the camera image and return it as the Data. This camera image acquisition process is executed by the Producer program described in the 4.2. If the car does not exist, drop Interest. The Data is returned to the consumer according to the PIT.

4.4.3 Switching Camera (Handover Phase)

In this system, cameras that return the Data are switched according to the movement of the car. We call this phase Handover Phase. Figure 7 shows a sequence diagram of Handover Phase. To simplify the sequence diagram, the interfaces and NFDs of a user and a car are omitted. Here, let us consider a case where the car has moved within the shooting range of the camera B from within the shooting range of the camera A.

The car first sends an Interest (`/TrackingCar/Register/{CarID}`) containing its own car ID to the node of the camera A. As a result, in the NFD of the camera A, the car is registered in the CarList. The user sends an Interest (`/TrackingCar/RoadCamera/{CarID}/1`) containing the car ID that the user wants to track., and it is forwarded to the router. And this Interest is multicast to the camera A and the camera B by the Multicast Strategy. Since the car is registered in the CarList of the camera A, the image of the camera A is returned to the consumer. On the other hand, since the car is not registered in the camera B, the Interest arriving at the camera B is dropped. After the car has moved into the shooting range of the camera B, the car subsequently sends the Interest (`/TrackingCar/Register/{CarID}`). As a result, the car is registered in the CarList in the camera B. In NFD of the camera A, the car is deleted from the CarList. Subsequently, when the consumer specifies the next frame number and sends the Interest (`/TrackingCar/RoadCamera/{CarID}/2`), the Interest is multicast to camera A and camera Bin the same way. Since the car is not registered in CarList of camera A, the Interest arriving the camera A is dropped. On the other hand, since the car is registered in CarList of camera B, the image of camera B is returned to the consumer.

As described above, even if a handover of the car occurs, the camera that distributes image can be switched by Interest multicast and consumer can continuously acquire the video.

4.5 Utilizing In-Network Processing

In this section, we consider utilizing in-network processing in ICN in the form of extending the designed tracking system. When acquiring a camera image by tracking a car, a car may exist near the boundary between the two cameras. In this case, when acquiring images of either camera, only a part of the car appears, or the car may not be reflected due to delay. Therefore, in this thesis, the data processing is executed in network. The content of data processing is to merge the images of the two cameras that may have a car. In ICN, the consumer can designate merge, etc. by using the name. Also, it is possible for the network to recognize the position information and signal strength of the car and to merge the required area automatically. By implementing this function, we show that ICN can provide flexible data processing in the network, shows the effectiveness of ICN, such as improvement of programmability, in a situation where control is performed on a multiple end nodes. The processing procedure in this thesis is described below. 0: The car sends the information of the car to the network. 1: The consumer sends Interest requesting images. 2: The router determines the camera nodes from which to acquire the image. 3: The router gets the raw data from the cameras. 4: The router processes the received the raw data. 5: The router sends the newly generated data to the consumer.

By incorporating this processing in the tracking system, it is possible to acquire necessary parts from multiple cameras, and it is possible to solve the problem that the car in the video is cut off by the boundary on shooting range of a camera. In order to realize this processing with a location-oriented communication protocol like IP, it becomes complicated to develop an application including solution of the address of the camera and provision of the data processing function. In ICN, incorporating data processing as well as data transfer in the network leads to improvement of programmability, In this research, we show that by implementing this system, the utilization of in-network processing in ICN will expand the possibilities and applications of ICN.

In the following, we describe the strategy prepared to implement this extension.

4.5.1 Strategy

We implement the ProcessCamera Strategy to perform image processing at the router. In this strategy, upon receiving an Interest, camera nodes that acquire an image are determined from the car information in the table. And JPEG frames are acquired from multiple camera nodes. After determining the allocation of multiple JPEG frames acquired from the car state, merging it into one frame, the router makes this frame content by the producer program. The consumer can also acquire frames subjected to image processing by similarly issuing the Interest requesting the next segment.

In this thesis, since the goal is to show that in-network processing can be realized with ICN, we only implement simple processing to change the distribution of images for each frame. Specifically, two cameras (Camera A and Camera B) are connected to the router, and the allocation of each camera is changed as the frame advances such that the allocation of each camera (A:B) is (8:0 → 7:1 → 6:2 → 5:3 → 4:4 → 3:5 → 2:6 → 1:7 → 0:8).

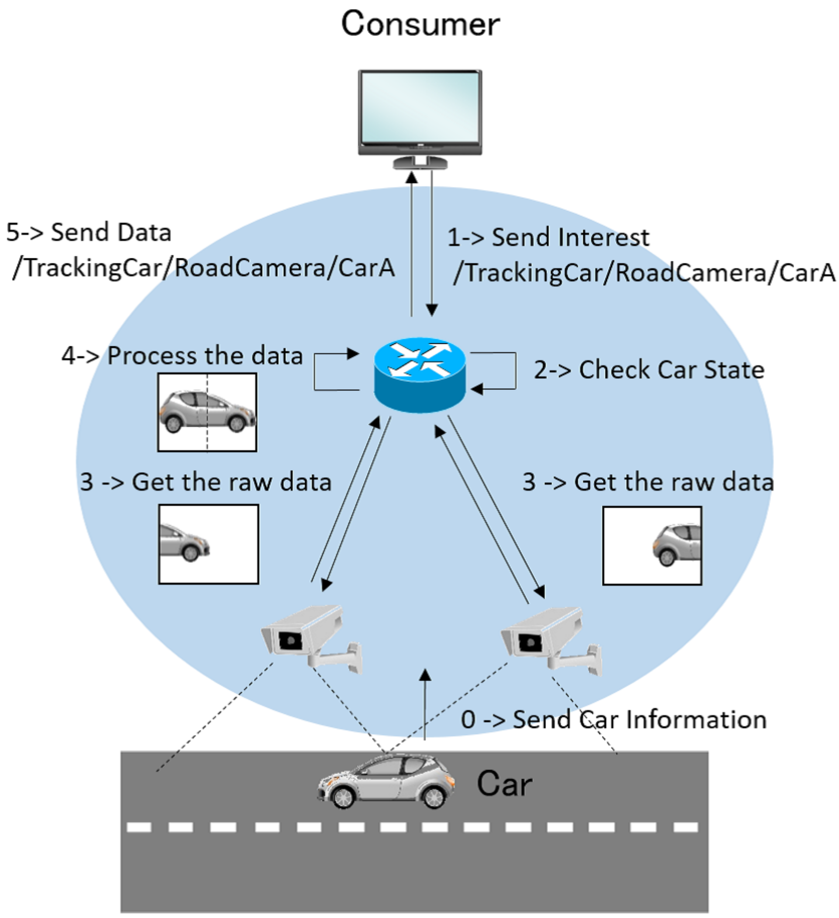


Figure 8: Overview of system utilizing in-network processing

Table 7: Description of tools

Tool	Version	Description
ndn-cxx	0.6.0	C++ NDN library
NFD	0.6.0	NDN packet forwarder
C++	C++ 11	Language of ndn-cxx and NFD
Ubuntu	14.04	User Node's OS
Raspbian	Sep 2017	Raspberry Pi's OS
mjpg_streamer	3:172	Tool to get the image from a camera

Table 8: Description of equipment

Equipment	Model	Description
User Computer	Panasonic Let's note CF-MX5	Install Ubuntu on VM
Small Computer	Raspberry Pi 3 Model B	Use as NDN node
Camera	RPi Camera Module V2	Connect directly to RPi
Radio Control Car	TAMIYA 1/24RC Heavy Dump	L 35cm$\times$ W 27cm$\times$ H 21cm

5 Implementation and Experiments

In this section, we implement the designed tracking system using actual equipment. Then, in order to confirm that the designed system operates correctly, we conduct simple basic experiments and confirm that cameras are switched appropriately and images are acquired by tracking a moving object. We describe the tools and equipments used in the implementation in Section 5.1, the abstraction of experiments in Section 5.2, the experimental environment in Section 5.3, and then the experimental methods in Section 5.4. Finally, we describe the experimental results in Section 5.5.

5.1 Implementation

Table 7 lists the software and tools used for the implementation. Table 8 lists the equipments used for the implementation. The user node installs Ubuntu Os in Virtual Box on Windows note PC, and install ndn-cxx and NFD to enable NDN communication. Raspberry Pi (RPi) is used for each small computer, which is installed ndn-cxx and NFD. Raspberry Pi Camera Module v2 (Figure 9)

Figure 9: Camera Module v2

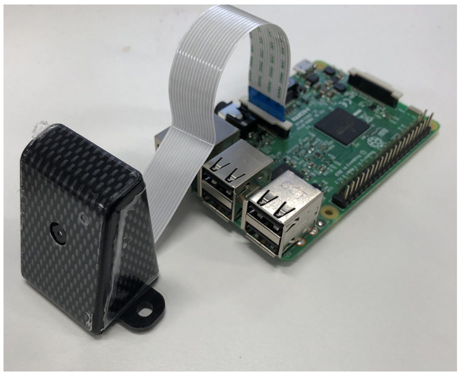


Figure 10: Raspberry Pi 3 Model B

is connected to RPi used as a camera node. Figure 10 shows the appearance of RPi with camera module. The image from the camera module can be acquired by command input for the camera module, but we acquire image by using mjpg-streamer in consideration of the operation speed. As a car to be used for the implementation, we use a radio control car. It is operated by a dedicated remote controller. RPi is mounted on the carrier of the radio control car and it is communicated from the radio control car to the network. The overall view of the radio control car is shown in Figure 11. For development language, we use C++ used in ndn-cxx and NFD.

5.2 Overview of Experiments

In this thesis, by conducting the experiment, we show that the designed system can select the camera correctly and deliver the video to the consumer properly according to the movement of the moving object. Multiple RPi and PC are connected by wi-fi and an overlay network of NDN is constructed. Three camera nodes generate contents from images captured by the camera module. RPi on the car sends a message, including the car ID to the network. When a consumer requests video, the Interest with the car ID specified is sent. The Interest is multicast to multiple camera nodes. Video data are returned only from the camera node whose car is within the shooting range. The consumer continues to send Interest, and the camera is switched correctly according to the movement of the car. By confirming the above operation is performed, we confirm that the proper data can be acquired by tracking the moving object.

In addition, as an extension of the tracking system for moving objects, in order to show that in-

Figure 11: Overall view of radio control car

network processing can be used, we apply the strategy to the router which processes images from multiple cameras and returns processed image as one frame. By confirming the applied strategy works properly, we show ICN can provide flexible services by using in-network processing.

5.3 Setup for Experiments

In experiments, we place three RPi with camera modules at floor A6F of Osaka University Graduate School of Information Science and Technology, like the Figure 12. On the floor, we place the PC used as the consumer and the RPi used as the router. Within the shooting range of the three cameras, we run a radio control car equipped with RPi by remote control. All RPi and the PC wirelessly connect with 2.4GHz band Wi-Fi. The network configuration is shown in Figure 13.

First, we describe the operation required for network setting of NDN. In all the nodes, after connecting to Wi-Fi and setting IP address, start the NFD with the "nfd-start" script. And, at the consumer terminal, we execute the "nfdc cage create udp://192.168.2.115" command to crate a UDP tunnel to NFD of the router. Next we execute "nfdc route add /TrackingCar udp://192.168.2.115" command to add a route toward the router. At the router terminal, we execute "nfdc face create" and "nfdc route add" commands for the three camera nodes (192.168.2.109, 192.168.2.113, and 192.168.2.114). At the car node terminal, we also execute "nfdc face create" and "nfdc route add" commands for the three camera nodes. This completes the network setting.

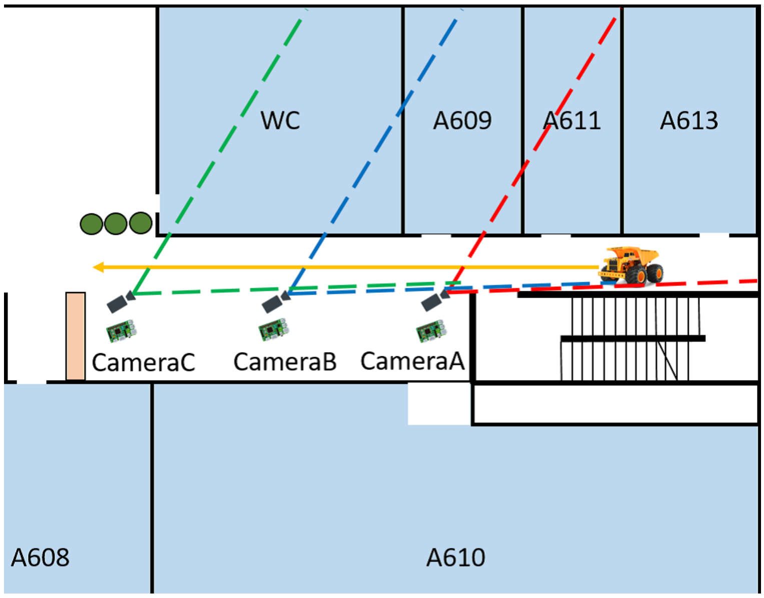


Figure 12: Experimental environment

In each camera node, in order to generate a JPEG frame of video from the camera module, we use the software mjpg-streamer. In mjpg-streamer, in addition the function to perform video streaming using proprietary transfer method via HTTP, it can generate image frames in JPEG format. We shot video at resolution 640×360, JPEG quality 50, frame rate 8 [fps]. The producer program receives an Interest, it acquires a JPEG frame from the video taken by mjpg-streamer and generates a content.

RPi on radio control car operates the Register program, and sends Interest to the camera. In this experimental environment, it is difficult to select and transmit nearby access point because it is a scale that allows direct wireless connection with all roadside cameras. Therefore, in this experiment, SSH is connected to RPi on the radio control car from another terminal, and the location of the radio control car is visually observed, and the forwarding destination of an Interest is manually specified.

5.4 Experimental Methods

In this thesis, in order to show that the Tracking System for moving objects designed and implemented by ICN operates correctly, two operation experiments are carried out and the following

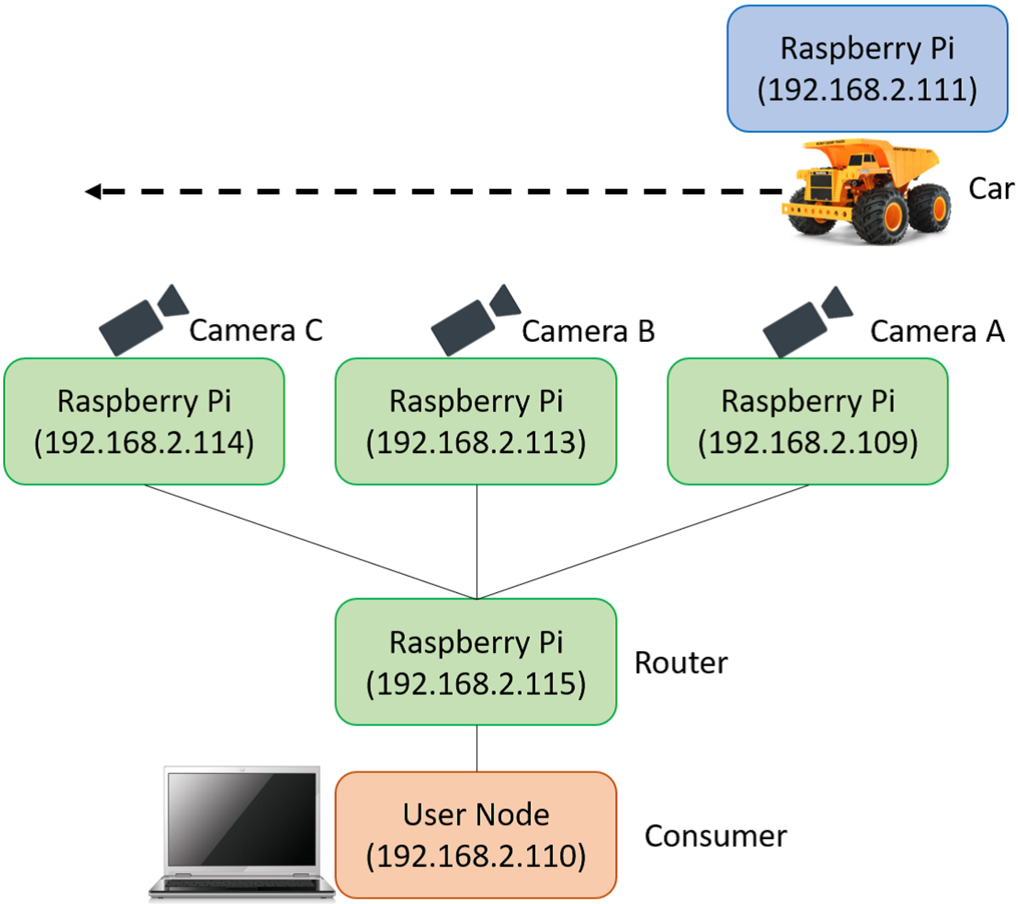


Figure 13: Configuration of experimental network

two points are shown.

- The consumer continuously sends the Interest, and we confirm that the camera node is switched correctly according to the movement of the car.
- We apply a strategy to a router that performs image processing to merge frames from multiple cameras, and confirm that in-network processing in ICN can be realized.

The procedure of each experiment is described below.

Experiment of Tracking-Car

First, we prepare for data distribution in all the camera nodes. In the NFD of each camera node, we execute “nfdc strategy set /TrackingCar /localhost/nfd/strategy/TrackingCar/%FD%01” command to apply the TrackingCar Strategy. After running the mjpg-streamer, we execute the producer program. When executing, we specify the name “/TrackingCar/RoadCamera” as an argument so that camera nodes can receive the Interest with this prefix.

In the NFDs of router and RPi on the car, we execute “nfdc strategy set /TrackingCar /localhost/nfd/strategy/multicast/%FD%03” command to apply the Multicast Strategy. As a result, the Interest is forwarded to each camera node.

Connecting to car node with SSH, we execute the script to start the Register program. In this script, if you specify the car ID (CarA) and execute it, it will wait for input. When entering the current location (in this experiment, the area name is 1 or 2 or 3) while waiting for input, the Interest is transferred to the camera nodes and the Car List of each camera node is updated by TrackingCar Strategy.

Next, the Consumer program is executed from the user node terminal. The Consumer program requests content named ”/TrackingCar/RoadCamera/CarA/<seaquenceno>” and sends the Interest. This Interest is multicast to three camera nodes via a router. Only the camera whose CarID specified in the name of Interest exists in the CarList responds to the Interest and transferred to the Producer program. In the Producer program which received the Interest, a JPEG frame is generated and converted into the contents. In NDN, a data exceeding MAX_NDN_PACKET_SIZE cannot be transferred with one packet, therefore JPEG frame is divided into segments. The camera node sends a first segment as the Data added version information and segment number. Since the segment size is included in the metadata of this Data, the consumer that received the Data will then increment the segment number by 1 and request next segment. Each time acquisition of one frame is completed, the frame number is incremented one by one to request the content of the next frame, and the video is continuously acquired. While the video is continuously received, when the car information of the network is updated by moving the radio control car, we confirm that the camera to be acquired is switched. Through these procedures, we confirm that the designed Tracking system is operating properly.

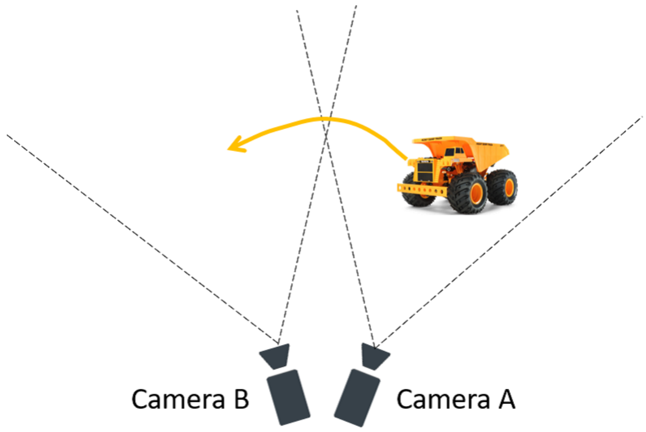


Figure 14: Location relationship between cameras and vehicle

Experiment of system utilizing In-Network Processing

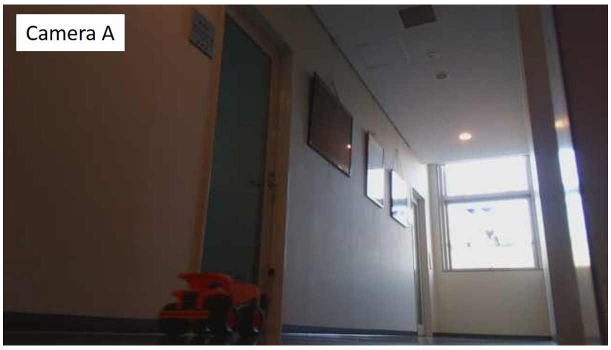
We describe the method of experiment of the system which utilizes in-network processing. In the NFD of the router, we execute the “nfdc strategy set /TrackingCar/localhost/nfd/strategy/ Process-Camera/%FD%01” command to apply the ProcessCamera Strategy. In this strategy, the router holds the car status in the table, acquires the video by selecting from multiple cameras according to the car status. And, the router determines the allocation of the camera in one frame, and creates the content. The arrangement of cameras in this experiment is shown in Figure 14. In this experiment, we set up two cameras and arrange them so that their shooting areas are adjacent to each other. It is assumed that the moving speed of the car can be predicted, and each time the next frame is requested, the area of each camera changes at a fixed rate (1/8 of the image area).

We confirm that if the consumer continuously request contents, the ratio of the camera in the acquired image frame changes as the frame advances.

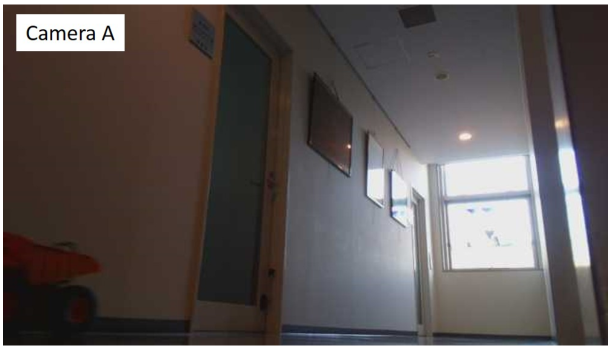
5.5 Experimental Result

Experiment of Tracking-Car

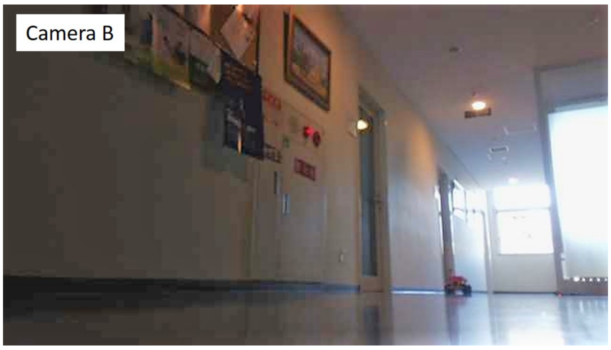
We moved the radio control car with the remote controller, requested video from the user terminal,



(a) 15th frame



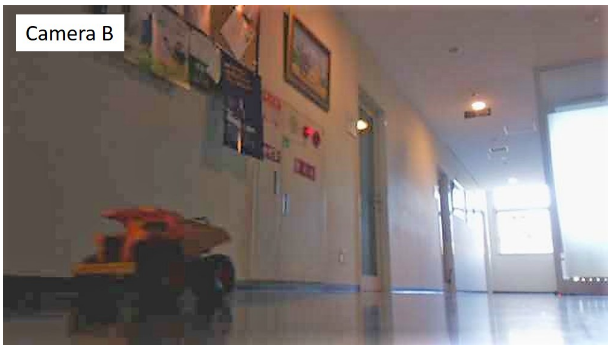
(b) 16th frame



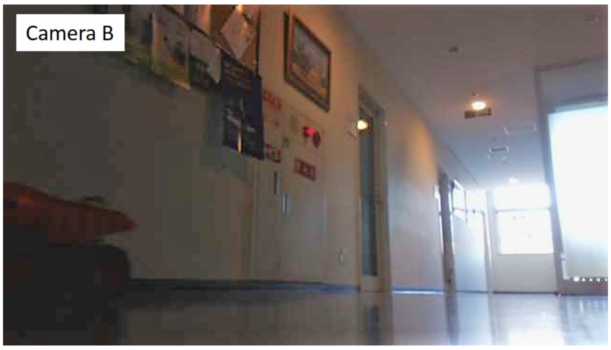
(c) 17th frame

Figure 15: Retrieved JPEG frames (Camera switching occurred from 16th frame to 17th frame)

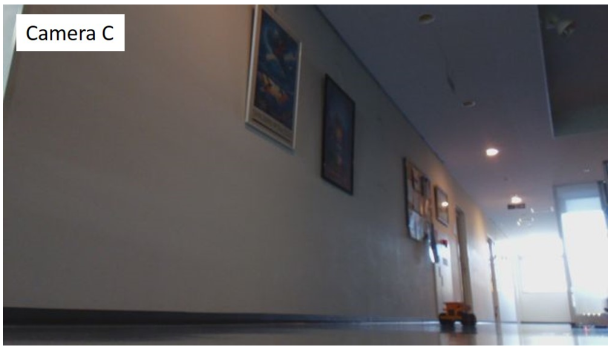
and acquired the JPEG frames. Parts of the acquired frames are shown in Figure 15 and Figure 16. According to the movement of the car, it was confirmed that the camera which distributes the



(a) 27th frame



(b) 28th frame



(c) 29th frame

Figure 16: Retrieved JPEG frames (Camera switching occurred from 28th frame to 29th frame)

images are switched at the 17th frame (Camera A → Camera B) and the 29th frame (Camera B → Camera C). From this fact, we confirmed that the network tracked the car so that it was able to

(a) 1st frame (b) 2nd frame (c) 3rd frame

(d) 4th frame (e) 5th frame (f) 6th frame

(g) 7th frame (h) 8th frame (i) 9th frame

Figure 17: JPEG frames retrieved by applying in-network processing

dynamically change the camera according to the car position and return the Data. The size of the acquired JPEG frame was about 10KB, and the frame rate was about 2[fps].

Experiment of In-Network Processing

Figure 17 shows the image frame acquired by applying the created strategy using in-network processing. The frame rate was about 1[fps]. As a result of the strategy processing. It was confirmed that the ratio of the images of the Camera A and Camera B was changed as the frame advanced. In the ICN, it was confirmed that flexible in-network processing such as changing content dynamically according to the state of the moving object is possible. Furthermore, by sending the speed of the car and so on to the network, it is possible to realize advanced services such as completely tracking the movement of the car and acquiring only the necessary area from multiple cameras.

6 Discussion

In this chapter, the consideration is described based on the results of the experiment.

In this experiment, we showed that ICN realizes dynamic service such that the network recognizes the position of one radio control car passing the three cameras existing in the same subnetwork, and appropriately selects and switches the camera shooting the radio control car.

Furthermore, we confirmed that ICN realizes the in-network processing which the relay router acquires and merges the images from the two cameras and return the processed frame. As an application of this service, there are conceivable methods for extracting the position of the car from the image coordinating with the adjacent camera node. [10] shows that the position of the car can be tracked by sharing the feature data between the cameras. By realizing in-network processing in ICN, there is no need to allocate calculation resources to all edges, and there is the merit that the number of nodes providing processing functions can be reduced. Moreover, we showed that it is easy to implement more flexible service provisions such as dynamically changing contents.

From the above, it was shown that ICN functions as a service enabler that can provide services themselves, and it is useful as an information distribution infrastructure that can realize not only data transport but also advanced services including control, management, data processing.

However, this experiment is verification under a small-scale network configuration. In order to evaluate the designed system in detail, it is necessary to consider deployment in a real scale. In the following, we consider issues for deployment in a real scale.

Radio communication of moving objects

In order for the network to track a car that is a moving object, it is necessary for a car to communicate with the network to register information on the car. In case of realizing with genuine ICN instead of overlay network, the car can communicate by appropriately selecting a nearby access point. As a result, the network car recognizes the position of the car. However, in the NDN, which is the overlay network, it takes time for processing such as registration and switching of the transfer destination face in the node, so that the position recognized by the network deviates from the actual position of the car. In this thesis, since it was an experiment in the overlay network, the car manually specified the camera node as the transfer destination and communicates.

When realizing the designed system in the NDN overlay network on a real scale, there is a high possibility that delay and packet loss will occur due to the processing at the time of switching the access point under the high speed movement of the car. In order to provide the network that can track moving objects in a real environment, it is necessary to have an infrastructure access point that can instantaneously complete handover processing in units of 100 milliseconds. For handover, the processing must be finished in an area where the radio waves of both the currently connected access point and the next access point can be transmitted and received. For example, in a car that is moving at a speed of 60 km/h, if the latency associated with handover (the time from the car transmission to the update of the CarList of the camera node) is 100 milliseconds, the car moves about 1.6 m. Therefore, it is necessary to arrange so that the coverage of adjacent access points overlaps by 2m and switches from the old access point to the new access point appropriately.

In order to apply it to the actual scale, it is necessary to solve these problems and conduct experiments.

Impact on network bandwidth

A lot of research on mobility support in ICN have been done, but in this research, we adopt multicast based mobility support. In this method, it is unnecessary to update the mapping server, so it is clarified from the experiment that the camera can be quickly switched by updating the car information in an edge. In the scenario in which it is desired to change the camera node or to communicate with the car, in the Mapping based method, the mapping server is updated frequently and cannot respond quickly, but the multicast based method is effective for seamless mobility support.

However, when multicasting is performed on the network scale of the real environment, there are problems of overhead of PIT update and compression of bandwidth. Also, it is necessary to decide how to set the range for multicasting. [26] evaluates the advantages and disadvantages of seamless mobility supports in real-time applications, but it reasoned that in order to devise the appropriate mobility support technology, further discussion and large scale deployment are necessary. In future system experiments on a real scale, it is necessary to evaluate the influence on network bandwidth considering the requirements for high-speed mobile communication.

Naming scheme

In this experiment, it was confirmed that one car can be tracked and images can be acquired. Even when different cars are requested, it is possible to acquire by issuing contents of different names.

However, when multiple cars (ex: CarA, CarB) exists in the same camera, the are actually requested with the Interest with a different name despite being the same image actually. The producer side issues a plurality of contents with different names given to the same video and there is a problem that is not efficient. Also, since the names are different even for the same video, the cache which is the advantages of ICN can not be used.

For more effective communication utilizing ICN, it is desirable that the same contents can be accessed with different names. That is, the naming scheme that allow to access the same content with either "`/TrackingCar/RoadCamera/CarA`" and "`/TrackingCar/RoadCamera/CarB`" is desired. As a result, only one content is generated, and caching can be used to realize efficient communication.

Naming schemes and name resolution systems are also research topics of ICN, and numerous studies have been done. In this research, naming scheme considering efficient communication is not considered, so it is necessary to devise naming which handles implementation model efficiently for practical use.

7 Conclusions and Future Work

In this thesis, in order to develop network functions that realizes dynamic service in ICN, we designed the tracking system for moving objects. Using the car as the moving object, we created an application that switches the road camera and distributes images according to the movement of the car. In addition, focusing on in-network processing in ICN, we created the strategy to process images from multiple cameras by relay nodes and distribute them. Based the designed system, we implemented it using the real machine and radio control car. From the results of experiments, we confirmed that ICN can realize a dynamic service that dynamically switches the camera according to car movement and delivers the video by providing the tracking function and deciding the camera. Furthermore, by utilizing in-network processing, content can be changed more flexibly, and ICN showed that it is the network architecture that can realize advanced services. Also, in order to deploy the system in a real scale, we discussed issues to be solved.

In future work, we plan to conduct experiments using high-performance equipment and actual vehicles, and evaluate communication capability in environments where scale has expanded.

Acknowledgments

This thesis would not accomplish without the united efforts and supports of a number of people. First, I would like to express my the deepest appreciation for my supervisor, Professor Masayuki Murata of Osaka University, for his valuable time, advice, and continual support. Furthermore, I am tremendously grateful to Professor Shingo Ata of Osaka City University for his significant guidance, precise advice, and helpful discussions. I am also thankful to Associate Professor EUM Suyong of Osaka University for his instructive advice. Moreover, I would like to express gratitude to Associate Professor Shin'ichi Arakawa, Assistant Professor Yuchi Ohsita, Daichi Kominami, and Appointed Assistant Professor Naomi Kuze and Tatsuya Otoshi for their insightful comments and feedback.

In addition, I sincerely Mr. Atsushi Ooka and Mr. Taku Kitagawa for giving me many thoughtful advices. Moreover Mr. Gao Yuhao also helped me, and were valuable sources of research knowledge for me. Finally, I owe my gratitude to all the members of the Advanced Network Architecture Research Laboratory at the Graduate School of Information Science and Technology, Osaka University.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pp. 1–12, Dec. 2009.

[2] P. Gusev and J. Burke, “NDN-RTC: Real-time videoconferencing over named data networking,” in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pp. 117–126, Sep. 2015.

[3] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, “Information Centric Networking in IoT scenarios: The case of a smart home,” in *2015 IEEE International Conference on Communications (ICC)*, pp. 648–653, Jun. 2015.

[4] C. Tschudin and M. Sifalakis, “Named functions and cached computations,” in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 851–857, Jan. 2014.

[5] NFD Team, “NFD Management protocol.” `https://redmine.named-data.net/projects/nfd/wiki/Management`. Accessed: 2017-11-01.

[6] NFD Team, “NFD Developer’s Guide.” `https://named-data.net/wp-content/uploads/2016/10/ndn-0021-7-nfd-developer-guide.pdf`. Accessed: 2017-5-18.

[7] A. Rowstron and G. Pau, “Characteristics of a vehicular network,” tech. rep., University of California Los Angeles, Computer Science Department, Jul. 2009.

[8] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, “VANET via Named Data Networking,” in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) 2014*, pp. 410–415, Apr 2014.

[9] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, “Navigo: Interest forwarding by geolocations in vehicular Named Data Networking,” in *Proceedings of IEEE World of Wireless Mobile and Multimedia Networks (WoWMoM) 2015*, pp. 1–10, Jun 2015.

[10] A. Noda, M. Hirano, Y. Yamakawa, and M. Ishikawa, "A Networked High-Speed Vision System for Vehicle Tracking," in *Proceedings of IEEE Sensors Applications Symposium*, pp. 343–348, Feb. 2014.

[11] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, *et al.*, "Named Data Networking (NDN) project," Tech. Rep. NDN-0001, PARC, Oct. 2010.

[12] NDN Project, "ndn-cxx: NDN C++ library with eXperimental eXtensions." `http://named-data.net/doc/ndn-cxx/current/`. Accessed: 2017-11-27.

[13] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A Survey of Mobility Support in Named Data Networking," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) 2016*, pp. 83–88, Apr. 2016.

[14] A. Azgin, R. Ravindran, and G. Wang, "A Scalable Mobility-Centric Architecture for Named Data Networking," in *Proceedings of IEEE 23rd International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2014.

[15] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure namespace mapping to scale NDN forwarding," in *Proceedings of 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 281–286, Apr. 2015.

[16] F. Hermans, E. Ngai, and P. Gunningberg, "Global Source Mobility in the Content-Centric Networking Architecture," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pp. 13–18, Jun. 2012.

[17] D. Li and M. C. Cuah, "SCOM: A Scalable Content Centric Network Architecture with Mobility Support," in *Proceedings of IEEE Ninth International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 25–32, Dec. 2013.

[18] J. Lee, S. Cho, and D. Kim, "Device Mobility Management in Content-Centric Networking," *IEEE Communications Magazine*, vol. 50, Dec. 2012.

[19] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "Mobility Support in Content Centric Networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 13–18, Aug. 2012.

[20] L. Wang, O. Waltari, and J. Kangasharju, "MobiCCN: Mobility support with greedy routing in Content-Centric Networks," in *Proceedings of 2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 2069–2075, Dec. 2013.

[21] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi, "PMC: Publisher mobility support for Mobile Broadcasting in Content-Centric Networks," *ASIA Future Internet 2012 Summer School*, Aug. 2012.

[22] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, "Anchor-less Producer Mobility in ICN," in *Proceedings of the 2nd International Conference on Information-Centric Networking*, pp. 189–190, Sep 2015.

[23] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for NDN," in *Proceedings of the 1st International Conference on Information-Centric Networking*, pp. 179–180, Sep. 2014.

[24] V. Jacobson, R. L. Braynard, T. Diebert, P. Mahadevan, M. Mosko, N. H. Briggs, S. Barber, M. F. Plass, I. Solis, E. Uzun, *et al.*, "Custodian-Based Information Sharing," *IEEE Communications Magazine*, vol. 50, Jul. 2012.

[25] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting Seamless Mobility in Named Data Networking," in *Proceedings of IEEE International Conference on Communications (ICC) 2012*, pp. 5854–5869, Jun. 2012.

[26] M. Alajlan and A. Belghith, "Supporting Seamless Mobility for Real-Time Applications in Named Data Networking," *Procedia Computer Science*, vol. 110, pp. 62–69, Jul. 2017.

[27] NDN Project, "NDN Essential Tools." `https://github.com/named-data/ndn-tools`. Accessed: 2017-11-27.