Regular Paper

Adaptive NDN Content Delivery Mechanism on Mobile Networks

Taichi Iwamoto† and Tetsuya Shigeyasu‡

†Graduate School of Comprehensive Scientific Research, Prefectural University of Hiroshima, Japan
‡Department of Management Information Systems, Prefectural University of Hiroshima, Japan
sigeyasu@pu-hiroshima.ac.jp

Abstract - Recently, named data networking (NDN) has attracted considerable attention from network researchers as a network architecture based on a content-centric manner. NDN delivers user requests according to the shortest path recorded in the forward information base (FIB) of relay content routers (CRs). Incidentally, the development of portable ICT devices enables the publishing of content even in the mobile environment. The current version of NDN, however, does not consider publisher migration. If the location of the content publisher changes, NDN cannot deliver the *Interest*/content to the appropriate destination accurately. In this study, we discuss how to address the publisher migration problem on NDN and propose a new adaptive content delivery method and priority cache holding method. The proposed methods are evaluated under the scenario that the mobile publisher changes its location. The results of the evaluations confirm that our proposal improves the performance of content delivery even in a mobile environment.

Keywords: NDN, publisher migration, cache management, adaptive content delivery

1 INTRODUCTION

Recently, many new network architectures based on information centric data delivery have been proposed [1]-[4]. Named data networking (NDN) is a popular architecture that has garnered considerable attention from network researchers as a network architecture based on a content-centric manner. NDN delivers users' requests and their corresponding contents according to the shortest path recorded in the forward information base (FIB) of relay content routers (CRs). Incidentally, the development of portable ICT devices enables the publishing of content even in the mobile environment.

The current version of NDN, however, does not consider publisher migration. For example, NDN does not update relay information in the FIB in accordance with the dynamic topology change. When the topology change occurs by node migration, relay information in the FIB can no longer be used. Wrong FIB information misleads *Interest* the content requested by users and increases network traffic needlessly. Therefore, a new mechanism for updating FIB information quickly in response to publisher migration is required.

Based on the situation of a publisher's migration, contents generated on the migrating publisher cannot be reached from the network. Network reachability for those contents will recover once the migrating publisher completes its location change and reconnects to the network. Thus, during the period from the beginning and end of migration, the users will not be able to obtain the contents of the migrating publisher if the CRs do not hold the cache. Hence, a new cache management method for maintaining the CR cache of a mobile publisher during its migration is required.

To address this publisher migration problem on NDN, this study proposes a new adaptive content delivery mechanism for mobile networks, comprising two parts: pre-forwarding with update method of old/wrong FIB information, and preferentially cache-keep method for migrating publishers' contents.

In the first method, the old/incorrect FIB information of relay CRs is canceled by control information on pre-forwarded contents transmitted from the migrating publisher before its migration. Subsequently, *Interest* destined for contents of migrating publishers will not be forwarded to the previous location of the publisher. In addition, after the migrating publisher arrives at a new location, another control information is transmitted to form new FIB information for relay CRs. Thereafter, *Interest* destined for the contents will be forwarded to an appropriate new location.

In the second method, during the publisher migration, the cache of the contents generated by the publisher will be preferentially stored on CRs. Therefore, content can be obtained more easily, even if there is no network access to the publisher.

The proposed methods will be evaluated considering scenarios where the mobile publisher changes its location. The results of the evaluations confirm that our proposal improves the performance of content acquisition ratios.

2 PROCEDURE OF NDN

NDN is an architecture that performs both content discovery and content delivery in a content-centric manner. NDN performs the procedure using two packet types: *Interest* and *Data*, as shown as Fig.1.

The *Interest* is used for requesting the desired contents. If a content request has newly arrived at a user, the user transmits this *Interest* toward a publisher/CR with original/cache content of the corresponding content. Meanwhile, *Data* is used for returning the content corresponding to *Interest*. The publisher/CR returns the content as *Data* if and only if it has the contents/cache corresponding to the received *Interest*. The *Data* will be forwarded along the reverse path of the previously received *Interest*.

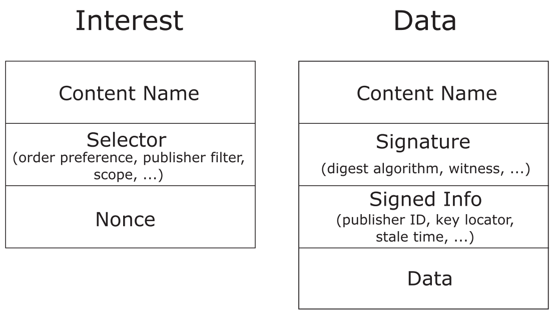

Figure 1: Frame format of *Interest* and Content on NDN.

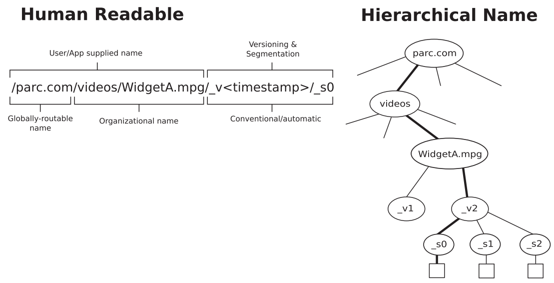

Figure 2: Naming structure of NDN.

In NDN, the name of the content is used as an identifier. All contents will be requested by specifying the names of each content. As shown in Fig. 2, the contents are named using a hierarchical structure divided by "/".

2.1 Construction of CR on NDN

As shown in Fig. 3, CR performs packet forwarding according to the following three tables.

- FIB
  FIB consists of information used for forwarding *Interest* toward a content publisher. When a CR receives a new *Interest*, the CR forwards it according to the information registered in its FIB. In the FIB, each record consists of two fields: prefix of the contents (i.e., the name of contents) and an interface number that must be forwarded the *Interest*.

- PIT (Pending *Interest* Table)
  PIT maintains a history of received content requests corresponding to the forwarded *Interest*. Each PIT record consists of two fields: the prefix of the forwarded *Interest* and several interfaces to which the *Interest* was forwarded. By referring to a PIT entry, the CR can return *Data* to the appropriate direction if the CR receives it in the future.

- CS (Content Store)
  CS is a buffer to store caches of the content temporarily. An entry of CS consists of the prefix and cache of the content. CR returns the cache of the content instead of the publisher generating the original content if and only if the CR has the corresponding cache of the requested contents. When the content is returned to a user from the CR in shorter hops than those required by the original publisher, the amount of network traffic and response time can be reduced effectively.

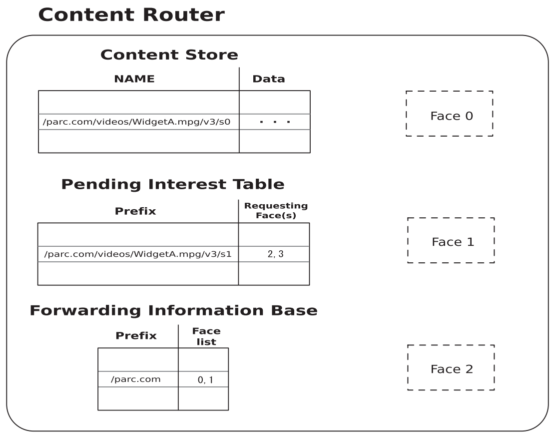

Figure 3: Contents Router in NDN.

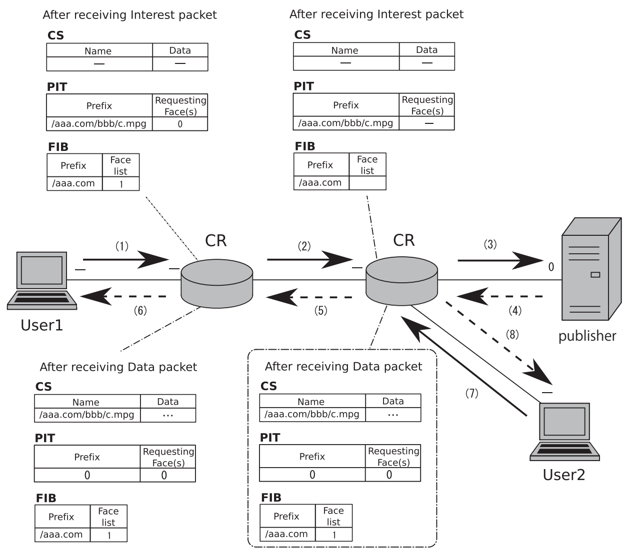

Figure 4: Basic procedure of NDN

2.2 Basic Procedure of NDN

Figure 4 illustrates the basic procedure of NDN. As shown in the figure, (1) the user transmits the *Interest* to the neighboring CR. The CR receiving the *Interest* checks whether the corresponding cache to the *Interest* is stored in its buffer. If a cache exists, the CR returns it as *Data* through the interface that the *Interest* came from; otherwise, the CR registers the information of the *Interest* into its PIT; (2) the *Interest* is forwarded to the next upstream CR. The next upstream CR is selected according to the entry of FIB. If there is no cache corresponding to the *Interest*, (3) the *Interest* is forwarded to the original publisher, who then returns the contents to the user.

During *Data* returning, the CR receiving the returning *Data* stores the *Data* to its CS as a cache, and the CR checks if it

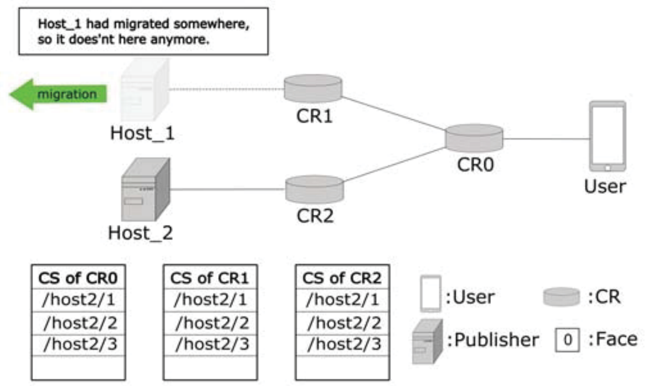

Figure 5: Missing access problem to contents during publisher migration

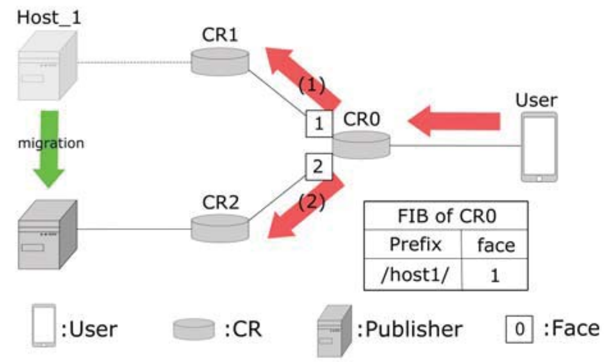

Figure 6: Incorrect *Interest* forwarding problem induced by old FIB information

has an entry corresponding to *Data* on its PIT. If the entry exists, the CR returns the *Data* through the interface(s) in accordance with the entry and removes the entry from the PIT. By repeating the process, CR can deliver the *Data* to the user requesting the contents, even if the *Interest* excludes the location information of the user.

In addition, when other user(s) request the same content, (7)(8), the CR responds by returning the cache of content stored in CS, with a shorter delay compared with traditional host-centric network. Moreover, if the CRs estimate the popularities and keep caches of the higher popular content, the cache hit ratio at the the CR can be effectively improved, and the amount of network traffic can be effectively reduced in NDN.

3 PROBLEMS ON CONVENTIONAL NDN

This section describes the three problems induced by publisher migration on conventional NDN.

3.1 Missing Access Problem to Contents during Publisher Migration

Once the mobile publisher starts its migration to a new location, users requesting the contents may lose access to the original content generated by the publisher. Figure 5 illustrates this problem. As shown, after the beginning of the migration of Host_1, *Interest*s destined for Host_1 cannot reach the desired contents if the requested contents are not stored in CSs on intermediate CRs (CR0, CR1) between the user and Host_1.

3.2 Incorrect Interest Forwarding Problem Induced by Old FIB Information

On conventional NDN, old FIB information will be kept holding on CRs, even if a publisher changes its location by its migration. Thus, the *Interest*s destined to a content generated by the publisher will be delivered to the old publisher location according to the old FIB information. As shown in Fig. 6, the *Interest* destined to the Host_1 content will be forwarded to direction (1), despite the right path to the current Host_1's new location being (2).

4 RELATED WORKS

This section describes the developments of network architectures related to the NDN. In order to reduce both of the network traffics and delays for content delivery, NDN is expected to be replaced instead of traditional location centric network, namely, IP network architecture. However, because current Internet infrastructure is established based on IP, most applications used worldwide are also guaranteed to work only over the IP architecture. Hence, for a smooth architecture migration from TCP/IP to NDN, some methods have been discussed.

The literature [5] discussed methods for coexisting IP and NDN architectures. In this literature, authors classify the existing methods into three types: stack modification, encapsulation and translation. In the literature [6], a method, named, IP/NDN, for coexisting IP and NDN by introducing translation mechanism has been. proposed. In the IP/NDN, translation is implemented to realize coexisting two architectures in same network. IP datagrams from senders are captured at TUN device, and translated into NDN Interest or Data. Those architecture coexistence methods can push architecture migration, IP to NDN.

Incidentally, NDN belongs to the category of Information Centric Networking (ICN) which is most focusing on getting/delivering contents efficiently instead of knowing node location. Currently, the most practical ICN system is CDN (Content Delivery Network). The literature [7] has proposed that content migration method to realize efficient content delivery under the client node mobilities on vehicular network. In the literature, authors proposed a strategy to place contents on network node as edge router cache, based on deep reinforcement learning approach. However, in this method, only client mobilities are considered. In other words, how to deal with the mobilities of publisher is not discussed.

Previously, a method for a proactive selective neighbor caching strategy has been proposed [8]. In this strategy, the contents going to be requested by mobile users will be cached proactively on selected neighbors. Neighbor candidates for proactive caching are selected based on the cache cost delay and mobility behavior. This strategy is for mobility support on the CCN, focusing on user mobility and not on publisher mobility.

DONA also handles events related to the user (consumer mobility) [9]. In DONA, content request/delivery is handled by introducing a resource handler (RH) mechanism, which acts as a domain name system (DNS) server. According to user requests, RH provides the information for rendezvous for content delivery from publishers. In DONA, the user's mobility event is handled by changing RH information.

Although the above mechanism focuses on mobility support on information-centric network architectures, they are focused on user (consumer) mobility rather than on publisher mobility.

4.1 PMC: Publisher Mobility Support Protocol in CCN

To solve the publisher migration problem on NDN, the publisher mobility support protocol (PMC) was proposed [10]. In the PMC, to cope with publisher migration, a publisher selects a *HomeNode*. The publisher registers its new location to the *HomeNode* when it migrates to another place. Using the *HomeNode*, *Interest*s that arrive at the old publisher's location owing to the old/incorrect FIB entry can be correctly forwarded to the new publisher's location. Figure 7 demonstrates the procedure of *Interest* forwarding on PMC. As shown, after a publisher migration, the publisher sends an MR request, including its new location information, to the *HomeNode*. The *HomeNode* returns an MR response to the mobile publisher. At this time, CRs belonging to the forwarding path of the MR response update their FIB entry according to the received MR response.

After the transmission of the MR response, CRs can forward the future *Interest* destined for the contents generated by the mobile publisher.

In the PMC, publisher mobility can be solved by introducing location support of the mobile publisher using *HomeNode*, holding information of both the old and new locations of the migrating publisher. However, without support from the *HomeNode*, PMC cannot address the mobility problem of the publisher, autonomously, Hence, in the following sections, we propose a new autonomous solution for the mobility problem by the publisher.

5 PROPOSAL

We propose a new solution for coping with the problem induced by publisher migration on conventional NDN. Our solution consists of two parts: 1) content pre-forwarding and updating old FIB information, and 2) prioritizing holding for caches generated by mobile publishers until it recovers network connections.

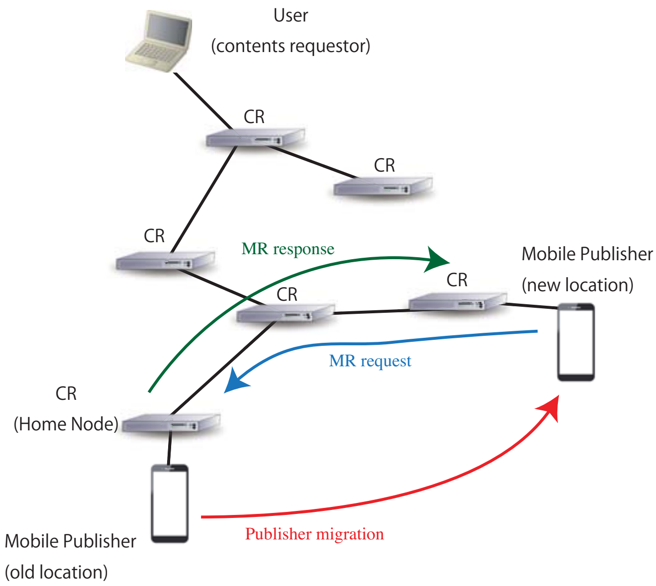

Figure 7: Procedure of PMC

5.1 Content Pre-forwarding and Updating Old FIB Information

This section describes the first part of the proposal. The first proposal, namely content pre-forwarding and updating FIB information, is further divided into two methods: before leaving method and after arrival method.

Hence, the followings describe these two methods in detail.

5.1.1 Before Leaving Method: A Method for Contents Pre-forwarding and Updating FIB Information

As we mentioned previously, during the publisher migration, all users willing to acquire content lose access to them if the desired contents are not stored in the CS of CRs belonging to the *Interest* forwarding path. Therefore, this section proposes a method for content pre-forwarding from the mobile publisher to the neighboring CR before publisher migration and for updating FIB information according to the header information of the pre-forwarding contents. By pre-forwarding, *Interest* can be delivered to the CS holding the corresponding contents even if the mobile publisher changes its location. This phenomenon increases content acquisition ratio. By using the pre-forwarded contents, this method also deletes the old (wrong) FIB information on CRs belonging to the shortest path from users to the previous publisher location. For the deletion of FIB information, caches of pre-forwarded content are marked with a *M-flag* (Migration flag), which indicates that the publisher that generated the content already left the network from the old location. Similar to the conventional approach, the cache of the pre-forwarded content is returned when the CR receives the corresponding *Interest*. However, the CR deletes the FIB information corresponding to the content name when it receives the content with the *M-flag*. Figure 8 demonstrates the procedure of this mechanism.

The detailed procedure of NDN employing both content pre-forwarding and FIB information updating by *M-flag* is described as follows:

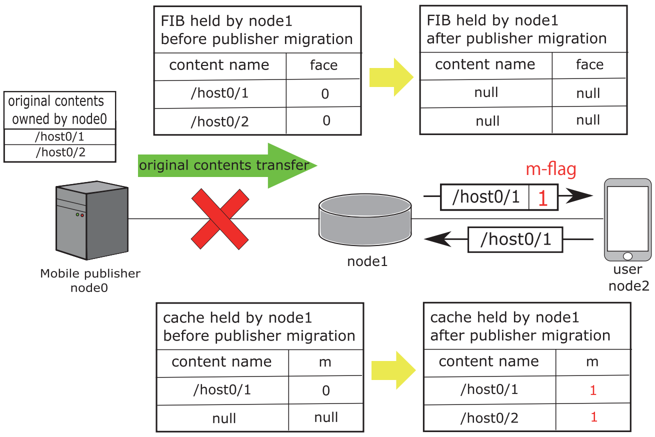

Figure 8: Procedure of content pre-forwarding and updating FIB information by *M-flag*

1. Procedure on CR when receiving contents with *M-flag*
   - (a) Store received content into its CS with *M-flag*.
   - (b) Forward received content with *M-flag* to its downstream CRs registered in its PIT.
   - (c) Delete its FIB information corresponding to the same content name with the received content.
2. Procedure on CR when receiving contents without *M-flag*.
   - (a) Search FIB information having the same content name as the newly received content. Register the new FIB with the information of the received face if the corresponding FIB information is not recorded yet.
   - (b) Search its CS whether content with the same content name is cached or not. If not, store the received content. Otherwise, when the content having the same content name is already cached, CR further checks whether the *M-flag* is set. If so, it unsets the *M-flag*.
   - (c) Forward the received content without M-flag if the corresponding entry is recorded into its PIT.
3. Procedure on CR when it receives the *Interest* (This is similar to the conventional NDN procedure.)

   CR returns corresponding content when it contains the desired cache. Otherwise, it records the requested information regarding the received *Interest* on its PIT and forwards the *Interest* to its upstream CR.

As described in the above pre-forwarding process, a CR that receives an Interest for a content for which M-flag is set returns the content only to the CRs listed in the face registered in its own PIT.

M-flag has the role of deleting FIB information that is no longer needed in order to prevent unnecessary forwarding of Interest to the mobile publisher that has moved to new location.

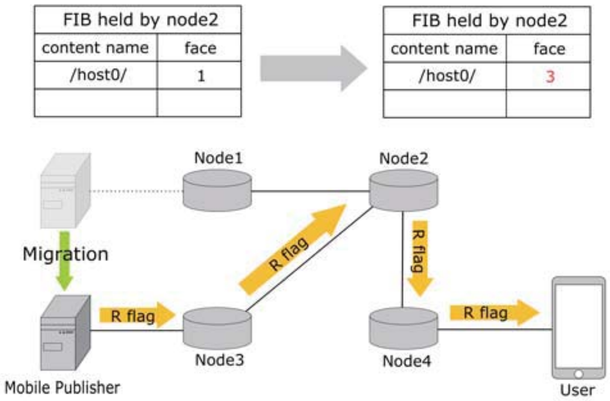

Figure 9: Procedure of dissemination of new FIB information by *R-flag*

However, as described in the following section, the mobile publisher will send another R-flag to set the new location information to all potential users after the move.

Therefore, the pre-forwarding process reduces unnecessary Interest forwarding traffic by removing FIB information from users who have sent Interest to the relevant content and CRs on their Interest forwarding path.

5.1.2 After Arriving Method: A Method for Dissemination of New FIB Information

This section describes the after arriving method that disseminates new FIB information to CRs. In this method, the migrated publisher disseminates the control packet from the new location toward users when its migration is finished. The control packet with *R-flag* (Rebuild flag) on its packet header is transmitted from the migrated publisher. By receiving the control packet with *R-flag*, the CRs that belong to the shortest path among the mobile publisher and users renew their FIB information immediately.

Figure 9 illustrates the procedure for dissemination of new FIB information using *R-flag*. In this method, mobile publishers transmit new control packets with *R-flag* after the completion of their migration. The control packet is then forwarded to the user. At this time, the control packet with *R-flag* is forwarded in all directions that received at least one *Interest*, regardless of its content name.

CRs receiving the content with *R-flag* set new FIB information according to the arrival face of the content with *R-flag*. Accordingly, CRs can obtain the desired content without turning on multicast *Interest* forwarding, which increases needless traffic. The content with the R-flag is forwarded to the downstream CRs until end users.

The detailed procedure for introducing the *R-flag* is described as follows:

1. Procedure for CRs receiving content with *R-flag*
   - (a) Check the FIB information relating to the same content name with the content possessing an *R-*

flag. If it does not exist, the CR adds the FIB information.

(b) Store the received content into it CS.

(c) Forward the content with R-flag to the downstream CRs. To reduce the forwarding traffic, the content is only forwarded to the downstream CRs from all faces that have received any previous Interest.

2. Procedure for users receiving content with *R-flag*

   A user receiving content with *R-flag* checks its FIB information. If there is no corresponding FIB entry, the user adds the FIB entry according to the received content with the *R-flag*.

The proposed method uses R-flag to inform all potential users of the mobile publisher's new location. The CR that receives the R-flag forwards the R-flag to downstream CRs from all faces that have received any previous Interest.

In other words, the R-flag to update FIB information is forwarded to all CRs that have users under them. Therefore, as the number of mobile publishers in the network increases, the network load due to R-flag forwarding also increases.

5.2 Priority Cache Holding for Mobile Publisher Contents

In the previous section, we proposed a method that employs both pre-forwarding and updating old FIB information. By implementing this method, owing to cache pre-forwarding, the cache hit ratio can likely be improved even when the mobile publisher is disconnected from the network.

The pre-forwarded caches, however, will be removed from the CR buffer when the buffer overflows by the new arrival cache. If the pre-forwarded cache is removed from all CRs' buffers, although the mobile publisher has not yet completed its migration, the cache cannot be obtained by any user.

Therefore, to improve cache acquisition performance, it is desirable to keep holding the pre-forwarded cache preferentially, while the mobile publisher is disconnecting from the network.

Figure 10 presents an overview of the proposed priority cache holding method. Before starting its migration, the mobile publisher records the current neighboring CR (in Fig. reffig:priori, the current neighbor is Node1).

The mobile publisher transfers its original content according to the pre-forwarding method, as described in 5.1.1. On priority cache holding, pre-forwarded caches in CRs are recognized by referring to their *M-flag*; such caches are given preference over those without *M-flag*. In case of the necessity to remove any cache from CS on CR due to the arrival of new content, the cache without *M-flag* will be removed first. If and only if there is no cache without *M-flag* in the CS, the pre-forwarded cache, which is the cache with *M-flag*, will be removed from CS (Please note that the cache without *M-flag* can be obtained from the original publisher because the publisher is connected to the network).

After the mobile publisher finishes its migration, the publisher sends a control packet, named *Address* packet, to the

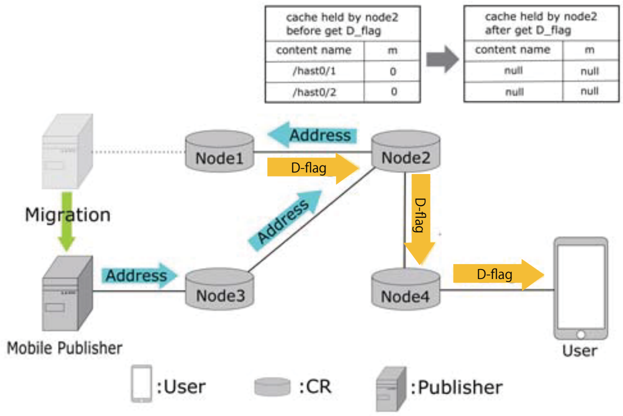

Figure 10: Method for priority cache holding for mobile publisher contents

CR (Node1, in the case of Fig. 10) recorded before the migration. The CRs receiving the *Address* forward it to the designated CR according to the location information without referring to the PIT, FIB, and CS, indicating that *Address* will be transferred by location base.

The destination CR transfers the control packet with *D-flag* (Delete flag) to the downstream from all IFs that transferred the pre-forwarded cache, namely content with *M-flag*. The forwarding is repeated until the packet with the *D-flag* arrives at the user end. The CRs that receive the packet with *D-flag* unset *M-flag* from all caches corresponding to the received *D-flag*. Subsequently, the caches are treated the same as common caches on NDN.

5.3 Advantage and Disadvantage of Proposal

5.3.1 Effects on Content Acquisition Delay

As described in the section4, PMC method has been proposed to cope with the publisher migration. However, on the PMC, all first requests for migration publisher's content must be forwarded to the *HomeNode* for reaching the new publisher location. This leads to increase unneeded delay. In addition, consumers could not get the mobile publisher's content before the publisher finish its migration, on the PMC.

On the other hand, our proposal method can deliver content without reaching old location (*HomeNode* in case of PMC). In addition, consumers can get the mobile publisher's content if the desired content is pre-forwarded to the neighbor CR of the mobile publisher, even before the mobile publisher finish its migration.

5.3.2 Overhead Induced by Proposal

As described in section 5.1.1, the *M-flag* process in the proposed method is not a method to immediately rewrite old FIB information on all CRs. Therefore, it is not a method that significantly increases the network load.

In addition, the effect of pre-forwarding the *M-flagged* content depends on the cache capacity of the neighboring CRs

Table 1: Simulation paprameters

Parameter	Value
Number of nodes	24
Interest generation rate	100 [request/sec]
Number of contents	1,000
Interest Packet size	1,024 [bytes]
Content Packet size	1,024 [bytes]
Cache capacity	infinity
Simulation length	50[sec]
Time to start publisher migration	10 [sec]
Time to finish publisher migration	40 [sec]

before migration of the mobile publisher. When the cache capacity is small, the effect is not high, but it does not degrade the performance of content delivery compared to conventional NDN that does not consider publisher migration.

On the other hand, the proposed method delivers *R-flags* to all CRs having users under them when a publisher migration is completed.

Of course, R-flags are delivered together in a single packet where routes for users overlap. Therefore, in a network where many users are connected to a small number of edge CRs, the load generated by the R-flag forwarding process is not a significant problem. However, in a network where many users are equally connected to many edge CRs, the load caused by *R-flag* processing becomes significant.

6 PERFORMANCE EVALUATION

To clarify the applicability of the proposed methods, this section evaluates the methods using computer simulation.

6.1 Effects of Ratio of Pre-forwarding Contents on Proposed Method

This section evaluates the effects of the ratio of pre-forwarding contents on the proposed method for content pre-forwarding and updating old FIB information.

6.1.1 Evaluation Environment

The evaluation environment is described in this section. Table. 1 lists the parameters used in the simulation.

In this paper, we assumed to use the NDN architecture in which mobile publisher publishes SNS messages consisting text-based information mainly (not the multimedia content). Hence, the size of the content will be delivered to the consumers are short in the evaluations. For the simplification of the evaluations, we used same value, 1024 byte as the length of the both Interest and Content. Although it would be better to use more accurate length of the both packets (at least, Content length is larger than Interest in terms of SNS message), length of the both packets does not influence the characteristics of the evaluation results because of we have mainly focused on content acquisition ratio.

Figure 11 shows the simulation topology. The number of publishers, CRs, and users are 1, 18, and 5, respectively. The mobile publisher starts to migrate from the neighbor of CR1

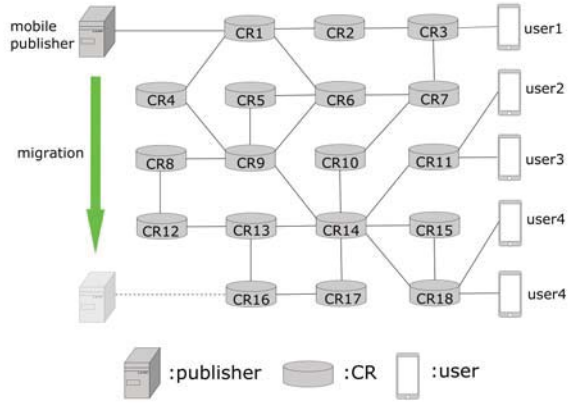

Figure 11: Evaluation topology

to the neighbor of CR16 during the simulation. Content requests arrive at all users at a rate of 100 [request/s].

The mobile publisher published 1,000 original contents. The content name recorded in *Interest* is randomly selected within the range. Pre-forwarding contents are also randomly selected at the beginning of publisher migration.

In the simulation, the ratio of pre-forwarding content, time to live (TTL) of *Interest*, and the capacity of the PIT are varied to clarify the characteristics of the proposed method. For the evaluation, the content acquisition ratio is derived as a value of the number of contents obtained divided by the number of contents requested.

6.1.2 Relationship between Content Acquisition Ratio and Ratio of Pre-forwarding Content

This section reports the characteristics of the content acquisition ratio under varying amounts of forwarding content. Values of TTL and PIT are 0.1 [s] and infinity, respectively.

Figure 12 presents the simulation results of three methods, namely pre-forwarding with *M-flag* control, pre-forwarding with both *M-flag* and *R-flag* control, and conventional as pre-forward(M), pre-forward(M/R), and Conventional, respectively.

In addition, Conventional implies the performance of original NDN. The figure shows that both pre-forwarding with *M-flag* and pre-forwarding with *M-flag* and *R-flag* increase the content acquisition ratio according to the amount of pre-forwarding content. In addition, pre-forwarding with both *M-flag* and *R-flag* always achieves a higher performance than that with only *M-flag*. The difference between the two methods is due to the fast rebuild of the *Interest* forwarding path by the *R-flag*.

6.1.3 Relationship between Content Acquisition Ratio and Length of TTL

This section reports characteristics of content acquisition ratio under varying lengths of TTL. For this evaluation, all contents are forwarded to the neighbor of the publisher at the beginning of the publisher?s migration.

Figure 13 presents the evaluation results, wherein the horizontal and vertical axes represent the length of TTL and con-

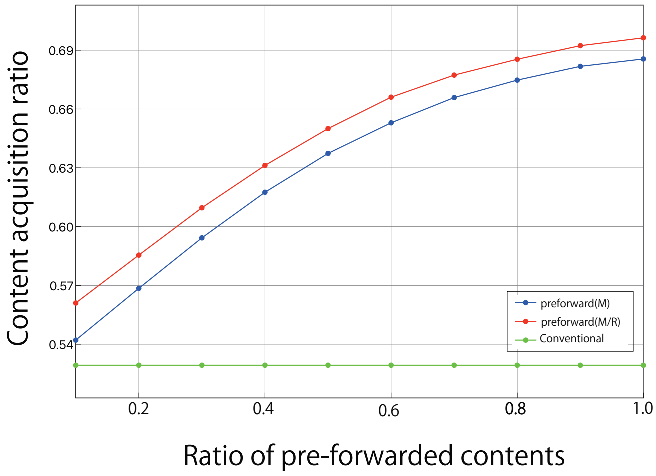

Figure 12: Characteristics of content acquisition ratio - amount of pre-forwarding content

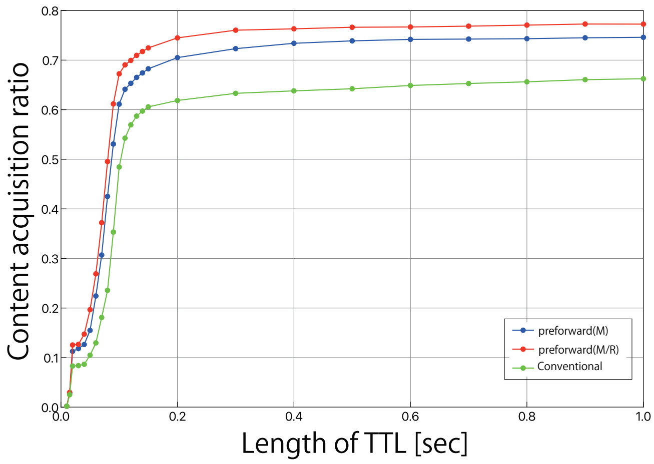

Figure 13: Characteristics of content acquisition ratio - Length of TTL.

tent acquisition ratio, respectively. The colored lines show the same mean as the previous figure.

All methods increase the content acquisition ratio with increasing TTL length. In addition, the increase in the content acquisition ratio becomes small when the TTL exceeds 0.1 [s]. This is because the average round trip time (RTT) for content acquisition is 0.2 [s]. Furthermore, if the TTL is more than the RTT, the content acquisition ratio did not increase.

As the figure shows, pre-forward(M/R) always maintains the highest performance compared to the other two methods.

6.1.4 Relationship between Content Acquisition Rate and PIT Capacity

This section evaluates the characteristics of content acquisition rate and PIT capacity. We use 0.1 [s] as the TTL on the evaluation.

Figure 14 presents the results of the evaluation. As shown, the content acquisition rate of all methods increases in accordance with the PIT capacity when the PIT capacity is smaller than 5. However, the increase in the content acquisition rate

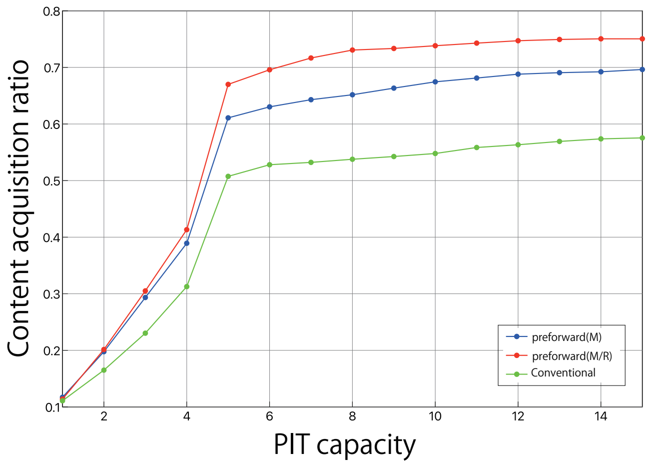

Figure 14: Characteristics of content acquisition ratio - PIT capacity.

Table 2: Simulation paprameters

Parameter	Value
Number of nodes	24
Interest generation rate	100 [request/sec]
Number of each publisher's content	100
Interest Packet size	1,024 [bytes]
Content Packet size	1,024 [bytes]
Cache capacity	100 [caches]
Simulation length	100 [sec]
Time to start publisher migration	20 [sec]
Time to finish publisher migration	80 [sec]

decreases when the PIT capacity exceeds 5.

As shown in this figure, pre-forward(M/R) always has an advantage over the other two methods.

6.2 Effects of Priority Cache Holding for Mobile Publisher's Contents

This section evaluates the effects of the second part of the proposal, namely the priority cache holding for mobile publisher's contents.

Simulation parameters used in this section are shown in Fig. 2. In this evaluations, we assume to use our proposal at the suburban area. At the mountain district, many points are outside the communication range due to the geographical features. Hence, network inaccessibility time by publisher migration will be longer. As shown in the table, the mobile publisher starts its migration at 20 [s] after the beginning of the simulation and finishes its migration at 80 [s].

Interest is generated at each user end every 0.01 [s]. In each *Interest*, one content name is selected randomly from the contents of two publishers regardless of the publisher's mobility (mobile or stationary). Before the migration of the mobile publisher, it pre-forwards its content to the neighbor CR (CR1 in this case). In addition, pre-forwarded contents are treated using the method of pre-forwarded cache holding.

Figure 16 present the results of content acquisition ratio under the varying amounts of pre-forwarding contents. As

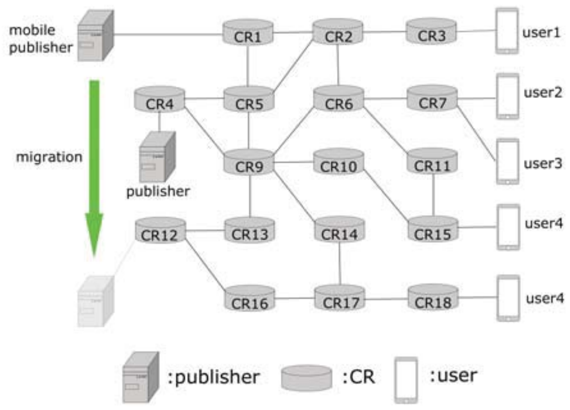

Figure 15: Topology for evaluation of the effects of priority cache holding

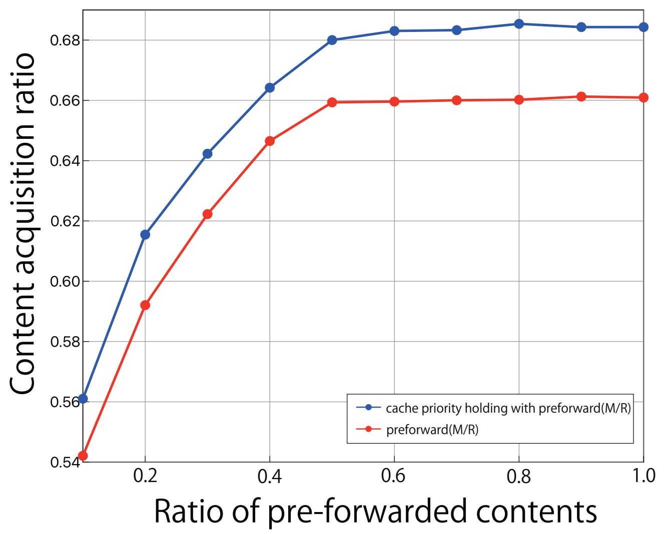

Figure 16: Relationship between content acquisition ratio and ratio of pre-forwarding content on priority cache holding method

shown, by adding the method of priority cache holding to the first part of our proposal (pre-forward(M/R)), higher content acquisition ratio is achieved, regardless of the ratio of pre-forwarded contents. By holding the cache of the mobile publisher's contents while migrating, users can obtain the mobile publisher's content. Moreover, the contents generated at the stationary publisher can be obtained directly from the publisher, even without a CR holding the cache of a stationary publisher. Hence, the priority cache holding of the mobile publisher has no negative effects in terms of contents acquisition ratio.

7 CONCLUSION

The development of ICT enables the publishing of content for delivering many network users, even by mobile devices. This makes it more flexible and adds more richness to the content generation activities. Moreover, ensuring content provisioning is gaining importance for such mobile content publishing activities.

This paper proposed solutions to deal with such issues by implementing new features on NDN that has received much attention as future network architecture. We proposed a method for content pre-forwarding and updating old FIB information effectively, which avoids degrading the content acquisition ratio. In addition, the priority cache holding of mobile publishers? contents is effective for further improvement of the content acquisition ratio. The applicability of the proposed methods is clarified by computer simulations.

In this study, we evaluated the performance of the proposed system in a scenario where the network contained one mobile publisher. However, in the real situation, multiple mobile publishers must be considered. In addition, it is obviously that our proposed content pre-forwarding method is only useful when the root CR, which means the next CR to the mobile publisher can store the additional cache during the publisher migration process. In this paper, in order to clarify the constitutive performance of the proposal, we have evaluated the methods under the situation that CR holding unlimited cache buffer. However, in the real situations, all CRs hold a finite cache buffer. Hence, we will continue to discuss detailed parameters for implementing our method, particularly for the priority cache-holding policy.

REFERENCES

[1] G. Tyson, A. Mauthe, S. Kaune, P. Grace, and T. Plagemann, "Juno: An adaptive delivery-centric middleware," Proc. of 2012 IEEE Consumer Communications and Networking Conference (CCNC), pp. 587-591 (2012).

[2] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, "Developing information networking further: From PSIRP to PURSUIT," in Proc. of Broadnets, pp. 1–13 (2010).

[3] B. Ahlgren et al., "Design considerations for a network of information," Proc. of CoNEXT, pp. 66 (2008).

[4] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking named content," Proc. of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp. 1–12 (2009).

[5] F. Fahrianto and N. Kamiyama, "Comparison of Migration Approaches of ICN/NDN on IP Networks," Proc. of 2020 Fifth International Conference on Informatics and Computing (ICIC), pp. 1–7, (2020).

[6] S. Luo, S. Zhong and K. Lei, "IP/NDN: A multi-level translation and migration mechanism," Proc. of NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–5 (2018).

[7] S. Malektaji, A. Ebrahimzadeh, H. Elbiaze, R. H. Glitho and S. Kianpisheh, "Deep Reinforcement Learning-Based Content Migration for Edge Content Delivery Networks With Vehicular Nodes," Trans. on IEEE Network and Service Management, Vol. 18, No. 3, pp. 3415-3431 (2021).

[8] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, "Proactive selective neighbor caching for enhanc-

ing mobility support in information-centric networks," Proc. of the second edition of the ICN workshop on Information-centric networking (ICN '12). pp.61–66 (2012).
[9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture, " SIGCOMM Comput. Commun. Rev., Vol. 37, No. 4, pp. 181–192, (2007).
[10] D. Han, M. Lee, K .Cho, T. Kwon and Y. Choi, "Publisher mobility support in content centric networks," Proc. of The International Conference on Information Networking 2014 (ICOIN2014), pp. 214-219 (2014).

(Received: October 30, 2021)
(Accepted: September 5, 2022)

Taichi Iwamoto received his Bachelor and Master degrees from Prefectural University of Hiroshima, 2020 and 2021, respectively. He has worked in the field of Content Centric Networking architecture.

Tetsuya Shigeyasu received his B.S. and M.S. degrees from Yamaguchi University, 2000, and 2002, and his Ph.D. degree from Osaka University in 2010, respectively. In 2022, he joined Hiroshima International University as a research associate. In 2011, he moved to Prefectural University of Hiroshima. He is currently a professor of the Department of Management Information Systems. He is a dean of Faculty of Management and Information Systems, and a dean of Faculty of Regional Development. His current research interests include computer networks, mobile networking, He is a member of IEEE, ACM, IEICE, and a senior member of IPSJ.