PAPER

Proposal and Performance Evaluation of Hybrid Routing Mechanism for NDN Ad Hoc Networks Combining Proactive and Reactive Approaches

Quang Minh NGO[†], *Nonmember*, **Ryo YAMAMOTO**[†], **Satoshi OHZAHATA**[†], *and* **Toshihiko KATO**[†a)], *Members*

SUMMARY In this paper, we propose a new routing protocol for named data networking applied to ad hoc networks. We suppose a type of ad hoc networks that advertise versatile information in public spaces such as shopping mall and museum. In this kind of networks, information providers prepare fixed nodes, and users are equipped with mobile terminals. So, we adopt a hybrid approach where a proactive routing is used in the producer side network and a reactive routing is used in the consumer side network. Another feature of the proposed protocol is that only the name prefix advertisement is focused on in the proactive routing. The result of performance evaluation focusing on the communication overhead shows that our proposal has a moderate overhead both for routing control messages and Interest packets compared with some of conventional NDN based ad hoc routing mechanisms proposed so far.

key words: *Ad Hoc Network, Named Data Networking, Proactive Routing, Reactive Routing*

1. Introduction

Recently, the Information Centric Network (ICN) is widely studied as a future Internet architecture well suited for large scale content distribution. The Named Data Networking (NDN) [1] is adopted widely as a platform for ICN research activities. The fundamental concept in NDN is the name of required content, not the address of hosts containing content. A consumer requesting a content sends an Interest packet containing the content name. A producer providing the corresponding content data returns a Data packet to the consumer. NDN routers transferring the Data packet cache the packet for future redistribution.

Originally, NDN is designed for wired network topology, but it can be effectively applied to wireless multi-hop ad hoc network topology. In wireless ad hoc network, the routing mechanism is more important research topic than wired fixed network, because network nodes move around. In NDN, the purpose of routing is how to construct Forwarding Information Base (FIB) for name prefixes, which specifies the correspondence between a name prefix and an interface called face (or a neighbor identifier) to the content with this name prefix.

There are several proposals on the routing in NDN. For the wired NDN topology, the proposed named OSPFN [2] and NSLR [3] are examples introduced in an early stage. Both of them are based on the link state routing protocol, which maintains and advertises link status between neighbors, shares the topology information, and creates routing tables from it. The protocol in [4] is a relatively new proposal based on the link state routing considering multipath routing.

In the case of the NDN based wireless ad hoc network, both the proactive and the reactive approaches are proposed [5]–[9]. This trend is the same as the IP based ad hoc network. MobileCCN [6] and TOP-CCN [7] are examples of the proactive routing mechanism. MobileCCN can be said an NDN version of Routing Information Protocol (RIP) [10]. TOP-CCN is an NDN version of Optimized Link State Routing (OLSR) [11]. On the other hand, E-CHANET [8] and REMIF [9] are examples of reactive routing mechanism, which are designed based on Ad Hoc On-Demand Distance Vector routing (AODV) [12].

These NDN based ad hoc routing mechanisms have pros and cons. The proactive routing can create FIB responding to an up-to-date network topology, but has some overhead of routing control message exchange. On the contrary, the reactive routing has no overhead of routing, but has some overhead of Interest packet transfer.

We target at ad hoc networks providing various useful information in public spaces, such as station, shopping mall and museum. Content providers advertise helpful information for users, such as location map, advertising catalog, and exhibition details. In this kind of ad hoc networks, a lot of users share information from providers, and this means that it is highly possible that users contain the same information. This situation allows the Data packet caching in the NDN technologies to realize efficient information sharing among users.

In order to provide a routing in those NDN ad hoc networks, we proposed a new mechanism in our previous papers [13], [14], which is based on the following two approaches. First, in the type of ad hoc networks we suppose, a content producer side has a stable network where producers and intermediate routers are located in fixed positions. On the other hand, consumers are mobile nodes which change their locations quite often. Therefore, we take a hybrid approach that a proactive routing is adopted in a producer side network, because of its in-advance route setting, and a reactive routing is adopted in a consumer side network, because

---

Manuscript received December 25, 2018.
Manuscript revised April 6, 2019.
Manuscript publicized June 18, 2019.

[†]The authors are with University of Electro-Communications, Chofu-shi, 182–8585 Japan.
a) E-mail: kato@is.uec.ac.jp

DOI: 10.1587/transinf.2018EDP7440

of its flexibility for mobility.

The second is about the procedure of proactive routing. The NDN proactive routing procedures proposed so far [2]–[4], [6], [7] are focusing on advertising both the network topology and the name prefix. However, the point of NDN routing is how the name prefixes are disseminated. In order to realize this requirement, it is sufficient that the shortest path information is maintained only for individual producer. So, our proposal is a new proactive NDN routing focusing on just the name prefix advertisement.

Although the basic idea was presented in our previous papers, the performance evaluation results described in those papers are limited. The problems include the followings. Our previous papers assumed very simple radio transmission where only the distance decides the receipt or non-receipt of radio. [13] does not consider the node movement, and [14] considers very simple moving model where only one consumer moves in a linear direction. They do not consider MAC layer retransmission. So, the performance evaluation with more realistic conditions is required.

This paper describes the detailed design of the proposed routing protocol, with extending our previous papers by introducing detailed data structures, and the results of the performance evaluation focusing on the routing control and Interest transfer overheads. The performance evaluation is conducted with ndnSIM [15], a widely used NDN simulator implemented over the ns-3 network simulator [16]. The ndnSIM simulator is originally designed for wired network configurations, and need to be extended for supporting wireless ad hoc networks. For example, a neighbor node is identified by face for the neighbor. However, in the case of wireless ad hoc network, it is common that the face in a node is just one (a WiFi interface) and that a neighbor node needs to be identified by its identifier (e.g., MAC address). This paper also discusses how to implement NDN wireless ad hoc network over ndnSIM.

The rest of this paper consists of the following sections. Section 2 describes the related work on NDN, NDN based ad hoc routing, and implementation of NDN wireless ad hoc network. Section 3 proposes our new protocol. Section 4 gives some discussions on how to implement an NDN based wireless ad hoc network. Section 5 shows the implementation of the proposed protocol over ndnSIM and the results of the performance evaluation. Section 6 concludes this paper.

2. Related Work

2.1 Overview of Named Data Networking

NDN nodes (consumers, NDN routers and producers) maintain the following three major data structures [1].

- Forwarding Interest Base (FIB): used to forward Interest packets toward producers of matching Data.
- Pending Interest Table (PIT): keeps track of Interest packets forwarded to producers so that returned Data packets can be sent to consumers.
- Content Store (CS): caches received Data packets temporarily.

When an Interest packet arrives on some face, the content name in the Interest is looked up. If CS has a copy of Data packet exactly matching the received content name, it will be sent out to the face the Interest packet arrived on and the Interest packet will be discarded. Otherwise PIT is examined and, if there is a PIT entry exactly matching the received content name, the Interest's arrival face will be added to the PIT entry and the Interest packet is discarded. Otherwise FIB is examined and, if there is an FIB entry matching the corresponding name prefix, then the Interest packet will be sent to the face specified in the FIB entry.

As described above, the routing mechanism in NDN is a procedure to create FIB entries for published name prefixes. As for the routing in wired NDN topology, proposed protocols so far [2]–[4] are based on Open Shortest Path First (OSPF) [17], which is a widely used link state based intra-domain routing protocol. Among them, Named-data Link State Routing protocol (NLSR) [3], for example, introduces two types of link state advertisements (LSAs): Adjacency LSA and Prefix LSA. An Adjacency LSA is similar to LSA defined in OSPF and contains a list of neighbor name and cost of the link to neighbor. A Prefix LSA is designed for NDN and contains name prefixes. An NDN node sends Periodic "info" Interest packets for neighbor detection. If it receives an "info" Content reply, it considers that the neighbor is alive. An NDN node also sends periodic "Root Active" Interest packets. If any link state information has changed, its reply is returned. After that, an Interest packet requesting a new LSA and its corresponding Data packet are exchanged. Based on the information available in the Adjacent LSAs, each node builds a network topology, from which next-hops for each destination are derived. From the Prefix LSAs, the name prefixes associated with each destination are obtained. From them, the FIB entries are constructed.

2.2 NDN Based Ad Hoc Routing Mechanisms

For NDN based ad hoc network, there are a lot of research activities on routing protocols [5]. Among them, MobileCCN [6] and TOP-CCN [7] are typical examples of a proactive routing mechanism. In MobileCCN, NDN nodes regularly broadcast their own FIB, obtain neighbors' FIB, and re-create own FIB. The idea is similar to that of RIP, in which routers send their own routing table to their neighbors periodically [10]. As is in RIP, the scalability is a problem in MobileCCN.

TOP-CCN is an application of OSLR [11] to NDN based ad hoc routing. TOP-CCN introduces a new packet called Content Announcement (CA) packet. It also introduces the idea of multipoint relay (MPR) and publisher MPR (PMPR). A CA packet contains name prefixes, node identification (ID) and type of sender, list of neighbors' ID and type and so on. It is used for neighbor discovery/MPR selection, through single hop broadcast, and for link state information announcement, through multi-hop flooding. A multi-hop CA packet is generated by PMPR and flooded by MPRs and PMPRs, and it is used to create a network topology and FIB. Since the base of TOP-CCN is OLSR used in IP networks, however, multi-hop CA packets provide over-specified information. For example, a route between consumers, which is never used in NDN, can be obtained from this information.

On the other hand, a reactive routing mechanism is original in an ad hoc network. There are many examples [5], including REMIF [9], which we use in the performance evaluation. REMIF does not use any routing control messages and therefore NDN nodes do not maintain FIB. Instead, route to producer is detected at flooding Interest packets. In order to avoid a broadcast storm problem, REMIF adopts differed re-broadcasting with remaining energy checking. Although REMIF has better performance than E-CHANET [8] as for the Interest forwarding overhead [9], the overhead may increase depending on the node density and the average hops between consumers and producers.

2.3 Implementation of NDN Based Ad Hoc Networks

As described above, the original NDN is designed for wired networks, especially for point-to-point link based networks. Specifically, a neighbor node is identified by a face to the node, and Interest and Data packets are forwarded from one face to another face. Moreover, the neighbor nodes are supposed not to change so often. On the other hand, wireless NDN networks, especially a wireless ad hoc NDN networks need to have different features. At first, in the case of wireless NDN, there is only one network face installed in a node. Interest and Data packets to and from neighbor nodes are transferred through this face and the neighbor nodes are identified by their IDs such as MAC addresses. The packets received from the face are retransmitted through the same face to next neighbor nodes. Moreover, neighbor nodes will be changed according node movement.

In order to implement NDN ad hoc networks, the above mentioned features need to be considered in the implementation framework. In ndnSIM, a widely used NDN simulator, however, the environments for multi-hop wireless ad hoc networks are not prepared. The papers proposing wireless NDN mechanisms do not describe how to implement the proposals clearly, ether.

The paper proposing TOP-CCN [7] states that it uses ndnSIM for evaluating TOP-CCN, but there are no descriptions on its implementation. E-CHANET is said to be implemented over ns-2 [8], but no descriptions are given about the implementation details. REMIF is also said to be implemented over ndnSIM [9], but no descriptions are given either. [18] provides a performance comparison of an OLSR based ad hoc network and an NDN based ad hoc network. It uses an implementation using ns-3 and ndnSIM, but the implementation details are not described.

Recently, there are other proposals on NDN based ad hoc networks [19]–[21]. [19] proposes a NDN based ad hoc network, which is built over IEEE 802.11 wireless LAN. The proposed mechanism in this paper uses MAC address of NDN nodes, and adds the fields of MAC address and MAC address list in Interest and Data packet. Although such an approach allows nodes to use MAC addresses in Interest and Data packet transfer, the added fields are meaningful only in wireless networks. So, the MAC address related mechanisms need to be realized in the strategy layer in the NDN protocol stack architecture. [20] is another example of NDN wireless network proposed by the same authors as [19]. This paper proposes an NDN based ad hoc network, which is based on the location and remaining energy of nodes. In this proposal, the position of nodes is included in Interest and Data packets, which is a similar approach with [19]. We think that the node location information need to be also handled in the strategy layer. [21] is an example of NDN based vehicular ad hoc network (VANET), in which the MAC addresses of nodes are also included in Interest and Data packets.

3. Proposal

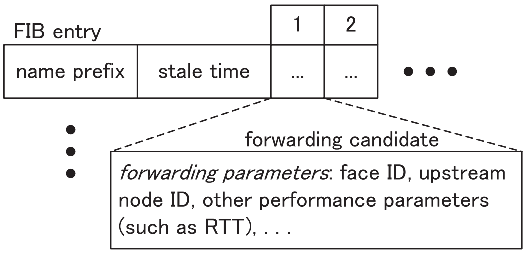
3.1 Design Principles

We have adopted the following design principles for our hybrid NDN based routing mechanism.

- As described above, we divide a whole NDN network into the producer side and the consumer side. In the producer side, NDN nodes including producers and intermediate routers have their location fixed. So, a proactive routing mechanism is introduced in this part. On the other hand, the consumer side includes mobile nodes working as consumers or intermediate routers. Those nodes move around and the network configuration often changes. In this part, a reactive routing mechanism is introduced.
- For the producer side, our proactive routing focuses only on name prefix advertisement. It constructs a directed acyclic graph (DAG) starting from each producer. An FIB entry for a specific name prefix is given by pointing upstream nodes traversing the corresponding DAG in a reverse direction. If there are more than one upstream nodes, both of them are registered in the entry and used for multipath forwarding [22].
- In order to create a DAG for a specific name prefix, the corresponding producer issues a *Name Prefix Announcement Request* (*NPAreq*) packet. It is broadcasted, and if any receiving NDN nodes are on the corresponding DAG, they return a *Name Prefix Announcement Reply* (*NPArep*) packet by unicast.
- As for the consumer side, NDN nodes do not use any control packets for routing. Instead, the FIB entry is created by the first Interest packet for a name prefix. The first Interest packet is flooded throughout the con-

Table 1 Parameters in NPAreq and NPArep packets.

packet	parameters
NPAreq	producer node ID, producer nonce, name prefix list, hop count, number of child nodes.
NPArep	producer node ID, nonce.



note: forwarding candidates ranked by number of child nodes managed by upstream node or by other routing policies

Fig. 1 Structure of FIB at producer side.

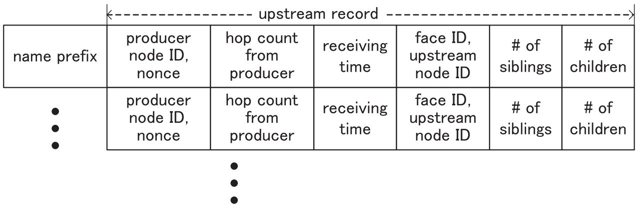


Fig. 2 Structure of DAG table at produce side.

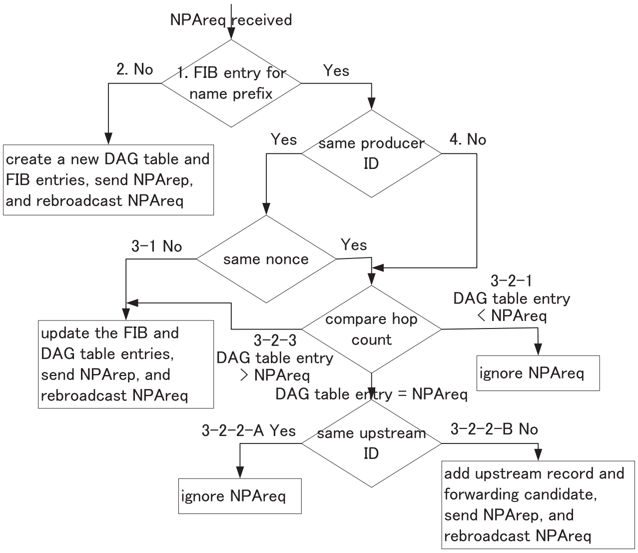


Fig. 3 Flow chart for a received NPAreq packet.

sumer side, and after it reaches some node in the producer side, this Interest packet is transferred to the producer. When the corresponding Data packet returned, a temporary FIB entry is created at the nodes in consumer side. For the following Interest packets for the same name prefix, this FIB entry is used.

3.2 Detailed Design for Producer Side

Table 1 shows the parameters contained in NPAreq and NPArep packets. *Producer node ID* is the MAC address of the producer node, and NPAreq and NPArep packets can be uniquely identified using this ID and *nonce* assigned by the producer. A producer generates NPAreq packets periodically, which contains the *name prefix list* that it is publishing. *Hop count* is the number of hops from the producer which generated this NPAreq packet. When a producer side node receives an NPAreq packet, it rebroadcasts the received packet with incrementing hop count and setting the *number of child nodes* (nodes located downstream in a DAG), and return an NPArep packet to the sender of the NPAreq packet. The number of child nodes is used for ranking upstream nodes in an FIB entry. The detailed procedure for specifying FIB are described below.

Figure 1 shows the structure of FIB used by producer side nodes. The structure is similar with that of the original NDN specified in [22]. An FIB entry is created for individual name prefix, and it may contain multiple forwarding candidates. Each candidate has the forwarding parameters, which include the interface ID, ID of neighbor node (upstream node in a DAG), and other parameters such as RTT. More than one forwarding candidates may be ranked according to some routing policies. In the proposed method, a node with more child nodes (downstream nodes) has higher priority. We selected this policy because the possibility of using Data packet caching will be higher in the node with more downstream nodes.

In order to construct FIB entries, the proposed method maintains the DAG table shown in Fig. 2. This table maintains the information given in NPAreq packets. An entry corresponds to one name prefix and includes one or more upstream records, each of which includes a list of producer ID, its nonce, hop count from the producer, the face ID and upstream node ID, and the number of sibling nodes and child nodes. These upstream records are ordered according to the hop count from the producer, and if the hop count is the same, according to the number of sibling nodes.

A node receiving an NPAreq packet follows the procedure given below and in Fig. 3.

1. Check whether there is an FIB table entry for the name prefix specified in the received NPAreq packet.
2. If there are no such entries, add a new FIB entry and a new DAG table entry with the MAC address of the sender of the NPAreq packet set in the upstream node ID. Send an NPArep packet to the NPAreq sender, and rebroadcast the NPAreq packet.
3. Otherwise, check whether there is an upstream record in the corresponding DAG table entry which has the same producer node ID. If there is such an upstream record, then look for records in which the nonce is the same as that in the NPAreq packet.

(3-1) If there are no such records, handle this NPAreq as a new advertisement. That is, delete the upstream record corresponding to the producer node ID and nonce pair in the DAG table entry, and delete if the list becomes empty, delete the forwarding candidate, if there are any in the corresponding FIB entry. After that, add a new forwarding candidate and a new upstream record when necessary. Send an NPArep packet to the NPAreq sender, and rebroadcast the NPAreq packet.

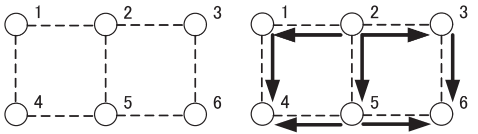
- (3-2) Otherwise, that is, when there are some upstream records having the same pair of producer node ID and nonce with the NPAreq packet, compare the hop count in the record with that in the NPAreq.
    - (3-2-1) If the hop count in the record is smaller, then ignore the received NPAreq packet.
    - (3-2-2) If two hop counts are the same, then check whether there are any upstream records which have the upstream node ID identical to the NPAreq sender address.
        - A) If there is such a record, ignore the received NPAreq packet.
        - B) Otherwise, that is, when the NPAreq is sent by a new upstream node, add a new upstream record in the DAG table entry, and a new forwarding candidate in the FIB entry, and return an NPArep and rebroadcast the NPAreq. This is for multipath forwarding.
    - (3-2-3) Otherwise, that is, when the hop count in the upstream record is larger than that in NPAreq packet, handle this NPAreq as a new advertisement. Act as in step (3-1).

4. Following the first part of step 3, the last step is for when there are no candidates with the producer node ID specified in the NPAreq packet, that is, when an NPAreq with the same name prefix from a new producer. In this case, compare the hop count in the upstream record with that in the received packet, and act in the same way as (3-2-1) through (3-2-3) according to the result.

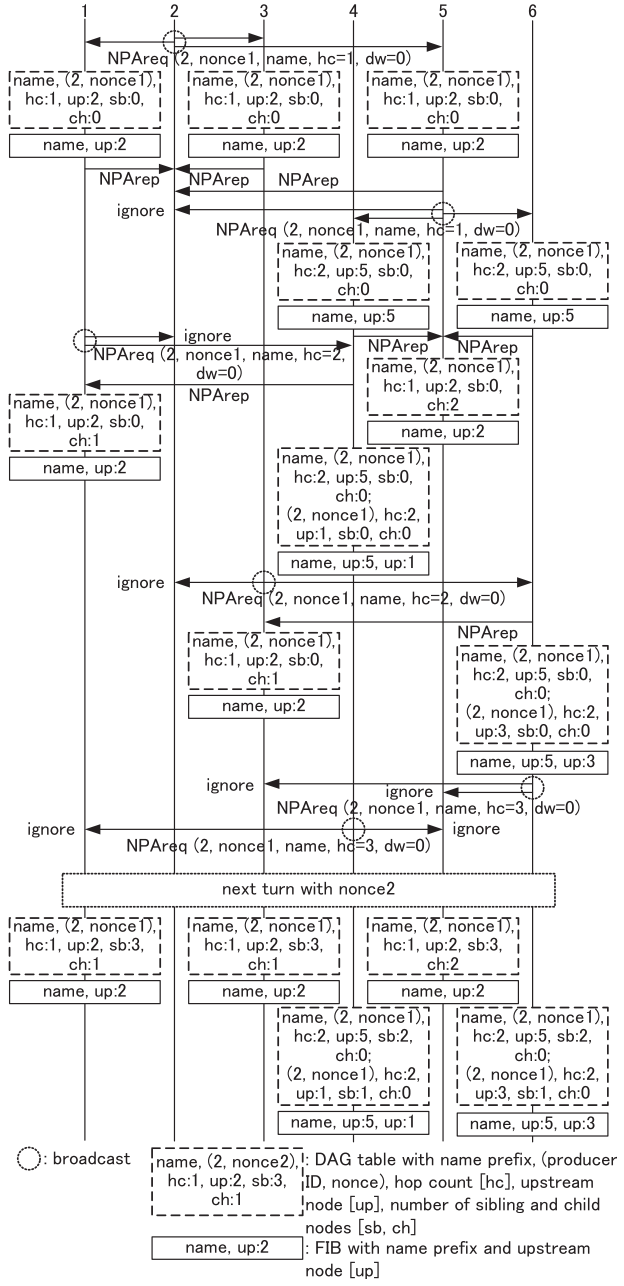
In any step where some upstream record is created or modified, the number of downstream nodes managed by upstream node needs to be modified according to the received NPAreq packet.

When a node receives an NPArep packet, it looks for an upstream record with the producer node ID and nonce in the packet, and increments the number of child nodes managed by this node by one.

Figure 4 shows an example of how this protocol works. As shown in Fig. 4 (a), there are six producer side nodes connected with wireless links shown in dashed lines. Among them, node 2 is a producer and the others are NDN routers.



(a) example network (c) generated DAG



(b) sequence of NPAreq and NPArep exchange

Fig. 4 Communication sequence at producer side.

As shown in Fig. 4 (b), in the beginning, node 2 broadcasts an NPAreq packet with producer node ID = 2, nonce1, “name”, hop count = 1, and number of downstream nodes = 0. Nodes 1, 2, and 5 receive this packet, create an FIB entry and a DAG table entry as shown in the figure, and return an NPArep packet individually. Then node 5 rebroadcasts the

NPAreq packet with changing hop count to 2, and nodes 4 and 6 respond it. Node 2 receives the packet but ignores it. When node 5 receives the NPArep packets from node 4 and 6, the number of child nodes in this node is set to 2.

Next, node 1 rebroadcasts the NPAreq packet, to which node 4 responds. As a result, the FIB entry and the DAG table entry in node 4 have two forwarding candidates and two upstream records to node 1 and 5. Similarly, the NPAreq packet rebroadcasted by node 3 is handled by node 6. In the end of this advertisement, the NPAreq packets are rebroadcasted by nodes 4 and 6, but nobody responds to them. The generated DAG is shown in Fig. 4 (c).

After some periods, node 2 broadcasts a new NPAreq packet with nonce2. After this new NPAreq packet is disseminated, the FIBs and the DAG tables of individual nodes are set as shown in the figure. It should be noted that the FIBs in nodes 4 and 6 have two forwarding candidates with node 5 and nodes 1/3 as upstream nodes, respectively. These candidates are ranked by the number of downstream nodes managed by upstream node (“ch”). Since node 5 has two child nodes, the forwarding candidate to node 5 is ranked first.

So far in this subsection, we do not mention PIT in producer side nodes. The PIT structure in producer side nodes is identical to that used in original NDN nodes [22], except that the interface ID is replaced by the pair of the interface ID and the neighbor node ID (MAC address). This will be discussed in the next section.

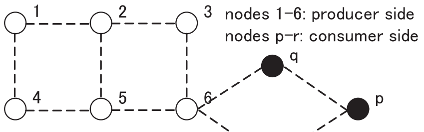
3.3 Detailed Design for Consumer Side

As described above, the NDN nodes in the consumer side network are mobile terminals, which move around. So, we introduce a reactive routing mechanism in the following way.

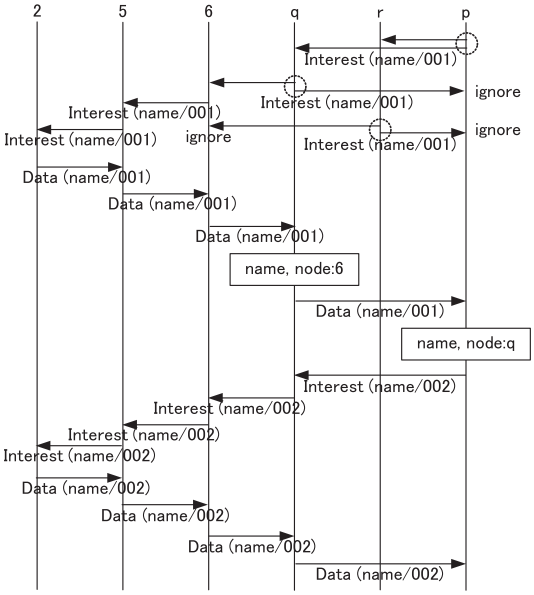
First of all, FIB is not set in the consumer side beforehand. When a node starts to retrieve a specific content, the first Interest packet for the content is flooded among consumer side nodes. When an Interest packet reaches some producer side node, it will be transferred to the corresponding producer. The producer sends back the Data packet containing the requested content. It is transferred through the reverse path of the Interest packet. When it goes through the consumer side nodes, FIB entry is set in individual nodes. The following Interest packets accessing to this name prefix use the FIB arranged. For the consumer side, we use the original formats of Interest and Data packets and the original structures of FIB and PIT, except that the first Interest packet is broadcasted and that a neighbor node MAC address is used as an interface ID.

Figure 5 shows an example of the communication sequence between a mobile consumer and a producer. As shown in Fig. 5 (a), the producer side nodes are the same as in Fig. 4 (a), and there are three consumer side nodes (nodes p, q, r). The dotted line shows a wireless link.

We assume that the FIBs are arranged in the producer side nodes. As shown in Fig. 5 (b), node p starts contest retrieval for name prefix “name” and the first Interest is for “name/001”. The Interest packet is broadcasted and nodes q and r receive it. Then node q rebroadcasts the Interest packet, and nodes 6 and p receive it. Node p ignores this Interest, because it is a duplicate one. Node 6 relays the received Interest packet to node 5 according to its FIB. On the other hand, node r also rebroadcasts the Interest packet, which nodes 6 and p receive. But both nodes ignore this Interest because of the duplication.



(a) example network



(b) communication sequence

Fig. 5 Communication sequence between consumer and producer.

The Interest packet is sent to node 2, producer, via node 5, and in response to it, the Data packet containing the content of “name/001” is returned along the reverse path of the Interest packet. That is, the Data packet goes via nodes 5, 6, and q, and reaches node p. When node q relays the Data packet, it creates an FIB entry for “name” which indicates that the upstream node is node 6. Similarly, when node p, consumer, receives this Data packet, it creates an FIB entry for “name” indicating that the upstream node is node q. For the following Interest packets, nodes p and q use the created FIB. That is, the next Interest packet requesting content for “name/002” is sent to node q in the unicast communication. Similarly, node q relays this Interest to node 6 directly.

When some nodes move and the communication link is broken, the Data packet is not returned and the timer for Interest packet will be expired. At that time, node p, consumer, will broadcast the lost Interest packet, and the similar procedure with the first Interest is performed.

4. Discussions on Implementing NDN Wireless Ad Hoc Networks

As described above, how to implement NDN based wireless ad hoc networks is not discussed well. One of the major reasons will be that, in a wireless ad hoc network, a neighbor node needs to be identified by its MAC address and that the neighborhood relationship will be changed according to the node mobility. Another reason is that one interface that is used for receiving Interest/Data packets is also used for re-sending them to another neighbor, as suggested in [18].

In this paper, we show two possible approaches for implementing NDN based wireless ad hoc networks over the ndnSIM simulator. As stated in the original proposal on NDN [1], it is possible that the content transfer function using Interest and Data packets (*content chunk layer*) works over TCP/IP protocol stack. Figure 6 shows a network architecture of NDN ad hoc network, where the UDP socket interface (the UdpFace class in ndnSIM) is used. Nodes (a consumer, a producer and an intermediate route) is connected via an ad hoc mode WLAN, and each node is assigned with an IP address. The neighbor node detection and the route establishment are realized by some ad hoc routing (OLSR in the figure). Interest and Data packets generated at the content chunk layer (the Forwarding-Strategy class in ndnSIM) are sent directly to their destinations through UDP/IP sessions. That is, the NDN intrinsic mechanisms, such as the suppression of Interest packet transfer by PIT and the use of Data packets in CS, are not performed in intermediate nodes. We use this implementation approach in the evaluation of OLSR based NDN ad hoc network in the next section.

Another approach is handling WLAN communication within the content chunk layer. The following are the outline of the implementation of wireless ad hoc network over ndnSIM.

- The content chunk layer (the ForwardingStrategy class) is located on top of IEEE 802.11 wireless LAN.
- A pair of face and MAC address is used to identify a neighbor node in the PIT and FIB handling.
    - ➢ In the PIT handling, a MAC address is added in the classes of incoming face and outgoing face which are stored in a PIT entry. In the PIT entry look up for selecting downstream node to pass a Data packet, the MAC address of the Data packet sender is checked together with the incoming face ID.
    - ➢ In the FIB setting, a MAC address is also kept in an FIB entry. When an Interest packet is received and its name is examined in the FIB, the MAC address of an upstream node is obtained together with the outgoing face ID.
- MAC addresses are handled in the WiFi network device (WifiNetDevice) class, and the ForwardingStrategy class cannot handle them. In order to allow the ForwardingStrategy class to handle MAC addresses, a remote MAC address is added in the Node class which maintains general information of a network node. When a WLAN frame is received, its source address is set in this field at the WifiNetDevice class and can be accessed in the ForwardingStrategy class. When a WLAN frame is sent, its destination MAC address is set at the ForwardingStrategy class and is reported to the WifiNetDevice class.
- In the Interest packet forwarding in the ForwardingStrategy class, the check whether the incoming face and the outgoing face obtained from FIB is the same is disabled.

We use the second implementation approach in the evaluation of the proposed method and REMIF in the next section.

5. Performance Evaluation

In this section, we describe the results of performance evaluation using the ndnSIM simulator version 1.0.

5.1 Simulation Conditions

Figure 7 shows the network configuration used in the simulation. In the fields of 300 m by 200 m, four producer side nodes are located in a grid configuration with 100 m distance. The location of these nodes are fixed through a

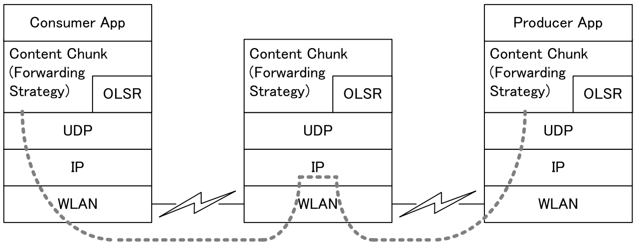


Fig. 6 Communication sequence between consumer and producer.

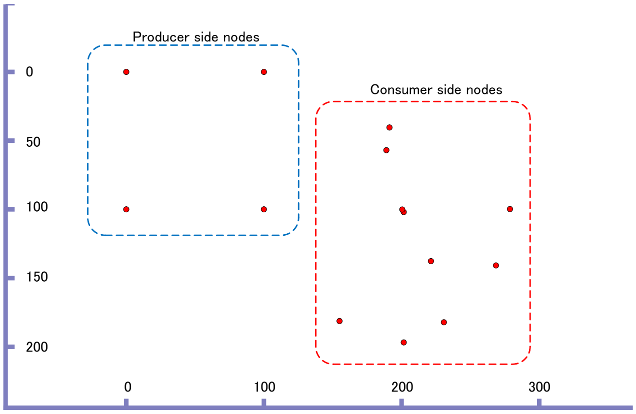


Fig. 7 Network configuration for simulation.

Table 2 Simulation parameters.

Radio propagarion	constant speed propagation delay model; three log distance / Nakagami propagation loss model
Wifi data mode	OFDM rate 24 Mbps
Mobility model	Random work 2D model; course change at every 2 second; mobility speed = 40 m/s, 20 m/s, or 10 m/s
Consumer	Randomly selected two, four, six and eight nodes; request different content/the same content; 10.0 through 10.3 Interests/s
Producer	node at (0, 0); Data packet size = 1200 bytes
Cache size	1000 packets at each node
Evaluation	15 seconds for each simulation run; 10 seconds for Interest packet origination, and 5 seconds just for timeout retransmission

simulation. In addition, ten consumer side nodes are deployed randomly with the center of (200, 100). These nodes move around according to a random walk model. All nodes communicate with each other through ad hoc mode IEEE 802.11a protocol.

The details of simulation condition are given in Table 2. As for the radio propagation, we used a setting used commonly in the ns-3 simulator. The data rate in IEEE 802.11a is 24 Mbps constant. The consumer side nodes move around according to the 2 dimensional random walk model with the constant mobility speed, where nodes change their direction at every 2 second. We adopted the mobility speed of 40 m/s, 20 m/s, and 10 m/s. Those values are large as a moving speed of human, but they are adopted for changing the wireless connection during a 15 second simulation run. Among the producer side nodes, the node located at the position (0, 0) works as a producer. As for the consumer side, two, four, six, or eight nodes work as consumers requesting different content or the same content. If each consumer requests different content, the Data packet caching is not effective in the simulation. If the same content is used for all consumers, the caching will be used effectively.

5.2 Evaluated Methods and Their Implementation Details

The methods evaluated in this section are the proposed method, REMIF (simplified version), and NDN over UDP/IP ad hoc network with OLSR routing (OLSR based NDN). OLSR based NDN is used in order to estimate the performance of TOP-CCN, because the exchange of Hello and TC (Topology Control) messages corresponds to that of CA packets in TOP-CCN. On the other hand, OLSR based NDN uses the IP based routing in intermediate nodes as shown in Sect. 4, and even if all consumers request the same content, the Data packet caching is not effective. So, when the same content is used, OLSR based NDN can be used to estimate an IP based ad hoc network.

The following describe the details of the implementation of three evaluated methods.

(1) REMIF

FIB is not specified, and Interest packets are always transferred with the destination address set to broadcast MAC address (“ff:ff:ff:ff:ff:ff”).

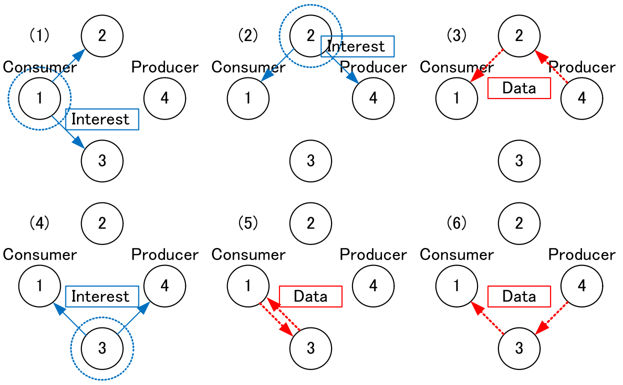


Fig. 8 Mis-ordering in REMIF.

On the other hand, PIT is used for returning Data packets to consumers. When a new Interest packet is received, the incoming face and the source MAC address of the Interest packet is stored in a new PIT entry. Since it is possible that the identical Interest packet is received via a different path, the duplication is detected by the Interest nonce stored in this PIT entry. A retransmitted Interest packet from a consumer contains the same nonce as the original Interest packet. In order to handle retransmitted Interest packets properly, a PIT entry for which a Data packet is not returned needs to be discarded when its lifetime expires. The lifetime of a PIT entry is set to the lifetime of Interest packet, 500 msec in this evaluation.

Since REMIF uses the broadcast in transmitting Interest packets, we observed a mis-ordering problem given in Fig. 8. Node 1 (a consumer) broadcasts an Interest packet in step (1) and node 2 rebroadcasts it in step (2). Then, node 4 (a producer) sends a Data packet to node 2, which transfers it to node 1 in step (3). After that, node 3 rebroadcast the Interest packet that is received in step (1). Since the PIT entry in nodes 1 and 4 are erased in step (3), they try to send the corresponding Data packet (node 1 has a cache for this Data packet). The Data packet is sent by node 1 to node 3, and node 3 returns it to node 1 again in step (5). Node 4 sends the Data packet again to node 3, which transfers it to node 1 in step (6). In both cases, the received Data packet is ignored because there are no corresponding PIT entry in node 1.

In order to avoid such a problem, we took the following way. In the PIT handling in the ForwardingStrategy class, when a Data packet is received, the records for incoming faces and outgoing faces are cleared, and then the PIT entry is erased by setting the PIT entry pruning timer. In the default, this value is set to 0 and the PIT entry is removed instantly. In this evaluation, we set this timer value to 50 msec. This means that our implementation ignores duplicate Interest packets received during 50 msec from the Data packet handling.

(2) Proposed method

In the performance evaluation here, we focus on the protocol behavior and the routing overhead when consumer side nodes move around. So, as for the routing protocols for producer side nodes, we set the FIB by hand before simulation runs start.

We implemented the FIB handling behavior in consumer side nodes by extending the REMIF program described above. At first, when a consumer side node receives an Interest packet, it looks for an FIB entry matching the name prefix included in the Interest packet. If there are no entries, it creates a new entry for the name prefix with the default face and the broadcast MAC address. A consumer side node transmit the received Interest packet according to the corresponding FIB entry.

When a consumer side node receives a Data packet, it registers the face from which the packet is received and the source MAC address of the data frame containing the Data packet in the corresponding FIB entry, if the MAC address in the entry is the broadcast MAC address.

When the network configuration of consumer side nodes changes, the FIB needs to be reconstructed. We implemented this mechanism in the following way.

- In order to detect the route change in consumer side nodes, we use the PIT entry pruning timer described above. When this timer is expired, the incoming and outgoing faces in the PIT entry examined. If they remain in the entry, we can decide that the Data packet corresponding to an Interest packet is not returned. These checks are executed in the PIT related class (the PitImpl class, specifically).
- If this timeout occurs consecutively (three times in our implementation), we decide that the route change occurs. Then, the outgoing face in the PIT entry is checked and, if the outgoing face has a unicast MAC address, the routine for clearing FIB entry in the ForwardingStrategy class is called.
- In the clearing FIB entry routine, the MAC address is set to the broadcast MAC address.

(3) OLSR based NDN

The OLSR based NDN method is implemented as the first approach described in Sect. 4. We can use the OlsrHelper class supported in the ns-3 simulator and the IpFaceHelper supported in the ndnSIM simulator. It should be noted that the calling of “Bing()” in the “CreateOrGetUdpFace()” method in the IpFaceStack class needs to be commented out, in ndnSIM version 1.0.

5.3 Evaluation Results

(1) Overview

We conducted three kinds of performance evaluation. The first is that using two consumers by changing the mobility speed. The second is that changing the number of consumers from two to eight with 20 m/s mobility speed. In these evaluations, individual consumers retrieve their own content, that is, no cache mechanisms are used. The third one is that where all consumers request the same content. In this case, cache mechanism is effective for REMIF and the proposed method. The conditions of the third evaluation is similar with that of the second evaluation.

In the evaluation for REMIF and the proposed method, we evaluated the following features, by changing the mobility speed of consumer side nodes or the number of consumers:

- the total number of Interest packets originated from consumers,
- the total number of Interest packets actually sent from consumers (including retransmissions),
- the total number of Data packets consumers received,
- the total number of forwarded Interest packets by all nodes, and
- the total number of forwarded Data packets by all nodes.

In the evaluation for OLSR based NDN, we evaluated the following features:

- the total number of Interest packets originated from consumers,
- the total number of Interest packets actually sent from consumers (including retransmissions),
- the total number of Data packets consumers received, and
- the total number of Hello and TC messages used in OLSR.

As for the sending interval of Hello and TC messages, we selected 0.5 sec and 1 sec, respectively. In order to establish routing information in the evaluation of OLSR based NDN, we introduce 5 second period before starting the content retrieval. In other word, simulation runs for OLSR based NDN take 20 seconds, consisting of 5 seconds for routing information setting, 10 seconds for Interest packet origination, and 5 seconds for timeout retransmission.

(2) Results of evaluation by changing mobility speed

Figures 9 through 11 show the results of the first performance evaluation. In the following figures, we normalize the number of packets by the total number of Interest packets originated from consumers. By adopting this normal-

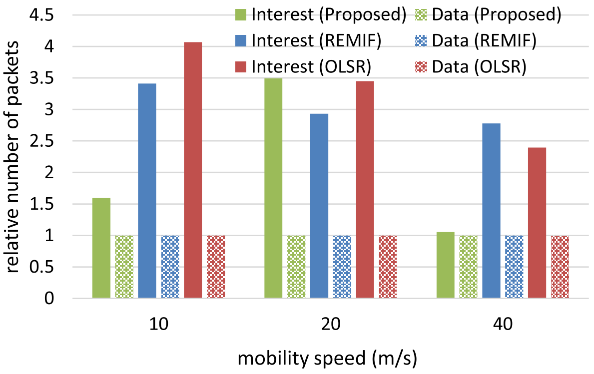


Fig. 9 Numbers of Interest packets actually sent from consumers and Data packets received by consumers (normalized by originated Interests; changing mobility speed).

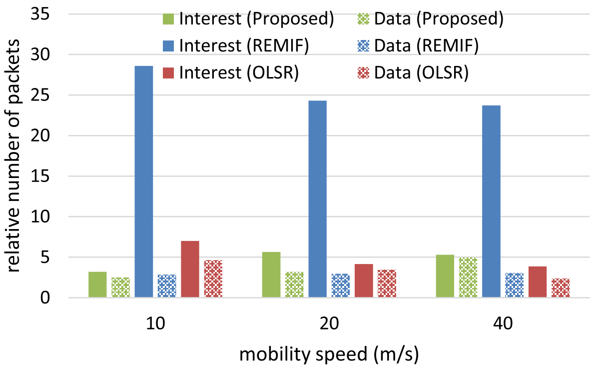


Fig. 10 Numbers of Interest and Data packets forwarded by all nodes (normalized by originated Interests; changing mobility speed).

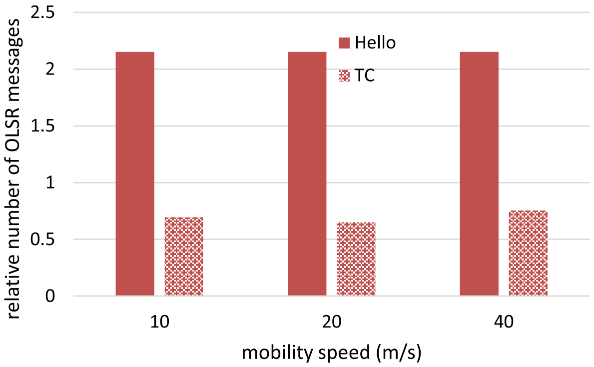


Fig. 11 Numbers of OLSR Hello and TC messages (normalized by originated Interests; changing mobility speed).

ization, the number of Data packet received by consumers shows the data delivery ratio.

In this evaluation, we selected three sets of two consumers and calculated the average of the three results. We used the same sets for REMIF and the proposed method, and other sets for OLSR based NDN, because OLSR based NDN uses 5 second route setting in the beginning of evaluation.

Figure 9 shows the total numbers of Interest and Data packets that consumers sent and received actually. The number of Interest packets is one through four times of that of the original Interest packets. The three methods have a similar tendency. Similarly, the number of Data packets that consumers received, i.e., the data delivery ratio, is 1 except the case of OLSR based NDN with 40 m/s speed, in which case the value is 0.99. With the 5 second retransition period, almost all Interest packets are satisfied by the corresponding Data packets.

Figure 10 shows the numbers of Interest and Data packets forwarded by all nodes in the network. Except the Interest packets in REMIF, the numbers are several times of the original Interest packets. The number of forwarded Interest packets in REMIF is more than twenty times of that of the original Interest packets.

Figure 11 shows the overhead of OLSR, i.e., the numbers of Hello and TC messages during the Interest origination and retransmission period. From this result, it can be said that the overhead of OLSR routing messages is not very large.

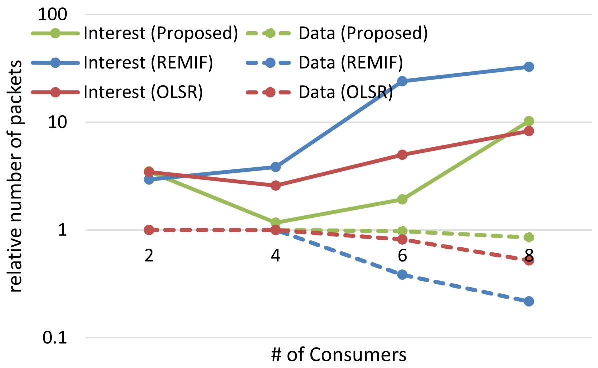


Fig. 12 Numbers of Interest packets actually sent from consumers and Data packets received by consumers (normalized by originated Interests; changing number of consumers).

Those results with two consumers show that although the number of forwarded Interest packets in REMIF is large, the data delivery rate is high for three methods, and that the mobility speed examined here does not affect the performance so much.

(3) Results of evaluation by changing number of consumers

Figures 12 and 13 show the results of the second performance evaluation. Here, we changed the number of consumers, which request their own content, from two to eight. The mobility speed is set to 20 m/s. It should be noted that the vertical axis is logarithmic in those graphs. In this evaluation, we selected one set of consumers individually for two, four, six, and eight consumer cases.

Figure 12 shows the total numbers of Interest and Data packets that consumers sent and received actually. The proposed method and OLSR based NDN have a similar tendency, but the data delivery ratio is high for the proposed method. When there are eight consumers, the ratio of the proposed method is 0.85 and that of OLSR based NDN is 0.52. On the other hand, the performance of REMIF is worse than the others. In the case of eight consumers, the number of Interest packets actually sent by consumers goes to as high as 32.7 times that of original Interest packets, and the data delivery ratio goes down to 0.27.

Figure 13, giving the total numbers of Interest and Data packets forwarded through the network, shows a similar results. In the case of eight consumers, the total number of forwarded Interest packet is 242 times of the number of original Interest packets. The proposed method and OLSR based NDN also give similar tendency in this figure.

From the results with changing the number of consumers, it can be said that the performance of REMIF is worse than the others according to the increase of consumers requesting different content. It should be noted that the REMIF used in this paper is a simplified version, which does not include the Interest suppression with deferring the

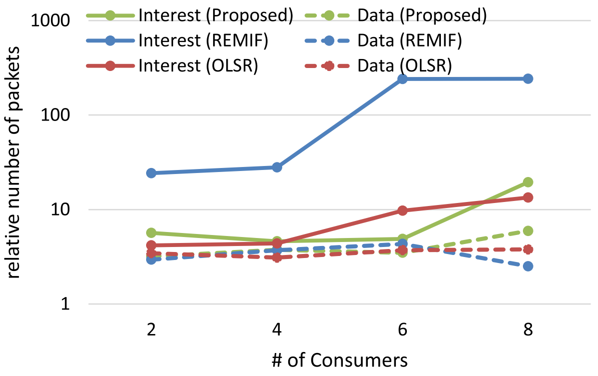


Fig. 13 Numbers of Interest and Data packets forwarded by all nodes (normalized by originated Interests; changing number of consumers).

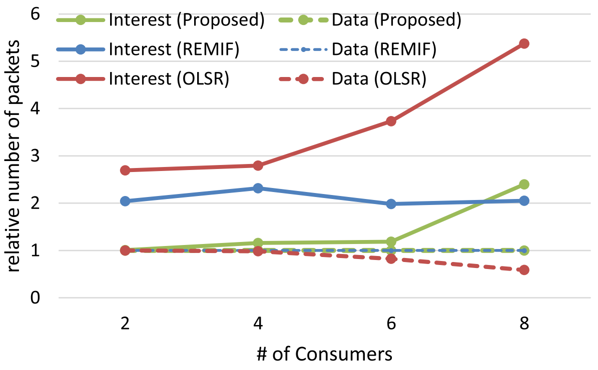


Fig. 14 Numbers of Interest packets actually sent from consumers and Data packets received by consumers (normalized by originated Interests; consumers requesting same content).

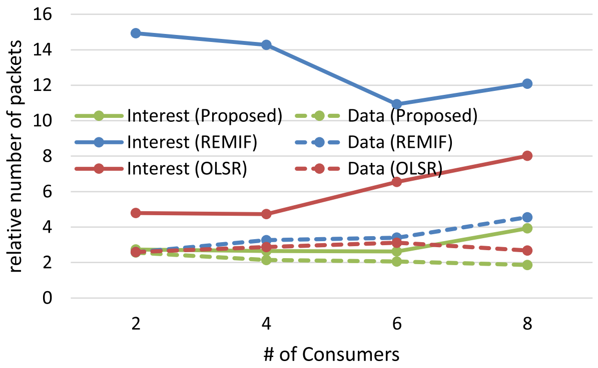


Fig. 15 Numbers of Interest and Data packets forwarded by all nodes (normalized by originated Interests; consumers requesting same content).

Interest packet flooding randomly. But, we believe that the Interest flooding without FIB may be a problem when the number of consumers is large.

(4) Results of evaluation with Data packet caching

Figures 14 and 15 show the results of the third performance evaluation. Here, all consumers request the identical content, and therefore the Data packet cache is expected to work effectively. The cache size of each node is 1,000 packets and the other conditions are the same as in the second evaluation. As described in Sect. 4, the caching does not work in OLSR based NDN, and so, it indicates the performance of IP based ad hoc network in this evaluation.

In this evaluation, we selected one set of consumers for each of two, four, six, and eight consumer cases. It should be noted that there are at most 103 content data required in this experiment, and so 1,000 packet cache size is large enough to store all of requested data in each node.

Figure 14 shows the total numbers of Interest and Data packets that consumers sent and received actually. In this figure, the results of the proposed method and REMIF changed largely compared with Fig. 12. The number of actually sent Interest packets is up to around twice of the original Interest packets. That of REMIF becomes less than 10% of Fig. 12 in the case of eight consumers. The data delivery ratio of the proposed method and REMIF is 1 through this evaluation. On the other hand, the result of OLSR based NDN is similar with that shown in Fig. 12. In the case of eight consumers, the data delivery ratio is 0.59.

Figure 15 shows the total numbers of Interest and Data packets forwarded through the network. In this figure, the result of REMIF changed largely from that in Fig. 13, although the number of forwarded Interest packets is still largest among the three method. In the case of eight consumers, the number was 242 times of that of original Interest packets, but it decreases to 12 times when the caching works well.

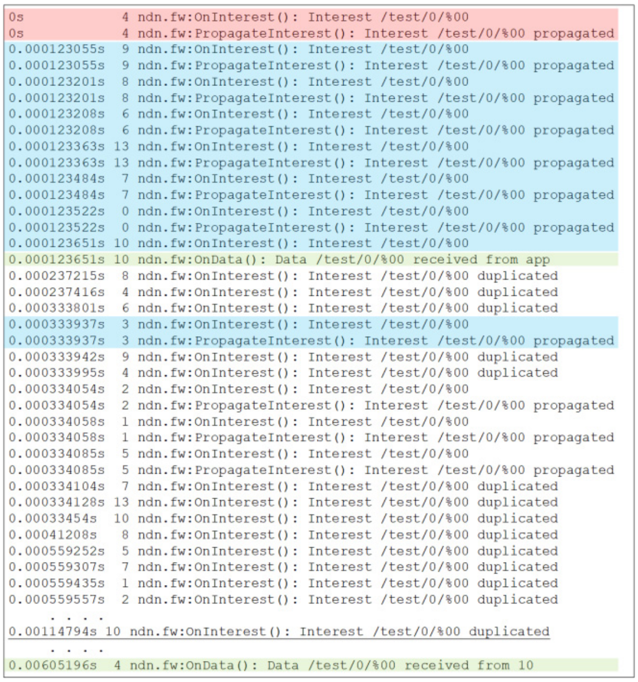
From those results, it can be said that the Data packet caching can reduce the traffic largely and that the performance can be increased compared with IP based ad hoc network.

(5) Discussions

(5-1) Comparison between simplified REMIF and original REMIF

We introduced a simplified version of REMIF in the evaluation. The features of the original REMIF are the random delay insertion in the Interest or Data packet forwarding, and the remaining energy check in the Interest forwarding. We did not implement either of them in our simplified REMIF. This means that nodes forward Interest and Data packets without any delay if necessary. As a result, a lot of unnecessary Interest and Data transmissions happened as described in Sect. 5.2 (1). So, we introduced some delay before releasing processed PIT entries (lifetime expansion of PIT entries), and succeeded in suppressing Interest forwarding for those arriving after the corresponding Data packet is processed. With this lifetime expansion and the identical Interest detection using nonce, the simplified REMIF floods only necessary Interest packets, i.e., at most one Interest per node.

Figure 16 shows an example of communication log. This is the beginning part of the log generated by the ForwardingStrategy class. Each line includes time, node id (from 0 to 13), method in this class, and other parameters including packet type and content name. The red part in the beginning indicates the Interest generation by the consumer (node 4). The blue part indicates the Interest packet for-



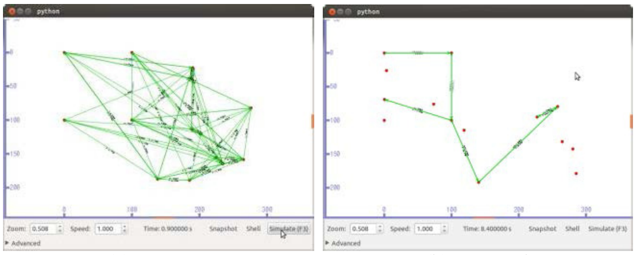
```
0s           4 ndn.fw:OnInterest(): Interest /test/0/%00
0s           4 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123055s  9 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123055s  9 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123201s  8 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123201s  8 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123208s  6 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123208s  6 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123363s 13 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123363s 13 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123484s  7 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123484s  7 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123522s  0 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123522s  0 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000123651s 10 ndn.fw:OnInterest(): Interest /test/0/%00
0.000123651s 10 ndn.fw:OnData(): Data /test/0/%00 received from app
0.000237215s  8 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000237416s  4 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000333801s  6 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000333937s  3 ndn.fw:OnInterest(): Interest /test/0/%00
0.000333937s  3 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000333942s  9 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000333995s  4 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000334054s  2 ndn.fw:OnInterest(): Interest /test/0/%00
0.000334054s  2 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000334058s  1 ndn.fw:OnInterest(): Interest /test/0/%00
0.000334058s  1 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000334085s  5 ndn.fw:OnInterest(): Interest /test/0/%00
0.000334085s  5 ndn.fw:PropagateInterest(): Interest /test/0/%00 propagated
0.000334104s  7 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000334128s 13 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.00033454s  10 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.00041208s   8 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000559252s  5 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000559307s  7 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000559435s  1 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
0.000559557s  2 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
     . . . .
0.00114794s 10 ndn.fw:OnInterest(): Interest /test/0/%00 duplicated
     . . . .
0.00605196s  4 ndn.fw:OnData(): Data /test/0/%00 received from 10
```

Fig. 16 An example of communication log of Interest and Data packet forwarding.

warding in the intermediate nodes and the producer (the producer is node 10). The green part indicates the Data packet forwarding. The rest is the discarding of Interest packets due to the duplication. Especially, the underlined part indicates the Interest discarding at node 10, where the corresponding Data packet is returned. This discard is done by the PIT entry kept during the extended lifetime. As shown in this log, unnecessary Interest forwarding is suppressed even in the simplified REMIF we used in the experiment. Therefore, we consider that the overhead of the simplified REMIF will be comparable with that of the original REMIF.

(5-2) Influence of moving speed

In the evaluation with changing the mobility speed given in Sect. 5.3 (1), the results did not vary largely. This is because the Wifi parameter setting we used in the ndnSIM simulator, which is the default setting in the ns-3 simulator, provided large Wifi coverage concerning the experiment field we supposed. Figure 17 shows examples of snapshots drawn by the ndnSIM simulator for REMIF and the proposed method. As shown in this figure, it was possible that many nodes in the experiment field communicate directly with each other. As stated in Sect. 5.2 (2), a route recovery is invoked when consumer side nodes detect consecutive PIT entry pruning timeouts. In the evaluation with two consumers by changing mobility speed, this route recovery mechanism was not invoked, although there were several route recovery invocations when the number of consumers is four, six and eight. This will be a reason for the results that the mobility speed did not give large impacts to the performance.



Zoom: 0.508 Speed: 1.000 Time: 0.900000 s

Zoom: 0.508 Speed: 1.000 Time: 8.400000 s

(a) REMIF (b) Proposal

Fig. 17 Snapshots of communication for REMIF and proposed method.

6. Conclusions

In the paper, we proposed a new hybrid routing protocol for NDN based ad hoc networks, which consist of producer side nodes with fixed positions, and mobile consumer side nodes. A proactive routing mechanism is used for the producer side nodes, and a reactive routing mechanism is introduced for the consumer side nodes. The point of our proposal is that, in some ad hoc networks we are focusing on, the producer side and the consumer side can be clearly separated, and that the suitable routing mechanisms can be selected for each side.

We showed how to implement NDN based wireless ad hoc network over the ndnSIM simulator and presented three kinds of performance evaluation with mobile nodes which move around according to the random walk model. The results of the performance evaluation show the followings.

- When the number of consumers is small, the proposed method, a simplified reactive routing (simplified REMIF), and a proactive routing (OLSR based NDN) have a similar data delivery ratio, although the number of flooded Interest packets is large in simplified REMIF. The mobility speed of consumer side nodes did not affect the delivery ratio so much.
- When the number of consumers requesting different content increases, the performance, i.e., the data delivery ratio and the routing overhead, of REMIF becomes worse. The data delivery ratio of the proposed method is better than that of OLSR based NDN supposing TOP-CCN.
- When the Data packet caching works effectively, the performance of the proposed method and REMIF is improved largely. The OLSR based NDN, which does not use the caching and therefore emulates IP based ad hoc network, has poor data delivery ratio than NDN based method. So, it can be said that the data caching is effective in the ad hoc network environment.

The results shows the effectiveness of the proposed method, but we think the performance evaluation with larger network is required. We also plan to study the application of NDN to other type of ad hoc network, such as VANET, where all nodes move and have chance to be producer.

References

[1] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking Named Content," Proc. CoNEXT '09, pp.1–12, Dec. 2009.

[2] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," NDN, Technical Report NDN-0003, pp.1–15, July 2012.

[3] A.K.M.M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data Link State Routing Protocol," Proc. ICN '13, pp.1–6, Aug. 2013.

[4] E. Hemmati and J. Garcia-Luna-Aceves, "A New Approach to Name-Based Link-State Routing for Information-Centric Networks," Proc. ICN '15, pp.29–38, Sept. 2015.

[5] X. Liu, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: A survey," Ad Hoc Network, Available online, pp.1–14, April 2016.

[6] S. Yao, X. Zhang, F. Lao, and Z. Guo, "MobileCCN: Wireless Ad-hoc Content-centric Networks over SmartPhone," Proc. ACM International Conference on Future Internet Tech. (CFI '13), pp.1–2, June 2013.

[7] J. Kim, D. Shin, and Y.-B. Ko, "TOP-CCN: Topology aware Content Centric Networking for Mobile Ad Hoc Networks," Proc. ICON '13, pp.1–6, Dec. 2013.

[8] M. Amadeo, A. Molinaro, and G. Ruggeri, "E-CHANET: Routing, forwarding and transport in Information-Centric multihop wireless networks," Computer Communications, vol.36, no.7, pp.792–803, 2013.

[9] R.A. Rehman, T.D. Hieu, H.-M. Bae, S.-H. Mah, and B.-S. Kim, "Robust and Efficient Multipath Interest Forwarding for NDN-based MANETs," Proc. WMNC '16, pp.187–192, July 2016.

[10] G. Malkin, "RIP Version 2," IETF RFC 2453, Nov. 1998.

[11] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, Oct. 2003.

[12] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF RFC 3561, July 2003.

[13] N. Minh, R. Yamamoto, S. Ohzahata, and T. Kato, "A Routing Protocol Proposal for NDN Based Ad Hoc Networks Combining Proactive and Reactive Routing Mechanism," Proc. 13th Advanced International Conference on Telecommunications (AICT 2017), pp.80–86, June 2017.

[14] N. Minh, R. Yamamoto, S. Ohzahata, and T. Kato, "Proposal and Performance Analysis of Hybrid NDN Based Ad Hoc Routing Combining Proactive and Reactive Mechanisms," International Journal on Advances in Networks and Services, vol.11, no.1&2, pp.1–10, July 2018, http://www.iariajournals.org/networks_and_services/.

[15] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, pp.1–7, Oct. 2012.

[16] "ns-3," https://www.nsnam.org/.

[17] J. Moy, "OSPF Version 2," IETF RFC 2328, April 1998.

[18] T. Teixeira, C. Wang, and M. Zink, "Can NDN Perform Better than OLSR in Wireless Ad Hoc Networks ?" Cornell University Library, pp.1–7, Sept. 2016, https://arxiv.org/abs/1609.06270.

[19] C. Park, R. Rehman, H. Tran, and B. Kim, "Enhanced Protocol for Wireless Content-Centric Network," Proc. International Conference on Mobile & Wireless Networks, pp.175–183, May 2016.

[20] R.A. Rehman and B.-S. Kim, "LOMCF: Forwarding and Caching in Named Data Networking Based MANETs," IEEE Trans. Vehicular Technology, vol.66, no.10, pp.9350–9364, Oct. 2017.

[21] E. Kalogeiton, T. Kolonko, and T. Braun, "A totology-oblivious routing protocol for NDN-VANETs," Cornell University Library, pp.1–10, Nov. 2017, https://arxiv.org/abs/1711.09870.

[22] C. Yi, et al., "A Case for Stateful Forwarding Plane," NDN, Technical Report NDN-0002, pp.1–16, 2012.

Quang Minh Ngo received the M.S. degree of Engineering from the Graduate School of Information System, The University of Electro-Communications. He is currently a PHD candidate at the Graduate School of Informatics and Engineering, the University of Electro-Communications. His research interests include an NDN technology for mobile ad hoc networks and vehicular ad hoc networks.

Ryo Yamamoto received his B.E. and M.E. degree in electronic information systems from Shibaura Institute of Technology, Tokyo, Japan, in 2007 and 2009. He received D.S. in global telecommunication studies from Waseda University, Tokyo, Japan, in 2013. He was a research associate at Graduate School of Global Information and Telecommunication Studies, Waseda University, from 2010 to 2014, and has been engaged in researching in wireless communication networks. He is presently an associate professor at Graduate School of Informatics and Engineering, The University of Electro-Communications. His current research interests are ad hoc networks, sensor networks, IoT/M2M networks, and network protocols for the networks. He is a member of IEICE and IEEE.

Satoshi Ohzahata received the B.S., M.E., and D.E. degrees from the University of Tsukuba, Ibaraki, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2007 and from 2007 to 2009, he was a Research Associate in the Department of Computer, Information and Communication Sciences and an Assistant Professor, respectively, at Tokyo University Agriculture and Technology. Since 2009, he has been an Associate Professor at the University of Electro-Communications, Tokyo, Japan. His interests are mobile ad hoc networks, the Internet architecture in mobile environments, and Internet traffic measurement. Dr. Ohzahata is a member of ACM and IPSJ.

Toshihiko Kato received the B.E., M.E., and Dr. Eng. degrees from the University of Tokyo, Tokyo, Japan, in 1978, 1980, and 1983, respectively, all in electrical engineering. In 1983, he joined KDD and worked in the field of OSI and Internet until 2002. From 1987 to 1988, he was a Visiting Scientist at Carnegie Mellon University. He is currently a Professor of the Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo. His current research interests include protocols for mobile Internet, high-speed Internet and ad hoc network.