Performance Evaluation of MANET Based on Named Data Networking Using Hybrid Routing Mechanism

Ngo Quang Minh, Ryo Yamamoto, Satoshi Ohzahata, and Toshihiko Kato
University of Electro-Communications, Tokyo 182-8585, Japan
Email: {mingus, ryo_yamamoto, ohzahata, kato}@net.lab.uec.ac.jp

Abstract—We have been studying on the application of the named data networking technology to mobile ad hoc networks. We suppose a type of ad hoc networks that advertise versatile information in public spaces such as shopping mall and museum. The proposed approach is a hybrid one 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. In this paper, we show the results of performance evaluation focusing on the communication overhead. The results indicate 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.

Index Terms—Ad hoc network, named data networking, proactive routing, reactive routing

I. Introduction

Your goal is to simulate the usual appearance of papers in a Journal of the Engineering and Technology Publishing. We are requesting that you follow these guidelines as closely as possible.

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 statuses 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]. In these reactive routing mechanisms, FIB is not used at all, but Interest packets are flooded to find producers or cached Data packets. Only Pending Interest Table (PIT) is used for forwarding Data packets.

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. The NDN technologies 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 has the following features. 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

Manuscript received March 25, 2019; revised August 7, 2019.
Corresponding author email: kato@net.lab.uec.ac.jp.
doi:10.12720/jcm.14.9.819-825

producer side network, because of its in-advance route setting, and a reactive routing is adopted in a consumer side network, because 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.

The third is about the procedure of reactive routing. The reactive routing of our proposal uses both FIB and PIT. Although the first Interest packet for a specific name prefix is flooded, the corresponding FIB entry is created by the returning Data packet and the following Interest packets for the name prefix are transferred by this FIB entry. The similar procedure is proposed in [15].

Although the basic idea was presented in our previous papers, the performance evaluation results described in those papers are limited, because they only provide rather simple theoretical analysis and do not evaluate detailed behaviors of routing protocols.

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 [16], a widely used NDN simulator implemented over the ns-3 network simulator [17].

The rest of this paper consists of the following sections. Section 2 describes our NDN ad hoc routing protocol. Section 3 shows the implementation of the proposed protocol over ndnSIM and the results of the performance evaluation. Section 4 concludes this paper.

II. Hybrid Routing Protocol for NDN MANET

A. Design Principle

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 [18].
- 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 consumer 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.

B. Detailed Design for Producer Side

Table I 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:

TABLE I: 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.

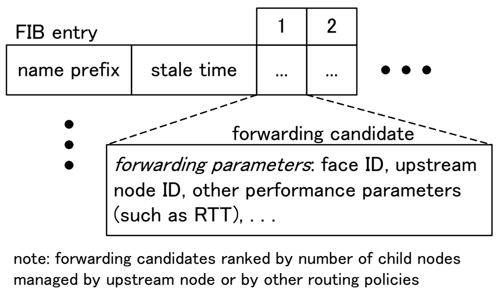

Fig. 1. Structure of FIB at producer side

Fig. 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.

	upstream record					
name prefix	producer node ID, nonce	hop count from producer	receiving time	face ID, upstream node ID	# of siblings	# of children
• • •	producer node ID, nonce	hop count from producer	receiving time	face ID, upstream node ID	# of siblings	# of children
	• • •					

Fig. 2. Structure of DAG table at produce node.

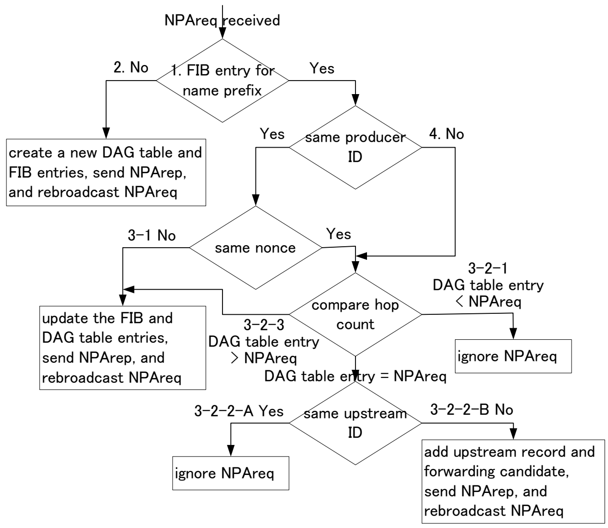

Fig. 3. Flow chart for a received NPAreq packet.

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 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.

C. Detailed Design for Producer 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.

III. Implementation and Performance Evaluation

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

A. *Evaluated Methods and Their Implementation Details*

The methods evaluated in this section are the proposed method, REMIF [9] (simplified version), an example of reactive NDN routing system, 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 [7], an example of proactive NDN routing system.

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

1) REMIF

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

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 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. 4. Node 1 (a consumer) broadcasts an Interest packet (1) and node 2 rebroadcasts it (2). Then, node 4 (a producer) sends a Data packet to node 2, which transfers it to node 1 (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 (5). Node 4 sends the Data packet again to node 3, which transfers it to node 1 (6). In both cases, the received Data packet is ignored because there are no corresponding PIT entry.

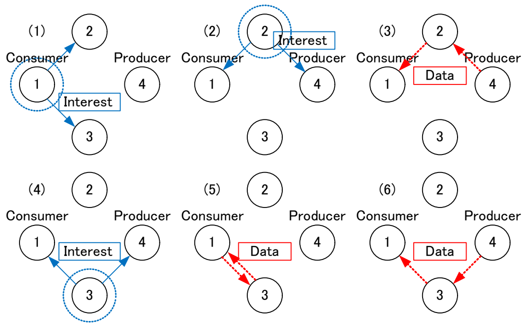

Fig. 4. Mis-ordering in REMIF.

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 by running the ForwardingStrategy class over UDP/IP stack. This can be done by UdpFace in ndnSIM. We 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.

B. *Simulation Conditions*

Fig. 5 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 ixed through a 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.

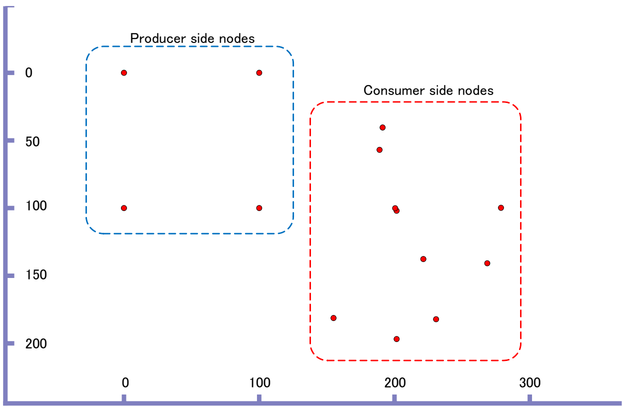

Fig. 5. Network configuration for simulation.

The details of simulation condition are given in Table II. 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 10 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 nodes work as consumers requesting different content. So, the data packet caching is not effective in the simulation.

TABLE II: 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, 10 m/s
Consumer	Randomly selected two nodes; request different content; 10 Interests/s
Producer	node at (0, 0); Data packet size = 1200 bytes
Evaluation	10 second for each simulation run; three runs for each mobility speed with different consumers

C. *Evaluation Results*

As for REMIF and the proposed method, we evaluated the following features, by changing the mobility speed of consumer side nodes:

- The number of Interest packets originated from two consumers,
- The number of Interest packets actually sent from two consumers (including retransmissions),
- The number of Data packets two consumers received,
- The total number of forwarded Interest packets, and
- The total number of forwarded Data packets.

Tables III and IV give the results for REMIF and the proposed method. As described above, the results are the average among three selections of consumer nodes.

TABLE III: RESULTS OF REMIF

	Originated Interests	Total Interests from Consumer	Data received by Consumer	Forwarded Interests	Forwarded Data
40 m/s	200	568	184	4992	867
20 m/s	200	604	164	5341	848
10 m/s	200	640	147	5658	787

TABLE IV: RESULTS OF PROPOSED METHOD

	Originated Interests	Total Interests from Consumer	Data received by Consumer	Forwarded Interests	Forwarded Data
40 m/s	200	294	160	783	420
20 m/s	200	358	145	858	396
10 m/s	200	267	197	911	530

TABLE V: RESULTS OF OLSR BASED NDN

	Originated Interests	Total Interests from Consumer	Data received by Consumer
40 m/s	200	334	187
20 m/s	200	292	153
10 m/s	200	376	151

The results show that the number of forwarded Interest and Data packets is smaller for the proposed method. This means that the overhead of sending Interest packets is small in the proposed method compared with REMIF.

Table V gives the result for OLSR based NDN, which shows the numbers of originated Interest packets, actually transmitted Interests packets, and received Data packets.

The result shows that the performance of OLSR based NDN is similar with the proposed method. However, in this evaluation, we used the hello interval of 0.5 sec and the TC (topology control) interval of 1 sec. So, there are some overhead by these OLSR messages actually. We need to evaluate these overheads in the future.

IV. Conclusions

In the paper, we showed the implementation and performance evaluation of a new hybrid routing protocol for NDN based ad hoc networks. The proposed method 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. We showed how to implement NDN based wireless ad hoc network over the ndnSIM simulator and presented the performance evaluation with mobile nodes which move around according to the random walk model. The results show that the overhead of Interest packet dissemination is relatively small for the proposed method.

References

[1] V. Jacobson, *et al.*, "Networking named content," in *Proc. of 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, Jul. 2012.

[3] A. Hoque, *et al.*, "NLSR: Named-data link state routing protocol," in *Proc. of ICN '13*, Aug. 2013, pp. 1-6.

[4] E. Hemmati and J. Garcia-Luna-Aceves, "A new approach to name-based link-state routing for information-centric networks," in *Proc. of ICN '15*, Sep. 2015, pp. 29-38.

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

[6] S. Yao, X. Zhang, F. Lao, and Z. Guo, "MobileCCN: Wireless ad-hoc content-centric networks over smartphone," in *Proc. of ACM International Conference on Future Internet Tech. (CFI '13)*, Jun. 2013, pp. 1-2.

[7] J. Kim, D. Shin, and Y. Ko, "TOP-CCN: Topology aware content centric networking for mobile ad hoc networks," in *Proc. of ICON '13*, Dec. 2013, pp. 1-6.

[8] M. Amadeo, A. Molinaro, and G. Ruggeri, "E-CHANET: Routing, forwarding and transport in Information-Centric multihop wireless networks," *Computer Communications*, vol. 36, pp. 792-803, 2013.

[9] R. Rehman, T. Hieu, and H. Bae, "Robust and Efficient multipath interest forwarding for NDN-based MANETs," in *Proc. of WMNC '16*, Jul. 2016, pp. 1-6.

[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, Jul. 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," in *Proc. 13th Advanced International Conference on Telecommunications (AICT 2017)*, Jun. 2017, pp. 80-86.

[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, Jul. 2018.

[15] E. Kalogeiton, T. Kolonko, and T. Braun, "A topology-oblivious routing protocol for NDN-VANETs," Cornell University Library, pp. 1-10, Nov. 2017.

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

[17] ns-3. [Online]. Available: https://www.nsnam.org/

[18] C. Yi, *et al.*, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, pp. 779-791, 2013.

Ngo Quang Minh 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 assistant 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.

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.

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 communication protocols 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.