Regular Paper

Improvement in IntErzone Routing Protocol of ZRP Based on Bloom Filter

Yuria Oigawa[1] Fumiaki Sato[1,a)]

Received: May 10, 2017, Accepted: November 7, 2017

Abstract: A mobile ad hoc Network (MANET) is a self-organizing and self-configurable wireless network of mobile nodes, which move arbitrarily without a pre-existing infrastructure. Routing is a critical issue in MANETs. One well-known protocol for ad hoc networks is the Zone Routing Protocol (ZRP). However, because ZRP sends many useless control packets, the network load increases and the network performance decreases. This paper examines a novel routing protocol that uses Bloom filters and the topology information of neighbor nodes to improve ZRP. A Bloom filter is often used as a content discovery method. In this case, the content ID is hashed to one value. In our proposed protocol, the node ID is hashed to one value, and a Bloom filter is generated from the ID values in a zone. The generated Bloom filter is propagated across a tree structure over a MANET. The tree is constructed by the peripheral nodes of the IntErzone Routing Protocol (IERP) of ZRP. The Bloom Filter sent from a peripheral node is managed as node information of the direction of the peripheral node. The number of RouteQuery packets of IERP can be reduced by using the managed Bloom filter. We compare the performance of the proposed routing protocol with the bordercast of IERP and show the feasibility of the proposed method as a routing protocol for MANETs.

Keywords: Zone Routing Protocol, IERP, Bloom filter

1. Introduction

A mobile ad hoc network (MANET) is a collection of wireless nodes that dynamically form a network and exchange information without a pre-existing fixed network infrastructure [1]. Much work has been done on the routing in MANETs. Many protocols and algorithms have been proposed, such as Destination-Sequenced Distance-Vector (DSDV) protocol [2], Dynamic Source Routing (DSR) protocol [3], Ad hoc On-demand Distance-Vector (AODV) protocol [4], Zone Routing Protocol (ZRP) [5]. Among these protocols, ZRP is a widely used application. However, when ZRP searches for a new route, it sends many useless control packets, which increase the network load and decrease the network performance.

Existing ad hoc routing protocols can be classified into two groups: proactive and reactive routing protocols. Proactive routing protocols attempt to continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. Reactive protocols invoke a route determination procedure on an on-demand basis. The preferred protocol is one that combines the advantages of both the proactive and the reactive protocols. ZRP [5] is a hybrid reactive/proactive scheme. In ZRP, a node proactively maintains routes destinations within a local neighborhood, which is considered as a routing zone. A node routing zone is defined as a collection of nodes whose minimum distance hop from the node is no greater than a parameter referred to as the zone radius. Each node maintains its own routing zone, but the routing zones of neighborhood nodes overlap. For routing, a node can proactively communicate with the IntrAzone Routing Protocol (IARP) within its zone and with the IntErzone Routing Protocol (IERP) beyond the routing zone. ZRP uses the concept of bordercasting. For example, if a node cannot find its destination within a zone, it sends the packet to the peripheral nodes of its routing zone and then the peripheral is responsible for searching the destination to its own zone proactively. This process continues until the source node finds its destination. The reactive routing process is divided into two phases: the route request phase and the route reply phase. Although ZRP performs better than any single proactive or reactive protocol, its performance can be further enhanced. The drawback of ZRP is that the RouteQuery packet is flooded throughout the network, so that there is a large overhead to unrelated nodes.

Therefore, the Enhanced Zone-Based Routing Protocol (EZRP) [6] was proposed to overcome the drawbacks of ZRP. In EZRP, each node calculates the reliability of the route. The reliability is based on the number of successful data transmissions. The reliable route is used without a route searching cost. For calculating the reliability, each node must save its searching records ten times for each route [6], [7].

A previous study of MANET routing [8] reported the reduction of the number of control packets by applying the Bloom filter [9], which is one of the search algorithms. The Bloom filter algorithm searches by using a simple bit sequence, and an application for network protocols was recently proposed. The routing scheme of the previous study [8] limits the flooding scope of data pack-

[1] Department of Information Science, Toho University, Funabashi, Chiba 274–8510, Japan
[a)] fsato@is.sci.toho-u.ac.jp

© 2018 Information Processing Society of Japan

ets by using a Bloom filter to search for a direction where the destination node may be present. Therefore, the number of control packets can be reduced. However, the cost of management of Bloom filters is large because each node must manage the Bloom filters.

In this paper, we propose an enhanced IERP of ZRP by using a Bloom filter to reduce the number of IERP control packets. Because only the peripheral node of IERP manages the Bloom filters, the cost of management is reduced. The Bloom filter of the proposed scheme is made from nodes in the zone of ZRP as a basic unit. The generated Bloom filter is propagated across a tree structure over a MANET. The tree is constructed of the peripheral nodes of IERP. A RouteQuery packet of IERP is forwarded in a direction where the destination node may be present, as determined by a Bloom filter.

This paper is organized as follows. Sections 2 and 3 summarize issues on existing routing protocols and related work, respectively. Section 4 describes our proposal protocol. Section 5 evaluates the proposed protocol in a simulation. Finally, Section 6 presents the conclusions of this paper.

2. MANET and ZRP

2.1 MANET Routing Protocols

A number of routing protocols have been proposed for ad hoc networks [1], [2], [3], [4]. As mentioned in Section 1, these protocols can be classified into two main categories: proactive (table-driven) and reactive (source-initiated or demand-driven). Proactive routing protocols attempt to keep an up-to-date topological map of the entire network. With this map, the route is known and immediately available when a packet needs to be sent. The approach is similar to the one used in wired IP networks.

In contrast to proactive routing, reactive routing does not attempt to continuously determine the network connectivity. Instead, a route determination procedure is invoked on demand when a packet needs to be forwarded. The technique relies on queries that are flooded throughout the network.

Both proactive routing and reactive routing have specific advantages and disadvantages that make them suitable for certain types of scenarios. Because proactive routing maintains information that is immediately available, the delay before sending a packet is minimal. In contrast, reactive protocols must first determine the route, and so the result may be a considerable delay if the information is not available in caches.

Proactive routing uses excess bandwidth to maintain routing information, whereas reactive routing involves long route request delays. Reactive routing also inefficiently floods the entire network for route determination. ZRP [5] aims to address these problems by combining the best properties of both approaches. ZRP can be classed as a hybrid reactive/proactive routing protocol.

2.2 Zone Routing Protocol (ZRP)

ZRP, as its name of Zone Routing Protocol implies, is based on the concept of zones. A routing zone is defined for each node separately, and the zones of neighboring nodes overlap. The routing zone has a radius ρ expressed in hops. The zone thus includes the nodes, whose distance from the node of interest is at most ρ hops.

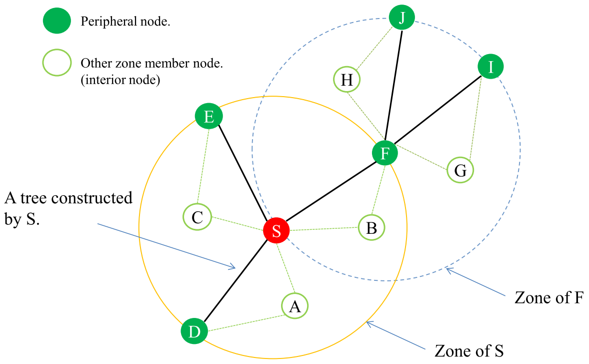


Fig. 1 Example of a zone of ZRP.

An example routing zone is shown in **Fig. 1**, where the routing zone of S includes the nodes A–F, but not G–J. The zone radius of Fig. 1 is 2. In the illustration, the radius is marked as a circle around the node of interest. It should be emphasized, however, that the zone is defined in hops, not as physical distances.

The nodes of a zone are divided into peripheral nodes and interior nodes. Peripheral nodes are nodes whose minimum distance to the central node is exactly equal to the zone radius ρ. The nodes whose minimum distance is less than ρ are interior nodes. In Fig. 1, nodes A–C are interior nodes, nodes D–F are peripheral nodes, and nodes G–J are outside the routing zone.

The number of nodes in the routing zone can be regulated by adjusting the transmission power of the nodes. Lowering the power reduces the number of nodes within direct reach, and vice versa. The number of neighboring nodes should be sufficient to provide adequate reachability and redundancy. On the other hand, too large a coverage results in many zone members and the update traffic becomes excessive. Furthermore, large transmission coverage increases the probability of local contention.

As mentioned in the Introduction, the locally proactive routing component of ZRP is IARP [10], whereas the globally reactive routing component is IERP [11]. IARP and IERP are not specific routing protocols. Instead, IARP is a family of limited-depth, proactive link-state routing protocols. IARP maintains routing information for nodes that are within the routing zone of the node. Correspondingly, IERP is a family of reactive routing protocols that offer enhanced route discovery and route maintenance services based on the local connectivity monitored by IARP.

The fact that the topology of the local zone of each node is known can be used to reduce traffic when global route discovery is needed. Instead of broadcasting packets, ZRP uses a concept called bordercasting. Bordercasting utilizes the topology information provided by IARP to direct query requests to the border of the zone. The bordercast packet delivery service is provided by the Bordercast Resolution Protocol (BRP). BRP uses a map of an extended routing zone to construct bordercast trees for the query packets. Alternatively, it uses source routing based on the normal routing zone. By employing query control mechanisms, route requests can be directed away from areas of the network that have already been covered [12].

Though the bordercast protocol can reduce the number of control packets of the reactive protocol of ZRP, IERP request packets are sent to all border nodes. These packets may cause network congestion that decreases network performance.

3. Related Work

3.1 EZRP

To overcome the drawbacks of ZRP, the EZRP was proposed [6], [7]. In EZRP, each node calculates the reliability of the route. In the case of a reliable route, the source node sends a data packet directly to the destination node on that route without route searching. In the case of an unreliable route, the source node searches again for a new route. To adapt EZRP, a Reliable Degree (RD) parameter was proposed to calculate the route quality in the proposed EZRP. When the RD of the route is high, a data packet is sent directly without route searching. When the RD is medium, route searching is performed in some limited directions. In this case, we assume that the node's movement distance from its previous position is small. When the RD is low, route searching is done in all directions, as in ZRP. For calculating the RD, each node must save the searching records ten times for each route.

3.2 Bloom Filter

The Bloom filter is a data structure that uses hash functions and bit strings of constant length to express where certain content exists on a site [9].

First, a bit string of n bits is prepared to compose the Bloom filter, and all bits are initialized to "0." Next, k hash values of the content are calculated by k hash functions and the content. The bit at the position which corresponds to each hash value is changed to "1." To judge the presence of content by using the Bloom filter, the bit of the Bloom filter at the position corresponding to the hash value of the content of interest is examined. It can be judged that the content exists if all the corresponding bits are "1." However, there is a possibility of judging that the content exists because of a hash value collision even when the content does not exist. However the opposite, that is, the content being judged not to exist when it does, cannot occur.

The Bloom filter has the feature of obtaining a Bloom filter which expresses all contents included in each Bloom filter by applying the OR operation to two or more Bloom filters. In such cases, the bit length of the Bloom filter does not change regardless of the number of contents. In this research, the key word of a content is expressed by using the Bloom filter. The key word is expressed in the same bit length regardless of the number of key words, and the expression of two or more key words of the contents becomes possible.

Figure 2 is an example of expressing the key word of contents with the Bloom filter. From the key word "*abc*," 4 and 9 are calculated by the hash function "hash1" and "hash2" respectively. Therefore the bits in 4th and 9th place of the Bloom filter are set to 1. Similarly, from the key word "*xyz*," 0 and 4 are calculated by the hash function "hash1" and "hash2" respectively. Therefore the bits in 0th and 4th place of the Bloom filter are set to 1. The Bloom filter integrates two keywords in a filter. If you want to search the keyword "*abc*" from this integrated filter, you prepare the search filter which includes only one keyword you wish to search. Then you apply AND operation between the integrated Bloom filter and the search filter. If the result is the same as the search filter, the keyword may be included in the integrated Bloom filter. However, sometimes the keyword is not included because of the false positive.

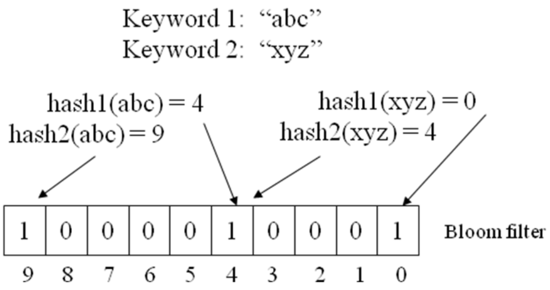


Fig. 2 Example of the Bloom filter.

3.3 MANET Routing Based on Bloom Filters

To reduce the number of routing overhead packets, the routing protocol based on the Bloom filter was proposed [8]. In the routing protocol, a node in a network periodically sends a keep-alive message to its neighbor nodes for the purpose of link management. By doing so, the node learns the node ID of its neighbor. Each node in the network calculates its own Bloom filter from its node ID. The node also has Bloom filters for all links to its neighbor nodes. These Bloom filters are updated by control messages between neighbor nodes.

A node updates its Bloom filter by using an update (0) message. The node sends an update (0) message to the neighbor nodes. The update (0) message is used to request a neighbor node to send its Bloom filter in order to update the Bloom filter of the requesting node. For example, if node A is asked to send its Bloom filter from node B, node A calculates the Bloom filter by the OR operation of A's node ID and the Bloom filters of other links except the one for node B. Then, node A returns the filter to node B. Therefore, node B holds a Bloom filter for the link corresponding to node A.

Next, node B sends an update (1) message to its neighbor nodes. The update (1) message of node B includes the Bloom filter made by the OR operation of B's ID and the Bloom filters for the links to other nodes except node A. The message also includes the node information to which node A should forward the message based on the information within two hops from node B. After receiving the update (1) message, node A holds the received Bloom filter as the one for the link corresponding to node B. Node A forwards the update (1) message to node C that is specified in the update (1) message as the forwarding node ID. Then, node C rewrites the information in the update (1) message based on the Bloom filter that node A has and the information of the node within two hops from node A. After receiving the update (1) message, node C holds the received Bloom filter as the one for the link corresponding to node A.

To transfer the data packet, each node decides the node to which it transfers the data in accordance with the following steps. The sender node calculates the Bloom filter for the receiver node of the packet. Then, the sender node compares the Bloom filter for each link with its neighbor nodes. If the Bloom filter hits the

© 2018 Information Processing Society of Japan

link of the neighbor node, the data packet is transferred to that link.

However, each node must exchange the update (0) message and update (1) message periodically. Though smaller than that of the flooding procedure of the DSR protocol, the cost is not small.

4. Enhancement of IERP Based on Bloom Filters

IERP of ZRP has the problem of increased network load because IERP floods control packets to its surroundings. In this research, by using the Bloom filter to control the propagation direction of IERP and BRP, we propose a method to reduce the number of IERP control packets.

4.1 Proposed Method

(1) Tree structure of peripheral nodes

In this method, a tree structure of nodes for transferring query packets is created. This process is started by a designated node selected in advance. The member nodes in this tree are only peripheral nodes. Each member node of the tree knows a parent node and child nodes. To construct the tree, the enhanced procedure of IERP is used.

In **Fig. 3**, three types of control messages and their directions are depicted. In this method, the following three steps are used to construct the tree structure.

[Step 1]

A root node of a tree sends a TreeQuery packet to the peripheral nodes of the zone in order to construct the tree structure. The node which receives the TreeQuery forwards it to its own peripheral nodes. In this case, the node does not forward the same TreeQuery packet twice. The first node that receives the TreeQuery is set as the parent node.

[Step 2]

The node which forwards TreeQuery waits for a TreeReply message for time interval *Tr*. After receiving some TreeReply messages, the node integrates the Bloom filters in the TreeReply messages and the filter of its own zone and sets it in the TreeReply and sends it to the parent node. The TreeReply message must be replied to the root node from the end node of the tree in order. Therefore, the TreeReply reception time of each peripheral node is set to a value corresponding to the depth of the tree.

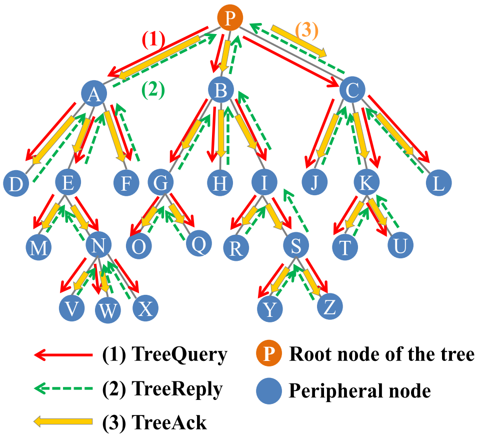


Fig. 3 Tree structure and control messages.

$$T_r(k) = (D_{\max} - D(k)) \cdot \Delta t \qquad (1)$$

In Eq. (1), $D(k)$ is the depth of the tree of node k. $Dmax$ is the maximum depth of the tree, and Δt is the time interval to wait for the TreeReply message for each depth.

The node which receives the TreeReply manages the sender of TreeReply as a child node. The Bloom filter of the TreeReply message is taken out and managed as node information of the direction of the link.

[Step 3]

If the root node receives the TreeReply message from its peripheral node, that peripheral node is set as the child node and the Bloom filter is taken out from the message and managed as node information of the direction of the child node. After that, the root node makes the Bloom filter for each child node. The root node integrates the Bloom filters of the child nodes and its own Bloom filter except for the destination child node. The integrated Bloom filter is set in the TreeAck and sent to the destination child node. Each node which receives the TreeAck message takes out the Bloom filter and sets it as node information of the direction of parent node. Then, the node makes the Bloom filter for each child node. The Bloom filter is the integration of the Bloom filter of the parent node and its own zone Bloom filter and child node except for the destination child node. The Bloom filter is set in the TreeAck and sent it to the child node.

When a change occurs, the tree structure is recreated at regular intervals in consideration of the mobility of the nodes, or is locally repaired. The maintenance process of the tree is discussed in Section 4.2.

[Root node selection]

Depending on the tree structure and the position of the sender and receiver nodes, it is known that a detour route can be formed, which leads to an increase of delay. Therefore, how to select the root node and how to build the tree is very important.

The method of randomly choosing the root node is a typical selection method with a low cost of calculating the root node. For example, each node generates a random number and elapses the time for a random interval and if no one transmits the TreeQuery message after the elapsed time, the node becomes root and transmits the TreeQuery. If the collision occurs, the node who has the larger IP address wins. Alternatively, there is a way that the first node that is trying to create the route at the time the tree is not yet created becomes the root. However, it is not always possible to compose efficient trees.

Since it is impossible to create an optimal tree for all nodes, creating an average efficient tree is one way. By choosing the node at the center of the network as the root node, we can construct an average short route as a whole. To do that, there is a method of shifting the position of the root node in the direction of the farthest partial tree from the tree once constructed. Shifting the root node in the direction of the furthest subtree makes it possible to average the depth of the subtree and construct an average short route. Also, if there is an imbalance in the transmission frequency of nodes, selecting a node with a high transmission/reception frequency as the root node is an efficient tree as a whole.

© 2018 Information Processing Society of Japan

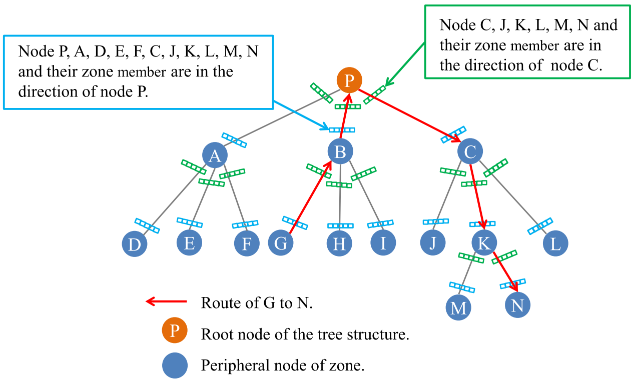

Node P, A, D, E, F, C, J, K, L, M, N and their zone member are in the direction of node P.
Node C, J, K, L, M, N and their zone member are in the direction of node C.
P
A
B
C
D
E
F
G
H
I
J
K
L
M
N
Route of G to N.
Root node of the tree structure.
Peripheral node of zone.

Fig. 4 Tree structure and control messages.

In this paper, our simulation mainly focused on the evaluation of the number of control packets and the effect of the Bloom filter size. Therefore, we selected a central node in advance as a root node, created a tree from that node, and conducted a simulation of route search using that tree.

(2) Route search

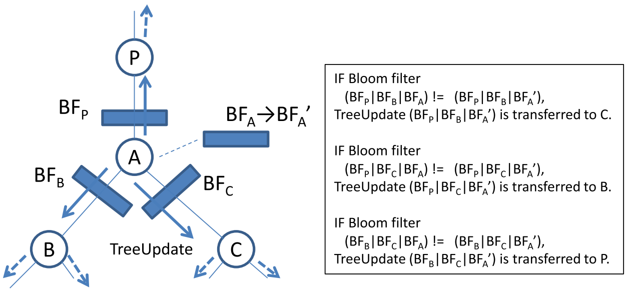
If a transmission request to the outside of the zone occurs, the node creates a RouteQuery packet. If the node is a member node of the tree, the node manages the Bloom filters for the parent node or the child node. Therefore, the destination node ID is compared with the Bloom filter of each direction, and then the RouteQuery packet is transferred in the hit direction. If the node is not the member node of the tree, the packet is transferred to the neighbor tree node.

The node which receives the RouteQuery replies with a Route Reply packet to the source node if the destination node ID is in its own zone. Otherwise, the node forwards the RouteQuery in the direction where the Bloom filter is hit by the destination node ID.

If the source node of the RouteQuery receives the RouteReply, the node sends the data packet by using the route information written in the RouteReply packet.

Figure 4 shows an example of forwarding of the RouteQuery packet across the tree structure. In this example, node G sends the RouteQuery to find destination node N. Because node G has only one link, to node B, node G sends the RouteQuery to node B. Though node B has 4 links of the tree, it sends the packet to only node P because node N is in the direction of node P. It is tested whether a destination node ID is included in the direction by Bloom filter. Node P uses the Bloom filter to determine whether the destination node ID is included in the direction. Node P also transfers the RouteQuery packet only in the direction of node C because node N is in the direction of node C. This procedure is repeated and the RouteQuery packet is then transferred to the destination node.

Because a Bloom filter can have a false positive, RouteQuery can also be transferred in the incorrect direction. The message to the wrong direction becomes the overhead. Furthermore, a tree structure is adopted to avoid the overlap of the node information of the Bloom filter. However, because the path of the tree is limited, it is possible that the route is not optimal.



IF Bloom filter
$(BF_P|BF_B|BF_A)$!= $(BF_P|BF_B|BF_A')$,
TreeUpdate $(BF_P|BF_B|BF_A')$ is transferred to C.

IF Bloom filter
$(BF_P|BF_C|BF_A)$!= $(BF_P|BF_C|BF_A')$,
TreeUpdate $(BF_P|BF_C|BF_A')$ is transferred to B.

IF Bloom filter
$(BF_B|BF_C|BF_A)$!= $(BF_B|BF_C|BF_A')$,
TreeUpdate $(BF_B|BF_C|BF_A')$ is transferred to P.

Fig. 5 TreeUpdate transfer.

4.2 Maintenance of Tree Structure and Bloom Filter

The routing table in a Zone is maintained by IARP. Therefore, node information in a Zone is collected by IARP. Though the calculation cost of the Bloom filter of nodes in a Zone is related to the number of nodes, the calculation cost is small. The cost to create one Bloom filter is to calculate a hash function five times in our simulation condition. The integrated Bloom filter is calculated by only "OR" operation of all Bloom filters in the zone. Searching for bloom filter is only "AND" operation between integrated Bloom filter and search filter, regardless of the number of Bloom filters integrated.

Each node in the tree manages the integrated Bloom filter of its own Zone, and Bloom filters for all directions of the tree (parent, and all children). Therefore, in our simulation environment, the node in the tree had to manage up to five Bloom filters.

In order to construct a tree structure, three types of control messages are exchanged. A TreeQuery message is a flooding-type message same as the RouteQuery of IERP. TreeReply and TreeAck is the unicast-type message sent along the tree structure. TreeQuery message does not include the Bloom filter. A TreeReply message includes one Bloom filter. A TreeAck message contains as many Bloom filters as the number of child nodes. The message making the tree structure is larger than the overhead of the route search of IERP (RouteQuery + RouteReply), however by using the tree once made for a long time the cost becomes relatively small.

Next, the maintenance cost of the tree must be considered. In order to manage the change of the Bloom filter of the Zone by the mobility (join, leave) of the nodes which are not the member of the tree, a TreeUpdate message can be used. For example, node P, A, B and C are the members of the tree depicted as **Fig. 5**. Node A has its own zone Bloom filter BFA and the three Bloom filters for all directions, BFP, BFB and BFC. If the BFA is changed to the BFA', if the change must be transferred to the direction, TreeUpdate will be sent. For example, if $(BF_P\,|\,BF_B\,|\,BF_A)$!= $(BF_P\,|\,BF_B\,|\,BF_A')$, the $(BF_P\,|\,BF_B\,|\,BF_A')$ will be transmitted to node C ("|" means OR operation). If node C decides that the Bloom filter for that direction must be transferred further, the node C will transfer the TreeUpdate message. The number of transfers is at most the number of member nodes of a tree −1.

However, if the node in the tree moves or leaves from the tree, the tree structure cannot be maintained and reconstruction is required. Furthermore, movement of a member node of a tree may change to a state in which it is possible to construct a more appro-

© 2018 Information Processing Society of Japan

priate route by reconstruction of the tree. Therefore, it is necessary to rebuild trees at regular intervals. The overhead is related to the frequency of the reconstruction.

5. Simulation

5.1 Simulation Overview

To evaluate the effectiveness of the proposed method, we used the network simulator QualNet6.1 to compare the performances of the proposed method and ZRP. The metrics to be compared are the number of query packets for the route search and the length of the route.

The parameters of this simulation are summarized in **Table 1**. For simplification, 81 nodes are arranged in a grid, and the nodes do not move. Three types of Bloom filter lengths, 32 bits, 64 bits, and 128 bits, are used. The number of hash functions is a maximum of five. The source and destination pairs are 32 patterns: 4 patterns from the center to the corner, 8 patterns for the four sides, 4 patterns for the diagonal, and 16 patterns from the center of the side to two corners on the opposite side. The communication patterns are shown in **Fig. 6**. The Zone radius is two. The number of simulations is 4 times in each pattern. In order to investigate the influence of the false positive of the Bloom filter, we select a center node of the network as a root node in advance. The number of nodes in the tree is 21 in this case. In this simulation, the node was fixed so as to eliminate the influence of the node mobility and focus on the effect of Bloom filter.

Table 1 Simulation parameters.

Parameters	Values
Simulator	Qualnet 6.1
Node layout	Grid
Distance between nodes	200 m
Number of nodes	81
Reachable distance of radio waves	210 m
Mobility of nodes	no
MAC Protocol	IEEE802.11b
Bloom filter length	32,64,128bit
Number of hash functions	1, 3, 5

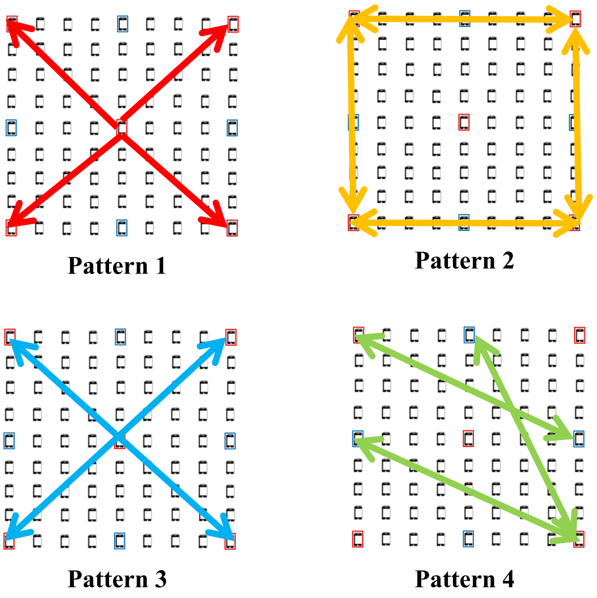


Fig. 6 Communication patterns used in the simulation.

5.2 Number of Query Packets

Figure 7 shows the number of query packets needed to find a route by using the proposed method and ZRP. The Bloom filter length is changed from 128 bits, 64 bits, and 32 bits. The number of hash functions is 3. The proposed scheme with any Bloom filter size is able to reduce the number of query packets from the conventional ZRP method in almost all experiments. In the case of the proposed method with the 128-bit Bloom filter, it is possible to reduce the number of query packets to about 47, which is 74% of the packets needed for the conventional ZRP.

Figures 8 and **9** show the number of query packets needed to find a route by using the proposed method with a different number of hash functions. The Bloom filter length is 32 bits in Fig. 8 and 128 bits in Fig. 9. The number of hash functions is changed

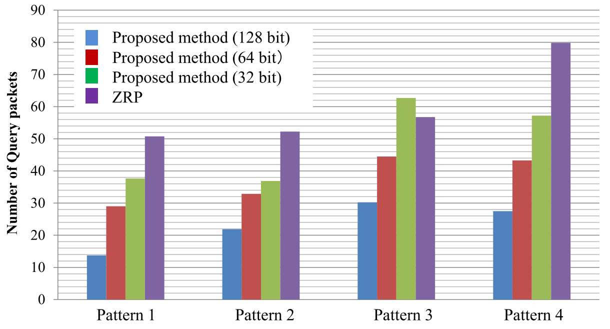


Fig. 7 Number of RouteQuery packets used to search for a route in the simulation (Number of hash functions is 3).

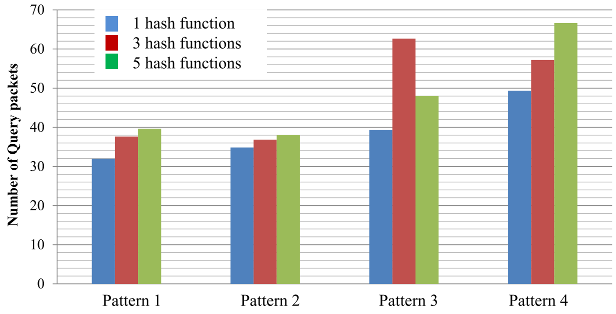


Fig. 8 Number of RouteQuery packets used to search a route in the simulation (Bloom filter length is 32 bits).

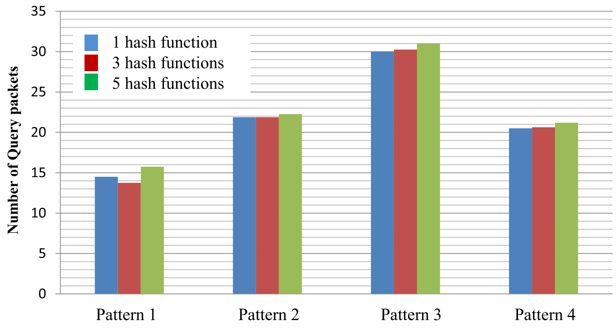


Fig. 9 Number of RouteQuery packets used to search for a route in the simulation (Bloom filter length is 128 bits).

© 2018 Information Processing Society of Japan

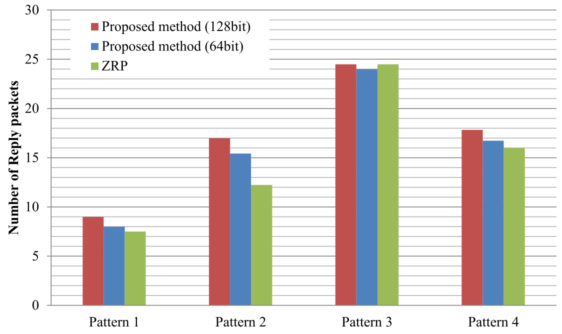


Fig. 10 Number of RouteReply packets used to search for a route in the simulation (Number of hash function is 3).

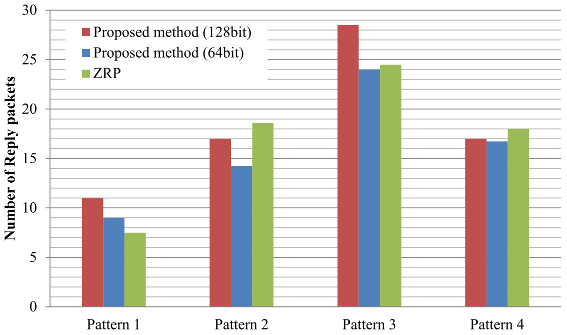


Fig. 11 Number of RouteReply packets used to search for a route in the simulation (Number of hash function is 5).

from 1 to 5. If the Bloom filter length is long enough (128 bits), the effect of the number of hash functions is small. However, if the Bloom filter is short (32 bits), the effect of the number of hash functions is large. From Fig. 9, when the number of the hash functions is too large, false positives occurs and the number of query packets increases.

5.3 Number of Reply Packets

Figures 10 and **11** show the number of reply packets for each communication pattern in the case of three and five hash functions and 128 bit and 64 bit Bloom filter size. From this result, it can be seen that the proposed scheme does not contribute to reducing the total number of ZRP reply packets. It can be considered that the search of a Bloom filter is effective only when transmitting/forwarding the query packet for grasping the position of the destination node. Since the destination node uses the same method as the conventional method when sending the reply packet to the sender node after the route has been found, the effect of the performance improvement by reducing the number of reply packets cannot be see.

5.4 Number of Hops of the Route

The number of hops of the route found by the proposed method (Bloom filter lengths are 128 bits and 64 bits) and ZRP is shown in **Table 2**. From the simulation results, no difference is seen between 128 bits and 64 bits of the proposed method. For communication pattern 2, the route of the proposed method is longer than that of ZRP. In this case, because the root node of the tree structure in the proposed method is the center of the area, the route searched by the proposed method passes through the center of the area. Therefore, the route found is far from the shortest path. For the other communication pattern, the route lengths of the proposed method and ZRP are not different. The reason considered is that the three communication patterns are on a path near the center of the region.

Table 2 Number of hops of route.

	128 bit	64 bit	ZRP
Pattern 1	8	8	8
Pattern 2	16	16	**8**
Pattern 3	16	16	16
Pattern 4	12	12	12

Because the proposed method uses a tree structure to find a route, the length of the route might be longer than that of ZRP. To find a short route, the root of the tree needs to be close to the center of the region. The selection method of the root node of the tree is our future work.

6. Conclusions

In this paper, we have proposed an enhancement of the IERP routing protocol by using a Bloom filter. In the enhanced IERP, a tree structure using the peripheral nodes of a zone is constructed. The Bloom filter is made by using the node IDs in a zone and is exchanged across the tree structure. The received Bloom filter is set as node information for the direction of a link. The filter decides the forwarding of a control packet to the link. Therefore, the number of control packets decreases as compared with the bordercast approach.

In a simulation, the performance of the proposed protocol is compared with the original IERP in terms of the number of control messages. In the simulation, 128-, 64-, and 32-bit lengths of the Bloom filter are used. The number of hash functions is 1, 3, and 5. From the simulation results, 74% of the control messages could be reduced when the number of hash functions is 1 with the 128-bit Bloom filter. As the Bloom filter size is reduced, the false positive probability of the Bloom filter increases, and the reduction rate of control packets declines.

In the evaluation, the number of control packets of the tree construction is not included, because the number of times to construct the tree structure is assumed to be very small relative to the number of times to explore the route. Our future work includes more detailed evaluations, including evaluating the number of control packets to construct the tree. An evaluation in a more realistic environment taking node mobility and layout into account is also a future challenge.

Acknowledgments Part of this work was supported by JSPS KAKENHI Grant Number 16H02813 and was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

References

[1] Sun, J.-Z.: Mobile Ad-Hoc Networking: An Essential Technology for Preserving Computing, *Proc. International Conference on Info-tech and Info-net*, Vol.3, pp.316–321 (2001).

[2] Rahman, A.H.A. and Zukarnain, Z.A.: Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks, *European Journal of Scientific Research*, Vol.31 No.4, pp.566–576 (2009).

[3] Johnson, D.B., Maltz, D.A. and Hum, Y.-C.: The Dynamic Source

Routing Protocol for Mobile Ad Hoc Networks, IETF Internet Draft (Apr. 2003).

[4] RFC3561: Ad hoc On-Demand Distance Vector (AODV) Routing, pp.1–37 (2003).

[5] Haas, Z.J., Pearlman, M.R. and Samar, P.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, draft-ietf-manet-zone-zrp-04.txt (July 2002).

[6] Koyama, A., Arai, J., Barolli, L. and Durresi, A.: EZRP: An Enhanced Zone-Based Routing Protocol for Ad-Hoc Networks, *Journal of Concurrency: Practice and Experience*, Vol.19, No.8, pp.1157–1170, Wiley InterScience (2007).

[7] Osanai, H., Koyama, A. and Barolli, L.: An Implementation and Evaluation of Zone-Based Routing Protocol for Moblie Ad-hoc Networks, *International Conference of Network-Based Information Systems* (*NBiS2011*), pp.517–522 (2011).

[8] Osano, T., Uchida, Y. and Ishikawa, N.: Routing Protocol using Bloom Filters for Mobile Ad Hoc Networks, *The 4th International Conference on Mobile Ad-hoc and Sensor Networks* (*MSN 2008*), pp.89–94 (2008).

[9] Bloom, B.: Space/Time Tradeoffs in Hash Coding with Allowable Errors, *Comm. ACM*, Vol.14, No.7, pp.422–426 (1970).

[10] Haas, Z.J., Pearlman, M.R. and Samar, P.: The Intrazone Routing Protocol (IARP) for Ad Hoc Networks, IETF Internet Draft (July 2002).

[11] Haas, Z.J., Pearlman, M.R. and Samar, P.: The Interzone Routing Protocol (IERP) for Ad Hoc Networks, IETF Internet Draft (July 2002).

[12] Haas, Z.J., Pearlman, M.R. and Samar, P.: The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks, IETF Internet Draft (June 2001).

Yuria Oigawa received B.S. degree from Toho University, Japan, in 2016. She currently works for NTT DoCoMo. Her research areas of interest include communication protocols, ad-hoc networks, sensor networks.

Fumiaki Sato received B.E. degree from Iwate University, Japan, in 1984. He received M.E. and D.E. degrees from Tohoku University, Japan, in 1986 and 1992 respectively. In 1986, he joined Mitsubishi Electric Corporation. From 1995–2005, he worked Shizuoka University. He is now a Professor in the department of Information Science of Toho University, Japan. He teaches undergraduate and graduate in Computer Networks, Mobile Computing, and Ad-hoc Networks. His research areas of interest include communication protocols, ad-hoc networks, sensor networks, P2P systems, network security, distributed processing systems, and communication software design. He is a member of IPSJ, IEICE, IEEE and ACM.

© 2018 Information Processing Society of Japan