Opportunistic Routing Using Prioritized Forwarders with Retransmission Control

Taku Yamazaki[†], Ryo Yamamoto[††,‡], Takumi Miyoshi[‡‡,‡], Takuya Asaka[*], and Yoshiaki Tanaka[†,‡]
[†] Department of Communications and Computer Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169–8555 Japan
[††] Graduate School of Information Systems, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182–8585 Japan
[‡] Global Information and Telecommunication Institute, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169–8555 Japan
[‡‡] College of Systems Engineering and Science, Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, 337–8570 Japan
[*] Faculty of System Design, Tokyo Metropolitan University
6-6 Asahigaoka, Hino-shi, Tokyo, 191–0065 Japan
Email: taku_yamazaki@aoni.waseda.jp, ryo_yamamoto@is.uec.ac.jp,
miyoshi@shibaura-it.ac.jp, asaka@tmu.ac.jp, ytanaka@waseda.jp

***Abstract*—In ad hoc networks, broadcast based forwarding protocols called OR (opportunistic routing) have been proposed. In general OR protocols, each receiver makes a forwarding decision using a random backoff time based on distance information autonomously. However, the potential forwarders must wait for the expiration of the backoff timer before the packet forwarding. Moreover, it is difficult to gain forwarding path diversity under sparse environments. In this paper, we propose a novel forwarder selection method for OR. In the proposed method, a terminal called prioritized forwarder, which forwards packets without using the backoff time, is selected among neighbours. In addition, we integrate a hop-by-hop retransmission control in the proposed method, which improves the packet transmission success rate under sparse environments. We evaluate the proposed method in addition to the conventional protocols by using computer simulation.**

***Index Terms*—ad hoc networks, opportunistic routing, prioritized forwarder, hop-by-hop retransmission control.**

I. Introduction

Ad hoc networks form self-distributed networks using mobile terminal such as smartphones without relying on infrastructures. However, owing to unstable wireless communications and terminal mobility, the network topology varies over time. In ad hoc networks, unicast routing protocols [1], [2] establish a specified transmission route between a source and a destination before the source initiates data transmission have been proposed. After the route establishment, data packets are forwarded to the destination along the route. However, these protocols are fundamentally designed to use the established route at the route discovery process continuously until the route is broken even if there exists better route. Moreover, a route re-establishment occurs frequently due to link disruptions under a high mobility or a poor wireless environment.

To overcome these problems, broadcast based forwarding protocols called OR (opportunistic routing) have been proposed. In a wireless communication, every terminal in a sender's communication range can receive the same packet at the same time due to the broadcast nature of wireless communications. By using this characteristic, OR can perform a packet forwarding using multiple receivers among neighbours without relying on a pre-specified route used in unicast routing protocols. The receivers make a forwarding decision based on a various metric (e.g. hop count, packet transmission success rate, signal strength, and geographical information). Thus, it gains forwarding path diversity and redundancy by selecting eligible forwarders according to the metrics.

As an OR protocol, ExOR (extremely opportunistic routing) [3] that forwards a packet based on a priority defined by a forwarder list on a packet header has been proposed. SSR (Self-selective routing) [4] and LFBL (listen first, broadcast later) [5] that forward packets using a backoff time based on the distance information among terminals also have been proposed. However, ExOR is difficult to adapt to mobile environments since the forwarder list becomes obsolete under such environments. SSR and LFBL may increase delay and decrease transmission efficiency according to the increase of hop counts since every forwarder must wait for the expiration of the backoff timer before forwarding packets. Moreover, SSR and LFBL decrease a packet transmission success rate with the decrease of terminal density since a few neighbours exists between a source and a destination in such a sparse area and there are not sufficient path diversity.

In this paper, we propose a novel forwarder selection method called PRIOR (prioritized forwarding for opportunistic routing). For a reduction of the backoff time, each forwarder selects a next hop prioritized forwarder that performs a packet forwarding without using the backoff time among neighbours. Moreover, to overcome the problem that decreases packet transmission success rate under sparse environments, we integrate a hop-by-hop retransmission control in PRIOR.

II. Related Works

A. ExOR

As mentioned in the previous section, ExOR selects multiple forwarders based on a forwarder list on a packet header. The forwarder list that is created by a source contains candidates of forwarders in the order of forwarding priorities determined by ETX (expected transmission count), which is calculated based on the transmission success rate. For the first sequence of a packet forwarding in ExOR, a source creates the forwarder list in a packet header before the source transmits the packet. On receiving the packet, each receiver checks the forwarder list on the packet header and forwards the packet in the order of the forwarding priority. If the terminal receives the packet from another higher priority terminal before the own packet forwarding, the terminal regards the packet as an acknowledgement and abort the packet forwarding on the terminal. Therefore, by selecting several forwarders based on the forwarding priority explicitly, ExOR can forward packets using stable path and improve the packet transmission success rate. However, computational complexity increases as a network scales due to ETX calculation. Moreover, ETX varies frequently over time under mobile environments such as ad hoc networks due to radio interference and terminal mobility. Namely, it makes ExOR difficult to calculate appropriate ETX.

B. Lightweight Forwarding Protocols

SSR and LFBL have been proposed as a lightweight forwarding protocol for ad hoc networks. In these protocols, every forwarder calculates a backoff time based on distance information in its distance table and a received packet header before the packet forwarding. The distance table contains a destination address, distance to the destination, TTL (time to live), and sequence number. Note that, the distance metric means logical distance in these protocols. In SSR, the hop count is used as the distance metric. LFBL uses signal strength as well as hop counts for the distance metric. To make it simple, we give an explanation how the protocols work when they only use hop counts as a distance metric. First, if a source does not have distance information to a destination on its distance table before the source initiates a data transmission process, the source performs request packet flooding towards the destination. On receiving the request packet, if the receiver has not received it before, the receiver records the distance to the source on its distance table and broadcasts the packet after a random time. Otherwise, the receiver discards the request packet. If the destination address is own address, the terminal broadcasts the reply packet towards the source. Here, the reply packet is forwarded using the same method as the data packet forwarding since at least the reverse forwarding path is already established during the request packet flooding. On receiving the reply packet or a data packet, each receiver calculates a backoff time according to the distance information. Here, the closer the distance to the destination, the shorter the backoff time becomes. In addition, these protocols add a random value to the backoff time to avoid collision among the forwarders. In SSR, if the forwarder receives the same packet from another terminal that is closer to the destination during the backoff time, the terminal broadcasts the acknowledgement packet to reduce unnecessary packet forwarding. Therefore, in SSR and LFBL, each receiver decides whether to forward packets or not autonomously instead of selecting forwarders explicitly. Namely, SSR and LFBL make an implicit forwarding decision in contrast to ExOR. Thus, each receiver autonomously decides whether to forward packets or not using the backoff time. However, each forwarder must wait for the expiration of the backoff timer on every packet forwarding. That may cause random collisions with an increase of forwarder candidate and delay may increase with an increase hop count. Moreover, these protocols may be not able to forward packets using multiple forwarders in a sparse area where a few neighbours exist between a source and a destination. In other words, if there are not sufficient terminals to gain path diversity, these protocols may decrease the packet transmission success rate.

III. The Proposed Method

A. Concept

Section II described the conventional OR protocols, however, these protocols have several disadvantages on computational complexity and path diversity. In this section, we propose a novel forwarder selection method called PRIOR. In PRIOR, a forwarder specifies single terminal as a PF (prioritized forwarder) that forwards a packet without using the backoff time. Moreover, every terminal checks whether packets are forwarded from the PF or not. To adapt to dynamic topology change in ad hoc networks, PRIOR introduces autonomous PF update mechanism. In addition, PRIOR performs a packet retransmission mechanism that can improve an end-to-end packet transmission success rate under sparse environments where the conventional OR protocols degrades their performance due to the lack of path diversity.

B. Opportunistic Routing Using Prioritized Forwarders

As mentioned above, PRIOR uses a PF that is able to transmit a packet without using a backoff time. A forwarder, which forwards a packet toward a destination, explicitly selects the PF among neighbours and adds its address in the packet header. Here, the PF is updated every time the packets are hopped. To perform this function, we modify a distance table used in LFBL. The distance table includes the PF address and UTX (unprioritized transmission count) as well as destination address, distance to the destination, and sequence number. UTX is used for updating the PF and detail description will be in III-C. In this subsection, we describe the proposed destination discovery process, a way of setting the PFs, and data transmission process as follows:

Before a source initiates a data transmission process, the source performs the request packet flooding toward a destination and waits the reply packet from the destination if the source does not have distance information to the destination. The request packet contains a destination address, source address, distance to the source, TTL, and sequence number.

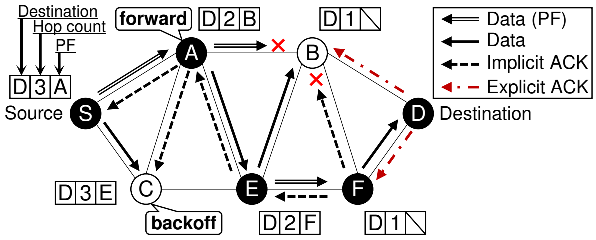

Fig. 1. Example of the packet forwarding in PRIOR.

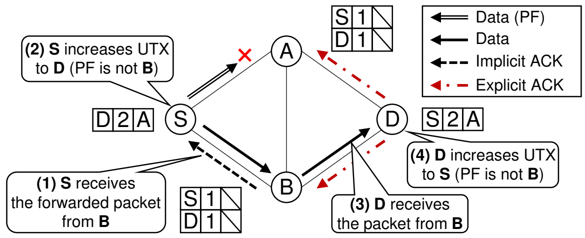

Fig. 2. Example of updating UTX in PRIOR.

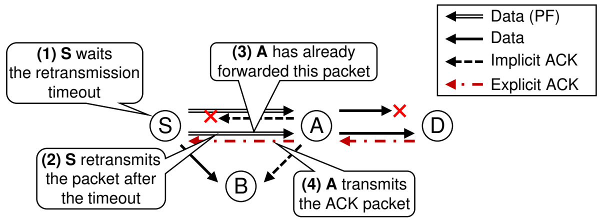

Fig. 3. Example of retransmission control in PRIOR.

On receiving the request packet, each receiver checks its information in the packet header. If the receiver does not have the distance information to the source, the receiver records the information on its distance table. Then, the receiver adds the forwarder address of the request packet as the PF. After that, the receiver updates the distance information to the source in the request packet with the information in own distance table and rebroadcasts the request packet. At this time, if the receiver has already received the same request packet, the receiver discards the packet. If the source does not receive any reply packet in a certain period, the source performs the request packet flooding with increased sequence number and TTL to expand the flooding area.

When the request packet reaches the destination, the destination broadcasts a reply packet toward the source only when the first request packet arrived. At the same time, the arrival of the request packet means that terminals between the source and the destination that forward the request packet have the distance information to the source since they have recorded the distance information during the forwarding. It also means that the reverse forwarding path is already established at least. Therefore, the reply packet can be forwarded to the source in the same way of data packet transmission using PFs with the reverse path. Figure 1 shows the example of the packet forwarding in PRIOR. On receiving a packet, each receiver checks that they have already received the packet or not. If not, the receiver checks the PF address in the packet header and forwards the packet without using a backoff time only when the PF address is coincided with own address. Otherwise, the receiver forwards the packet using a backoff time in same way as the conventional protocols. Then, each forwarder updates the PF address on the packet header with a new one that is chosen from own distance table.

C. Updating Prioritized Forwarders

In PRIOR, each forwarder must determine a next hop PF for a better forwarder selection. It means that the forwarder always requires that the next hop PF to keep being in a suitable position between the forwarder and the destination. However, the PF may get out of a previous hop forwarders' communication range or move to an ineligible position for packet forwarding due to terminal mobility. Therefore, the topology changes make PFs to be obsolete. To adapt to those topological changes, PRIOR adaptively changes the PFs. To realize the function, UTX, which represents the number of received packets sent from non-PFs for each pair of a source and a destination, is introduced on every terminal. Figure 2 shows the example of the sequence of PF update. During packet transmission, a forwarder decreases UTX to the destination of the packet if a PF forwards it. Otherwise, the forwarder increases UTX. Moreover, when the forwarder of the received packet is the PF to the source, the receiver decreases UTX to the source. Otherwise, the receiver increases UTX. If UTX reaches a threshold, the receiver changes the PF to the forwarder of the received packet and the receiver initializes UTX to 0. Note that these UTX increments and decrement is applied only once to a single sequence number.

D. Hop-by-Hop Retransmission Control

For dealing with the decrease of the packet transmission success rate under sparse environments, we integrate a hop-by-hop retransmission control into PRIOR. Figure 3 shows the example of the retransmission sequence. First, a forwarder sets a retransmission timeout and waits the expiration of the timer after a packet transmission. If the forwarder receives the same packet before the timer expires, the forwarder checks whether the forwarder of the received packet is closer to the destination than the forwarder or not. If it is, the forwarder regards the packet as the acknowledgement and finishes the retransmission control. When the timer expires, if the forwarder has not received the same packet yet, the forwarder regards the packet as a loss and increases the retransmission count and resets the retransmission timeout timer after retransmitting the packet. At that time, if the retransmission count reaches threshold that represents the maximum retransmission count, the terminal stops the retransmission control and the terminal discarded the packet. During the packet retransmission, if all of neighbours have already finished forwarding the same packet, the packet will never be forwarded. As a result, the forwarder misunderstands that the packet is lost. For solving this problem, a receiver transmits an acknowledgement packet to a non forwarder on receiving the packet that has already forwarded if it is forwarded from the further terminal.

IV. Performance Evaluation

A. Simulation Setups

In this paper, we evaluate the performance of AODV (Ad hoc on-demand distance vector), LFBL, and PRIOR using two computer simulations. The common environments are as follows: We use QualNet 5.1 [6] as a network simulator. Every terminal uses IEEE 802.11b and its transmission rate is set to 11 Mbps. We disable RTS/CTS (Request to send / Clear to send). In PRIOR, UTX threshold is set to 3. For the data transmission, we generate bidirectional traffic using UDP (user datagram protocol) and the pair of terminals transmits 1 Mbyte data each other. We also have done the simulation with and without the retransmission control in AODV and PRIOR except LFBL. Although AODV itself does not have a function of retransmission control, AODV uses built-in ARQ on IEEE 802.11 MAC. On the other hand, LFBL does not have the function of the retransmission control since LFBL only uses broadcast that does not use ARQ on IEEE 802.11 MAC. Thus, LFBL cannot uses retransmission controls.

*1) **Simulation 1**:* In simulation 1, we evaluate the impact of the terminal density change to the performance. In this simulation, terminals are placed randomly in 1,000 m $\times$ 1,000 m simulation area. Their communication range is set to 100 m. In AODV and PRIOR with retransmission controls, the maximum retransmission count is set to 1. The pair of terminals for generating bidirectional traffic is randomly chosen from all terminals. A random waypoint mobility model is used and the moving speed is randomly chosen from 0 m/s to 10 m/s without using waiting times. We change the number of terminals from 40 terminals to 180 terminals at every 20 steps.

*2) **Simulation 2**:* In simulation 2, we evaluate the impact of the maximum retransmission count change to the performance. Figure 4 shows the simulation topology. In the simulation, 25 terminals are placed into 5×5 grid with 100 m clearance and theirs communication range is set to about 292 m. The pair of terminals for generating bidirectional traffic is set to the edge terminal and the opposite edge of the grid. We change the maximum retransmission count from 0 to 6. As mentioned above, LFBL does not have the function of retransmission

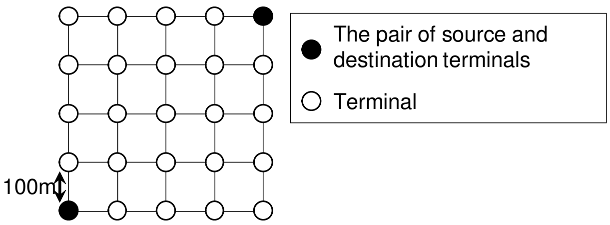

Fig. 4. Performance evaluation 2: simulation topology.

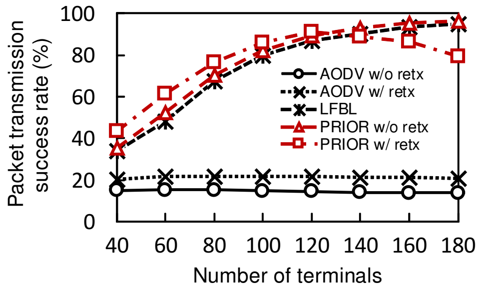

Fig. 5. Performance evaluation 1: packet transmission success rate.

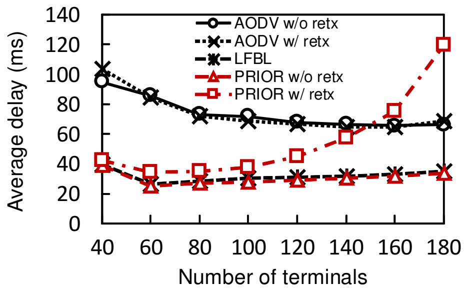

Fig. 6. Performance evaluation 1: average delay.

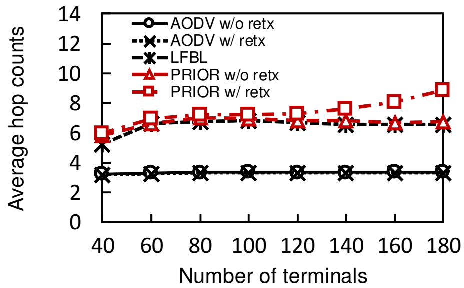

Fig. 7. Performance evaluation 1: average hop count.

controls, and thus the simulation result only shows when the maximum retransmission count is 0.

In these simulations, we evaluated these protocols from the following viewpoints: packet transmission success rate, end-to-end delay, and hop count. Note that, it is important that we do not include out of order packets in the simulation results. Namely, a destination discards the packet that comes out of order even if the destination has not received yet. Therefore, the destination only receives newer data packets that have a greater sequence number than the previously received packet.

B. Simulation Results

*1) **Simulation 1**:* Figures 5–7 show the results of simulation 1. Figure 5 shows that PRIOR with retransmission control achieves the highest packet transmission success rate from 40 to 120 terminals. Generally, each terminal does not have sufficient neighbours to gain path diversity in its communication range under sparse environments. Therefore, PRIOR improves the reliability of each hop by using retransmission control, and that improves the end-to-end packet transmission success rate. On the other hand, LFBL and PRIOR without retransmission control achieve higher packet transmission success rate than PRIOR with the retransmission control under the dense environment such as 140 to 180 terminals. This is because the potential forwarders increase as an increase of terminal density, and thus forwarders are able to make a forwarding decision by their own. Therefore, packets can be forwarded correctly without using the retransmission control. In contrast, PRIOR with retransmission control decreases the packet transmission success rate. PRIOR consumes network resources since each forwarder performs the retransmission control that decreases the packet transmission success rate under dense environments.

Figure 6 shows that LFBL and PRIOR without retransmission control lower the delay than the others. However, PRIOR with retransmission control increases the delay as the terminal density increases since its retransmission control consumes the network resource in excess.

Figure 7 shows that LFBL and PRIOR increase the hop count compared with AODV. AODV establishes a path between the source and the destination as short as possible. In contrast, LFBL and PRIOR make a forwarding decision based on the distance information by each receiver autonomously that is able to select the forwarding path adapting to environments at that moment. It may be able to select the shortest path in the best situation, however, it is difficult to always select the shorter path since the packet reception rate changes according to physical distance. In other words, these protocols tend to select a closer terminal that has a high packet reception rate. Therefore, LFBL and PRIOR may increase hop count comparing to AODV.

*2) **Simulation 2**:* Figures 8–10 show the results of simulation 2. Figure 8 shows that AODV and PRIOR improve the end-to-end packet transmission success rate by retransmitting lost packets autonomously. In AODV, the packet success rate increases as the maximum retransmission count increases. On the other hand, PRIOR does not gain the packet success rate improvement even if the maximum retransmission count is increased. This can be explained by the characteristic of the OR as follows. Although PRIOR performs the retransmission control to avoid packet losses, the retransmitter must wait the expiration of the retransmission timer for each retransmission. It incurs newer packet to detour around the retransmitters toward the destination. Namely, these packets would have arrived at the destination out of order. As a result, the packet transmission success rate does not increase even if the maximum retransmission count is increased.

Figure 9 shows that LFBL and PRIOR decrease the delay compared with AODV. In these protocols, a backoff time for each packet forwarding causes the delay increase while the packet transmission success rate is improved. That can reduce ETX, and thus the delay is decreased.

Figure 10 shows that the hop count of LFBL and PRIOR are higher than AODV. As mentioned above, LFBL and PRIOR tend to select the forwarder that has a high packet reception rate and that may increase the hop count.

V. Conclusion

In this paper, we proposed PRIOR and evaluated PRIOR with the conventional protocols by using the computer simulations. From the simulation results, PRIOR realizes a more efficient forwarder selection in mobile environments. It can improve the end-to-end packet transmission success rate and decreases transmission delay. Moreover, the integrated hop-by-hop retransmission control improves the end-to-end packet transmission success rate under sparse environments. For the future work, adaptation of hop-by-hop retransmission control to dense environments needs to be discussed.

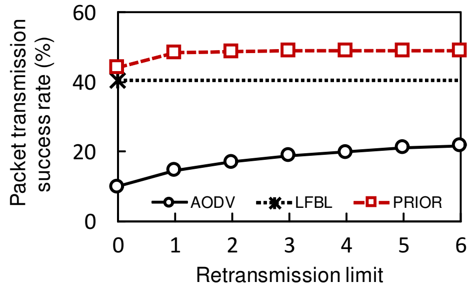

Fig. 8. Performance evaluation 2: packet transmission success rate.

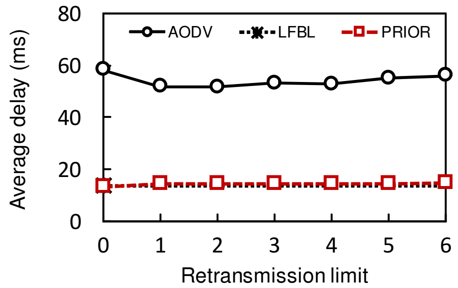

Fig. 9. Performance evaluation 2: average delay.

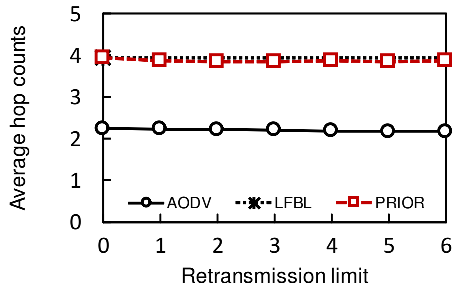

Fig. 10. Performance evaluation 2: average hop count.

References

[1] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," IETF, RFC 4728, Feb. 2007.

[2] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," Proc. 2nd IEEE Workshop on Mobile Comput. Syst. and Applications (WMCSA 1999), pp.90–100, New Orleans, USA, Feb. 1999.

[3] S. Biswas and R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks," ACM SIGCOMM Comput. Commun. Rev., vol.35, no.4, pp.133–144, Oct. 2005.

[4] G.G. Chen, J.W. Branch, and B.K. Szymanski, "Self-selective routing for wireless ad hoc networks," Proc. IEEE Int. Conf. Wireless and Mobile Comput. Netw. and Commun. (WiMob 2005), vol.3, pp.57–64, Montreal, Canada, Aug. 2005.

[5] M. Meisel, V. Pappas, and L. Zhang, "Listen first, broadcast later: topology-agnostic forwarding under high dynamics," Proc. Annual Conf. of Int. Tech. Alliance (ACITA 2010), London, UK, Sept. 2010.

[6] QualNet, website. http://web.scalable-networks.com/content/qualnet/