PAPER *Special Section on Parallel and Distributed Computing and Networking*

Decentralized Local Scaling Factor Control for Backoff-Based Opportunistic Routing

Taku YAMAZAKI[†,††a)], **Ryo YAMAMOTO**[†††,††], ***Members***, **Genki HOSOKAWA**[††††], ***Nonmember***, **Tadahide KUNITACHI**[††††], ***Member***, ***and*** **Yoshiaki TANAKA**[†††††,††], ***Fellow, Honorary Member***

SUMMARY In wireless multi-hop networks such as ad hoc networks and sensor networks, backoff-based opportunistic routing protocols, which make a forwarding decision based on backoff time, have been proposed. In the protocols, each potential forwarder calculates the backoff time based on the product of a weight and global scaling factor. The weight prioritizes potential forwarders and is calculated based on hop counts to the destination of a sender and receiver. The global scaling factor is a predetermined value to map the weight to the actual backoff time. However, there are three common issues derived from the global scaling factor. First, it is necessary to share the predetermined global scaling factor with a centralized manner among all terminals properly for the backoff time calculation. Second, it is almost impossible to change the global scaling factor during the networks are being used. Third, it is difficult to set the global scaling factor to an appropriate value since the value differs among each local surrounding of forwarders. To address the aforementioned issues, this paper proposes a novel decentralized local scaling factor control without relying on a predetermined global scaling factor. The proposed method consists of the following three mechanisms: (1) sender-centric local scaling factor setting mechanism in a decentralized manner instead of the global scaling factor, (2) adaptive scaling factor control mechanism which adapts the local scaling factor to each local surrounding of forwarders, and (3) mitigation mechanism for excessive local scaling factor increases for the local scaling factor convergence. Finally, this paper evaluates the backoff-based opportunistic routing protocol with and without the proposed method using computer simulations.
key words: *ad hoc network, wireless sensor network, opportunistic routing, backoff time, scaling factor control, binary feedback, duplicate packet forwarding*

1. Introduction

Wireless multi-hop networks, such as ad hoc networks and sensor networks, form self-distributed networks without relying on the network infrastructure. However, the communication quality of the terminals varies due to the radio interferences and terminal mobility. Furthermore, a method of replacing wire harnesses by wireless networks has been studied as one of the applications of wireless multi-hop networks [1]–[4]. The routing protocols should dynamically select a forwarding path dynamically in such fast-changing network environments.

Unicast-based routing protocols [5]–[7] that forward packets along a specific route have been proposed as a general routing protocol for ad hoc and sensor networks. However, due to the characteristics of using fixed routes in unicast-based routing protocols, it is difficult for them to adapt to unpredictable changes occurred in radio environments since they alter their established route only when the route is broken.

To adapt to temporary wireless quality variations, support methods for unicast-based routing [3], [4], [8]–[11] establish a detouring path around a temporarily disrupted path and shortcut an established path based on the route information that has been obtained from a routing protocol. However, they only focus on forwarding path changes based on the established route. Therefore, they cannot make a flexible forwarding path selection without performing route reconstruction.

To achieve flexible forwarding path selection, opportunistic routing protocols have been proposed [12], [13]. In opportunistic routing protocols, every receiver makes the packet forwarding decision based on various metrics (e.g., hop count, packet transmission success rate, geographical information, and so on) when packets are received to exploit the broadcast nature of wireless communications. Hence, the opportunistic routing protocols enable dynamic and flexible packet forwarding path selection without a route reconstruction at the packet level.

As general opportunistic routing protocols, the expected transmission count (ETX)-based opportunistic routing protocols have been proposed [14]–[18]. They make the forwarding decision based on a metric that is represented as ETX [19] which represents the expected number of transmissions to send a packet to the destination and is calculated from a packet transmission success rate and hop count. However, it is difficult for them to adapt their metrics to fast radio environmental changes since the metrics becomes obsolete immediately.

As another forwarder selection approach, location-based opportunistic routing protocols [20]–[23], which

Manuscript received January 7, 2019.
Manuscript revised June 9, 2019.
Manuscript publicized July 17, 2019.
[†]The author is with the College of Systems Engineering and Science, Shibaura Institute of Technology, Saitama-shi, 337–8570 Japan.
[††]The authors are with the Global Information and Telecommunication Institute, Waseda University, Tokyo, 169–8555 Japan.
[†††]The author is with the School of Informatics and Engineering, The University of Electro-Communications, Chofu-shi, 182–8585 Japan.
[††††]The authors are with the YAZAKI Research and Technology Center, YAZAKI CORPORATION, Yokosuka-shi, 239–0847 Japan.
[†††††]The authors are with the Department of Communications and Computer Engineering, Waseda University, Tokyo, 169–8555 Japan.
a) E-mail: taku@shibaura-it.ac.jp
DOI: 10.1587/transinf.2019PAP0004

forward a packet based on geographical information, have been proposed. However, they require every terminal to have the global positioning system (GPS) function for acquiring positions of the terminals. However, the utilization of GPS function limits the applications of multi-hop networks because it is difficult to accurately track positions of terminals in some situations such as indoor environments.

In case of wireless sensor networks and ad hoc networks, backoff-based opportunistic routing protocols that makes a forwarder selection based on the backoff time obtained by each potential forwarder using hop counts have been proposed [24]–[30]. In the protocols, each potential forwarder calculates a backoff time based on the product of a weight and global scaling factor during the packet forwarding procedure. The weight prioritizes potential forwarders and is calculated based on hop counts to the destination of a sender and receiver. The global scaling factor is a predetermined value to map the weight to the actual backoff time.

However, the backoff-based opportunistic routing protocols have three common issues associated with the usage of the predetermined global scaling factor. The first issue is that the necessity of unifying the global scaling factor for the backoff time calculation among all the terminals. The second issue is that the impossibility to change the value of the global scaling factor after the network is initiated. The third issue is that the difficulty associated with setting the global scaling factor to the appropriate value.

To address the aforementioned issues of the backoff-based opportunistic routing protocols associated with the global scaling factor, this paper proposes a novel decentralized local scaling factor control for backoff-based opportunistic routing protocols without relying on the predetermined global scaling factor. The proposed method sets an appropriate scaling factor for each local surrounding based on the binary state that is defined as re-receiving a single forwarded packet or multiple forwarded packets. Thus, the proposed method contains the following: (1) sender-centric local scaling factor setting mechanism in a decentralized manner where each forwarder specifies a local scaling factor for sharing the value of the local scaling factor among the receivers; (2) adaptive scaling factor control mechanism that increases or decreases the local scaling factor based on the binary state that is defined as re-receiving a single forwarded packet or multiple forwarded packets; and (3) mitigation mechanism for excessive local scaling factor increases for the local scaling factor convergence.

The remainder of this paper can be organized as follows. Section 2 introduces the common operations that are performed in backoff-based opportunistic routing protocols. Section 3 proposes a novel decentralized local scaling factor control for backoff-based opportunistic routing protocols. Section 4 evaluates the behavior and performance of the backoff-based opportunistic routing with and without the proposed method using computer simulations. Section 5 presents the conclusion and an introduction to our future work.

2. Backoff-Based Opportunistic Routing

This section introduces the common model of backoff-based opportunistic routing protocols [24]–[30] and their common operations. Note that we will consolidate and/or alter the original names, formulas, and variables in the original papers to easily compare the protocols in terms of their procedures based on the description of the paper [30].

The backoff-based opportunistic routing protocols generally contains the following two routing phases: phase for discovering a destination that can be referred to as the "discovery phase" and phase for forwarding the reply packets and data packets that can be referred to as the "data phase." In addition, every terminal has "cost table," which has entries that contain at least an address to the destination, a hop count, a sequence number, and a lifetime of the former information. Each receiver updates the entries with information from a source that can be obtained from the header of packets, namely the reverse path information of the original path. Therefore, they should be used with bidirectional flows between the source and the destination.

Discovery phase: If a source does not have a valid cost entry for the desired destination in its table before initiating the packet transmission, the source initiates a request packet flooding towards the destination. Note that the request packet must contain at least a source address, destination address, hop count of the packet traversed, sequence number, and time to live (TTL). When the destination receives the request packet, the destination broadcasts a reply packet towards the source with the reverse path. Then, each receiver performs forwarding in the same way as that of data packet forwarding since at least one reverse forwarding path to the source has already been established during the request packet flooding. Thus, the receivers can transit to the data phase for forwarding the reply packet. If the source would not receive any reply packet after a certain period, the source begins the request packet flooding again.

Data phase: After receiving a reply packet or a data packet, each receiver makes the forwarding decisions based on the backoff time. Then, the reply packets and data packets contain at least a source address, destination address, hop count to the destination, hop count of the packet traversed, and sequence number. After receiving these packets, each receiver r calculates the expected hop count $\hat{h}_{rd}$ to the destination d and the hop count difference δ_r based on Eq. (1) and Eq. (2),

$$\hat{h}_{rd} = h_{id} - 1, \tag{1}$$

$$\delta_r = h_{rd} - \hat{h}_{rd}, \tag{2}$$

where h_{id} denotes the hop count between the previous forwarder i and the destination d that was recorded in the packet as the hop count to the destination during the previous forwarding. h_{rd} denotes the hop count between the receiver r and the destination d recorded in the cost table of the receiver r. Each receiver r calculates the backoff time b_r based on δ_r as described below, then it waits for the expiration of the backoff time b_r so that it can become a "potential forwarder." If the potential forwarder receives the same packet while it is waiting for the backoff time expiration, it considers the packet to be an implicit acknowledgement (ACK) and cancels the forwarding because the other terminal has already forwarded the same packet. Hence, it is no longer considered to be a potential forwarder.

Backoff time calculation: As one of the backoff time calculation mechanisms in backoff-based opportunistic routing, this paper introduces backoff time calculation based on the sigmoid function [30]. During this calculation, a receiver r calculates the fixed backoff time based on the sigmoid function, and it adds a random backoff time based on the change in the sigmoid function. The fixed backoff time $\varsigma_r(\delta_r)$ can be calculated as

$$\varsigma_r(\delta_r) = \frac{1}{1 + \exp(-\alpha\,(\delta_r - \beta))}, \tag{3}$$

where α denotes the gain of $\varsigma_r(\delta_r)$ and β denotes the parameter that is used to shift the inflection point. A receiver r calculates the fixed backoff time $\varsigma_r(\delta_r)$ for determining the priority of packet forwarding based on the sigmoid function according to δ_r. Hence, for each δ_r, $\varsigma_r(\delta_r)$ is the lower bound of the backoff time. Furthermore, r adds a random backoff time to the fixed backoff time based on the change in the sigmoid function. The random backoff time $\mu_r(\delta_r)$ can be calculated as

$$\mu_r(\delta_r) = u(\varsigma_r\,(\delta_r + \gamma) - \varsigma_r\,(\delta_r)), \tag{4}$$

where γ denotes the boundary between slots. u denotes a uniform random number from $(0, 1)$. Using $\varsigma_r(\delta_r)$ and $\mu_r(\delta_r)$, the backoff time b_r can be calculated as

$$b_r = T_{\text{global}}(\varsigma_r\,(\delta_r) + \mu_r\,(\delta_r)), \tag{5}$$

where T_{global} denotes the global scaling factor which is a predetermined value and is shared globally among all the terminals. Finally, the receiver r calculates the backoff time b_r by computing the product of the global scaling factor T_{global} and the sum of the fixed backoff time $\varsigma_r(\delta_r)$ and $\mu_r(\delta_r)$.

According to the aforementioned operations, backoff-based opportunistic routing protocols perform dynamic path selections based on the backoff time calculation. However, there are three common issues that are associated with the practical use of the global scaling factor T_{global} as follows.

1. **Necessity of unifying the global scaling factor among all the terminals:** The backoff-based opportunistic routing protocols realize both the reduction of duplicate packet forwarding among potential forwarders who have the same hop count and the prioritization based on the hop count by using backoff time. However, to realize them, it is necessary to unify a global scaling factor of all the terminals to correctly select an appropriate forwarder since each potential forwarder needs to share the same calculation range of the backoff time among all the terminals based on the global scaling factor.
2. **Impossibility to change the global scaling factor after the network is initiated:** The global scaling factor directly affects the forwarder selection, and hence it is important factor for the backoff-based opportunistic routing protocols to appropriately select a forwarder among potential forwarders. However, it is not possible to change the global scaling factor from the predetermined value during the networks are being used due to the first issue in spite of directly affecting the forwarder selection.
3. **Difficulty associated with setting the global scaling factor to an appropriate value:** The appropriate value of the global scaling factor varies across various factors such as a local surrounding, data rate, and so on. However, it is difficult to set an appropriate value for the global scaling factor in spite of being not able to change the value due to the second issue.

Hence, it is required to decentralize and control the scaling factor locally to relieve the backoff-based opportunistic routing protocols of the global scaling factor for improving the applicability.

3. Decentralized Local Scaling Factor Control

3.1 Concept

To address the aforementioned issues of the conventional protocols, this paper proposes a novel decentralized local scaling factor control for backoff-based opportunistic routing protocols without using a predetermined global scaling factor to adapt to each local surrounding of forwarders. The proposed method comprises the following three mechanisms.

(1) Sender-centric local scaling factor setting mechanism. The first mechanism relieves the backoff-based opportunistic routing protocols from the usage of a predetermined global scaling factor to use a local scaling factor in a decentralized manner instead of using global scaling factor. It also enables to adapt to each local surrounding of senders since they specify their local scaling factor when the packet forwarding. Therefore, this mechanism realizes the reduction of both unnecessary packet forwarding and redundant backoff delay to set the appropriate value for each local surrounding of forwarders.

(2) Adaptive local scaling factor control mechanism. The second mechanism enables each forwarder to control the variation of its own local scaling factor based on the binary state of receiving either a single forwarded packet or multiple forwarded packets. Therefore, this mechanism can empower the first mechanism to ensure the adaptation of the local scaling factor without using any manual settings. Hence, the combination of mechanisms (1) and (2) solve the aforementioned common issues and perfectly relieve the backoff-based opportunistic routing protocols from the global and manual configurations of the scaling factor.

(3) Mitigation mechanism for excessive local scaling factor increases. The third mechanism provides a threshold for the local scaling factor convergence based on random values, which are stored in packets to mitigate the excessive local scaling factor increases. It can reduce the backoff delay to probably accept duplicate packet forwarding.

By using the above mechanisms, the proposed method realizes the decentralized and adaptive scaling factor control for each local surrounding of forwarders. The details of each mechanism are described in Sect. 3.3–3.5.

3.2 Backoff-Based Opportunistic Routing with Local Scaling Factor

To realize the above mechanisms, the proposed method requires that each terminal should have an additional field to store a local scaling factor for each cost entry in its cost table. In addition, the proposed method requires that the two additional fields in the data packet header, a local scaling factor τ_k and a random value u, are recorded and calculated by the forwarder, respectively. The fields in a packet header are updated by each forwarder while forwarding the packet. The operations of the backoff-based opportunistic routing protocol using the proposed method can be given as follows.

Discovery phase: A source initiates a communication to the destination based on the request packet flooding that is similar to that in the conventional methods as described in Sect. 2. Then, each terminal records its initial local scaling factor τ_0 to the destination and its initial slow start threshold s_{th} to the destination as T_{init} in its cost table along with the information that is used in conventional backoff-based opportunistic routing protocols. Note that the aforementioned initialization is described in Sect. 3.4.

Data phase: When a terminal receives a reply or data packet, it calculates δ_r based on Eq. (1). Next, using δ_r and the local scaling factor τ_k recorded in the packet header, it calculates a backoff time b_r based on the backoff time calculation (such as Eq. (5)) instead of using a global scaling factor T_{global} and waits for the expiration of backoff time. Note that the detailed sender-centric local scaling factor setting mechanism is described in Sect. 3.3. Subsequently, if the potential forwarder has not received an implicit ACK, it forwards the packet after updating the local scaling factor field and the random value field in the packet header. Note that the local scaling factor τ_k is stored in the cost entry to the destination and the random value u is the same one as the value used in the backoff time calculation, respectively. After the packet is forwarded, the forwarder checks the binary state of the reception of either a single forwarded packet or multiple forwarded packets. Based on the binary state, the proposed method increases or decreases its local scaling factor τ_{k+1}. Note that the local scaling factor control mechanism is described in Sect. 3.4 in detail.

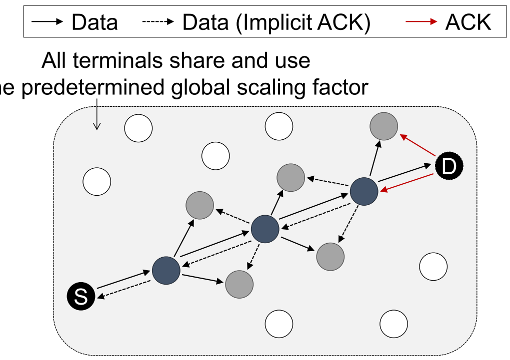


Fig. 1 Example of packet forwarding using the predetermined global scaling factor in backoff-based opportunistic routing.

3.3 Sender-centric Decentralized Local Scaling Factor Setting Mechanism

In conventional backoff-based opportunistic routing protocols, a predetermined global scaling factor has to be set among all the terminals for performing the backoff time calculation as shown in Fig. 1. In other words, potential forwarders must wait for the backoff time based on the global scaling factor even though it is not necessary for a potential forwarder such as a single potential forwarder to wait for the backoff time. Therefore, either an unnecessary backoff delay may be imposed because of the large global scaling factor or the unnecessary duplicate packet forwarding may be increased because of the small global scaling factor. Although the conventional protocols share the global scaling factor among all the terminals, it is sufficient to share the same scaling factor only among the receivers in the light of the characteristics of backoff-based opportunistic routing protocols.

To exploit the characteristics of backoff-based opportunistic routing protocols, this paper proposes a sender-centric local scaling factor setting mechanism in which a local scaling factor is specified by a sender without using the predetermined global scaling factor. In this mechanism, each sender has a local scaling factor for each destination, which is recorded in each cost entry in the cost table. When a sender forwards a packet, it updates the local scaling factor field of the packet header with its own scaling factor to the destination. Therefore, the sender specifies and notifies its local scaling factor for calculating the backoff time of the receivers as shown in Fig. 2. Hence, this mechanism realizes flexible scaling factor choices to ensure that the scaling factor can be used to adapt to each local surrounding. Note that the adaptive local scaling factor control mechanism is described in detail in Sect. 3.4.

3.4 Adaptive Local Scaling Factor Control Mechanism

To exploit the mechanism of Sect. 3.3, each forwarder sets a local scaling factor to adapt their local surroundings, and also be able to autonomously alter the value. This

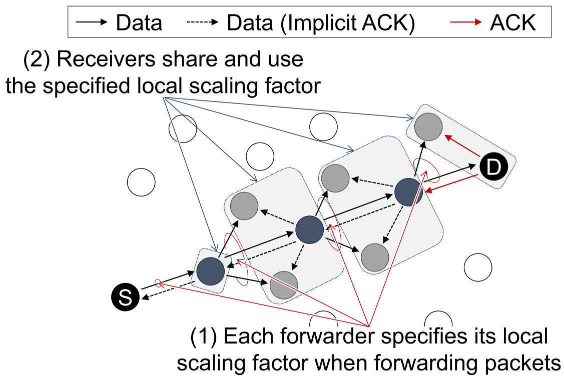


Fig. 2 Example of packet forwarding using a local scaling factor in backoff-based opportunistic routing.

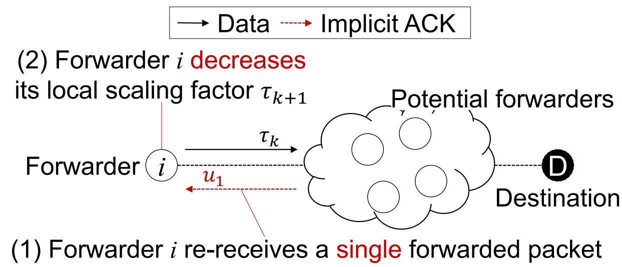


Fig. 3 Example of single packet forwarding.

section proposes an adaptive local scaling factor control mechanism based on binary feedback control algorithms that are inspired by the additive increase/multiplicative decrease (AIMD) algorithm [31].

Generally, the AIMD algorithm, which is one of the binary feedback control algorithms, has been extensively used in congestion avoidance algorithms [32]–[35] including transmission control protocol (TCP) congestion control [36], [37]. In a general AIMD algorithm in TCP, a terminal increases the congestion window size based on the additive function using the last congestion window size when there is no congestion. However, when the congestion is detected by observing the packet loss, a terminal decreases the congestion window size based on multiplicative function using the last window size. Therefore, the congestion avoidance algorithms could simplify their control just to observe the simple binary condition related to the occurrence of a packet loss.

In this paper, we have assumed the application of a binary feedback algorithm with local scaling factor control. In a practical situation, even though the backoff times of the opportunistic routing protocol calculated using potential forwarders are considered to be close, duplicate packet forwarding may occur because the general datalink layer protocol of the wireless medium such as 802.11 protocols [41] contains a function to avoid frame collisions based on the carrier sense multiple access/collision avoidance (CSMA/CA) algorithm. In the light of the characteristics, this paper assumes that a datalink layer protocol can somehow avoid frame collisions based on its collision avoidance algorithm. Therefore, the receivers can correctly receive the packets transmitted by several senders at almost the same time based on the backoff algorithm of the datalink layer protocol.

To exploit the above characteristics of the collision avoidance, the proposed method uses a binary state for receiving either a single forwarded packet or multiple forwarded packets by observing the forwarded packets after forwarding. First, each forwarder sets an initial value T_{init} as the local scaling factor τ_0. In general, the AIMD algorithm uses a slow start algorithm [38], [39]. Therefore, T_{init} should also be a larger value to a certain extent. After a packet is forwarded, the forwarder verifies and counts the number of re-forwarded packets that are forwarded to them by the next-hop forwarders during a certain period. Then, the forwarder increases or decreases its local scaling factor τ_{k+1} for the next time $k+1$ to use the functions that is defined below based on the above binary state of receiving either the single forwarded packet or multiple forwarded packets. If the forwarder only receives the single forwarded packet during the period, it decreases its scaling factor τ. If the forwarder receives the multiple forwarded packets during the period, it increases its scaling factor τ. Otherwise, it ignores the packet for adjusting the scaling factor control. The detailed increasing and decreasing functions are the following.

Decreasing function: When the forwarder only receives the single forwarded packet, which is the same as the transmitted packet, the forwarder estimates that the transmitted packet may be forwarded only by a single forwarder. Then, it is assumed that the collision avoidance can be achieved among the potential forwarders since the local scaling factor is sufficiently large or excessively larger than an ideal value. Hence, as shown in Fig. 3, the forwarder decreases its scaling factor τ that can be calculated as

$$\tau_{k+1} = \begin{cases} \tau_k - T_{\text{dec}} n_{\text{dec}}^2 & (\tau_k > s_{\text{th}}) \\ \tau_k - T_{\text{dec}} & (\tau_k \leq s_{\text{th}}) \end{cases}, \tag{6}$$

where, T_{dec}, decides the sensitivity of decreasing the scaling factor and denotes the unit of decreasing local scaling factor τ_k. n_{dec} denotes the continuous decreasing count, and s_{th} denotes the slow start threshold. T_{inc}, which is a predetermined value, is obtained based on the unit time of the datalink layer protocol such as the slot time, the short inter frame space (SIFS), the DCF inter frame space (DIFS), and so forth. If the current value of τ_k is larger than the current s_{th}, the forwarder exponentially decreases τ based on the continuous decreasing count. If the current value of τ_k is same or smaller than s_{th}, the forwarder linearly decreases τ for suppressing the excessive decreases.

Increasing function: In contrast with the aforementioned situation, when the forwarder receives multiple packets that are similar to the transmitted packet, the forwarder estimates that the transmitted packet may be forwarded by multiple forwarders. Figure 4 shows an example of re-receiving multiple forwarded packets. Therefore, the collision avoidance among the potential forwarders based on the backoff time is assumed to fail since the scaling factor is smaller than the ideal value. Hence, before τ is updated, the forwarder updates the threshold s_{th} that can be calculated as

$$s_{\mathrm{th}} = (\tau_k + s_{\mathrm{th}})\,\zeta, \tag{7}$$

where ζ $(0 < \zeta < 1)$ denotes the decreasing factor of s_{th}. Generally, ζ is set to be 0.5 for setting the subsequent s_{th} as the middle point between τ_k and the last s_{th}. After updating s_{th}, the forwarder increases the next local scaling factor τ_{k+1} using the threshold s_{th} that can be calculated as

$$\tau_{k+1} = \begin{cases} s_{\mathrm{th}} & (\tau_k < s_{\mathrm{th}}) \\ \tau_k + \eta \bar{R}_k & (\tau_k \geq s_{\mathrm{th}}) \end{cases}, \tag{8}$$

where, $\bar{R}_k$ denotes the smoothed link round-trip time and η $(0 < \eta \leq 1)$ denotes the decreasing factor of $\bar{R}_k$. Here, based on the calculation [40], the smoothed link round-trip time $\bar{R}_k$ can be calculated as

$$\bar{R}_k = (1 - \rho)\,\bar{R}_{k-1} + \rho R_k, \tag{9}$$

where ρ denotes a weight for the calculation of an exponential weighted moving average of the link round-trip time $\bar{R}_k$. If τ_k is same or larger than s_{th} when duplicate packet reception occurs, the forwarder should sufficiently increase its local scaling factor. Hence, the above equation uses the smoothed link round-trip time since a sufficient amount of time is added depending on the data rate of the wireless medium and the local surrounding.

3.5 Mitigation Mechanism for Excessive Local Scaling Factor Increases

As mentioned above, the proposed method ensures that the adaptive local scaling factor control mechanism increases or decreases the local scaling factor based on the binary state. However, the local scaling factor is difficult to converge on a certain value for avoiding all of the collisions. Therefore, the larger backoff delay is caused since the local scaling factor becomes an excessively large value. In other words, although there is a trade-off between the reduction of duplicate packets and backoff delay by setting the local scaling factor, it is not possible to choose to prioritize which index due to the above reason.

To alleviate the issue of the delay increase, this section proposes the mitigation mechanism for an excessive local scaling factor that increases based on the difference between random values. First, a terminal forwards a packet and subsequently waits for a re-reception of the same packet for a certain time. After that, when a terminal receives multiple forwarded packets as shown in Fig. 4, it calculates the difference between the random values u_{diff} $(-1 \leq u_{\mathrm{diff}} \leq 1)$ as

$$u_{\mathrm{diff}} = u_i - u_1, \tag{10}$$

where u_1 represents the random value that is stored in the initially received packet, and u_i $(i \in \mathbb{N} \wedge i \neq 1)$ represents the random value stored in the second or the successively received packet. Then, if the forwarder observes that the absolute value of the difference $|u_{\mathrm{diff}}|$ is larger than the threshold u_{th}, the proposed method increases the local scaling factor τ_{k+1}. Otherwise, the proposed method ignores the packet for adjusting the scaling factor control. This mechanism can alleviate an excessive increase in the scaling factor permitting a certain ratio of duplicate packet forwarding, and then it reduces the backoff delay because of the backoff time. Hence, this mechanism can adjust the trade-off between the reduction of duplicate packets and backoff delay based on the threshold u_{th}.

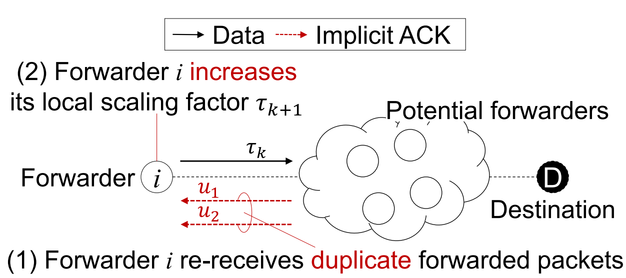


Fig. 4 Example of duplicate packet forwarding.

4. Performance Evaluation

4.1 Simulation Setup

The computer simulations have evaluated the performance of an opportunistic routing protocol with and without the proposed method.

Common environment: This simulation used QualNet [42] as the network simulator. Every terminal used IEEE 802.11b [41] and disabled the request to send/clear to send (RTS/CTS) function. In this simulation, although the backoff time calculation mechanism is based on [30], the remaining functions, such as prioritized forwarder (PF) and retransmission control, were disabled except for an explicit ACK function. In the backoff time calculation, α and γ were set to 1 to avoid backoff time range overlapping. To maximize the random backoff amount of 1-hop close terminals, β was set to 0.5. In the proposed method, we set $\xi = 0.5$, $\eta = 0.2$, $\rho = 0.1$, $\tau_{\mathrm{init}} = 200$ ms, and $T_{\mathrm{dec}} = 50$ µs based on DIFS in IEEE 802.11b. The simulations generated single bidirectional traffic using user datagram protocol (UDP) [43], which consists of 1,000 bytes of 5,000 packets in each flow between S and D. The simulations have used the topology that is shown Fig. 5.

Simulation 1: This simulation evaluated the performance and convergence of the local scaling factors of S and D

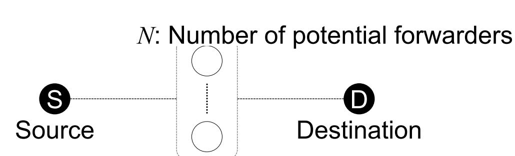


Fig. 5 Simulation topology.

varying the number of potential forwarders over time. The data rate of IEEE 802.11b was set to 11 Mbps. In this simulation, the number of potential forwarders N is changed from $1 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 1$ at every 100 seconds interval. This simulation evaluated the following items: (a) the transition of local scaling factor; (b) the transition of link round-trip time; (c) the distribution of u_{diff} and the transition of $\bar{u}_{\mathrm{diff}}$; and (d) the transition of the smoothed random value of the first re-received packet $\bar{u}_1$, and the end-to-end evaluation items (the average packet transmission success rate, average transmission delay, and total number of forwarded packets). Note that the smoothed values $\bar{u}_{\mathrm{diff}}$ and $\bar{u}_1$ were weighted moving average of u_{diff} and u_1 using the weight ρ, respectively.

Simulation 2: This simulation evaluated the impact of the variation in the number of potential forwarders when applying various data rates in IEEE 802.11b since the effectiveness of the proposed method should be verified when the characteristics of the wireless medium are changed. In this simulation, the number of potential forwarders N was varied from 1 terminal to 10 terminals for each data rate. This simulation evaluated the following items: (a) the average local scaling factor, (b) the total number of forwarded packets and (c) the average transmission delay.

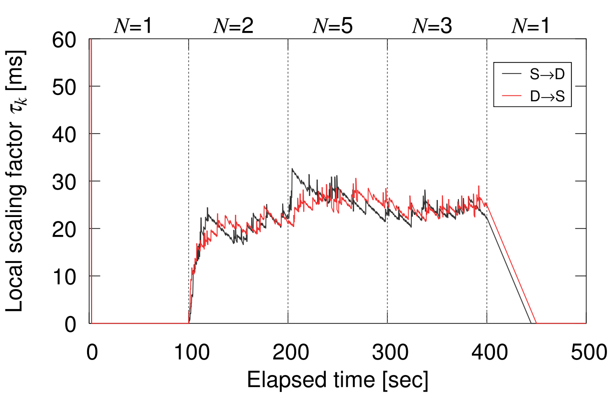


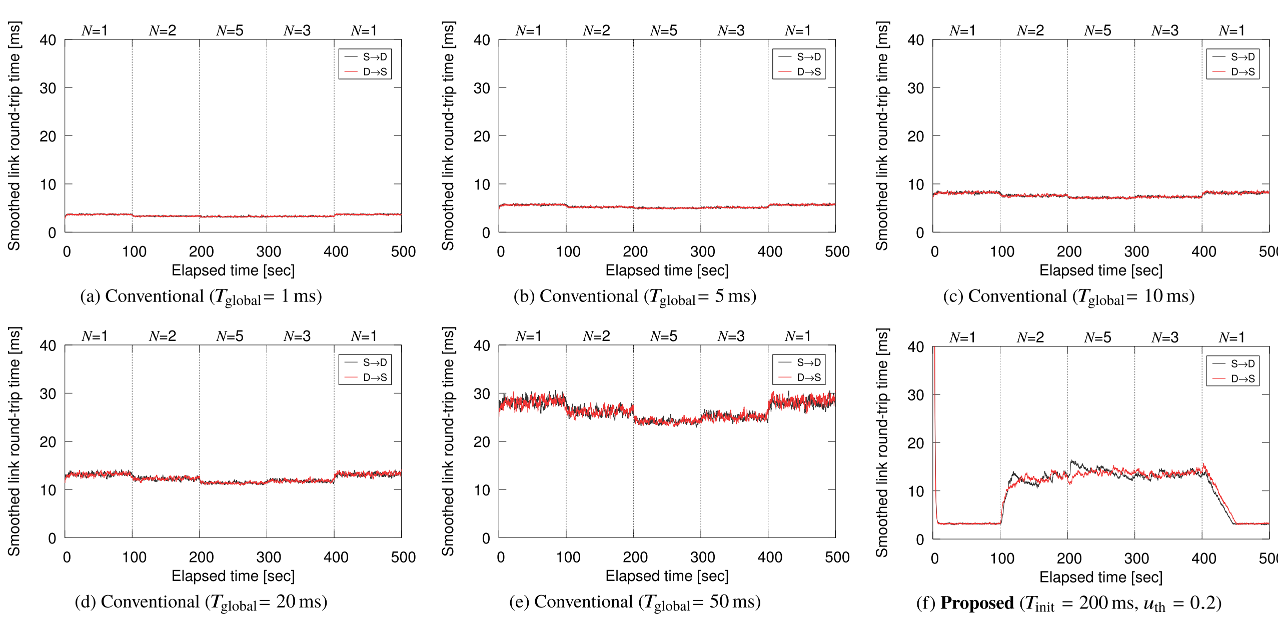
Fig. 6 Simulation 1: transition of the local scaling factor τ_k in the proposed method ($\tau_{\mathrm{init}} = 200$ ms and $u_{\mathrm{th}} = 0.2$).

4.2 Simulation Results

Simulation 1

Figure 6 shows the transition of the local scaling factor of terminals S and D. As can be seen from the result, the proposed method adaptively adjusts scaling factor with to the varied the number of potential forwarders. In particular, the proposed method reduces unnecessary backoff time by setting the local scaling factor τ_k to the minimum value when N becomes 1 since there is no necessity of avoiding duplicate packet forwarding. Moreover, S and D increase their local scaling factor when the number of potential forwarders increases since the potential forwarders must sense and cancel duplicate packet forwarding with respect to each other.

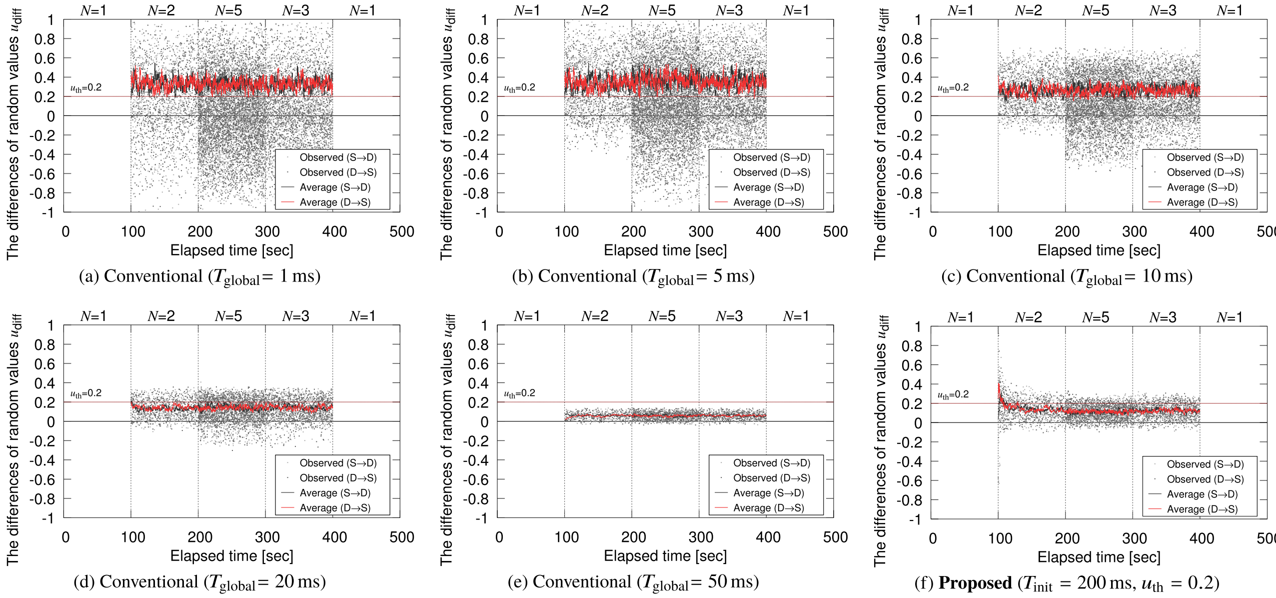
Figure 7 (a)–(f) show the transition of the smoothed link round-trip time. In the conventional method, the link round-trip time considerably fluctuates when the T_{global} is set to 1 ms. This is because the delay is irregularly imposed due to the traffic load derived from the duplicate packet forwarding. As the global scaling factor T_{global} increases, the backoff time of each potential forwarder also increases. As a result, the link round-trip time significantly increases regardless of the number of potential forwarders N due to the backoff delay. In the proposed method, the smoothed link round-trip time changes according to the number of potential forwarders N. The proposed method changes the



(a) Conventional (T_{global} = 1 ms) (b) Conventional (T_{global} = 5 ms) (c) Conventional (T_{global} = 10 ms)

(d) Conventional (T_{global} = 20 ms) (e) Conventional (T_{global} = 50 ms) (f) **Proposed** ($T_{\mathrm{init}} = 200$ ms, $u_{\mathrm{th}} = 0.2$)

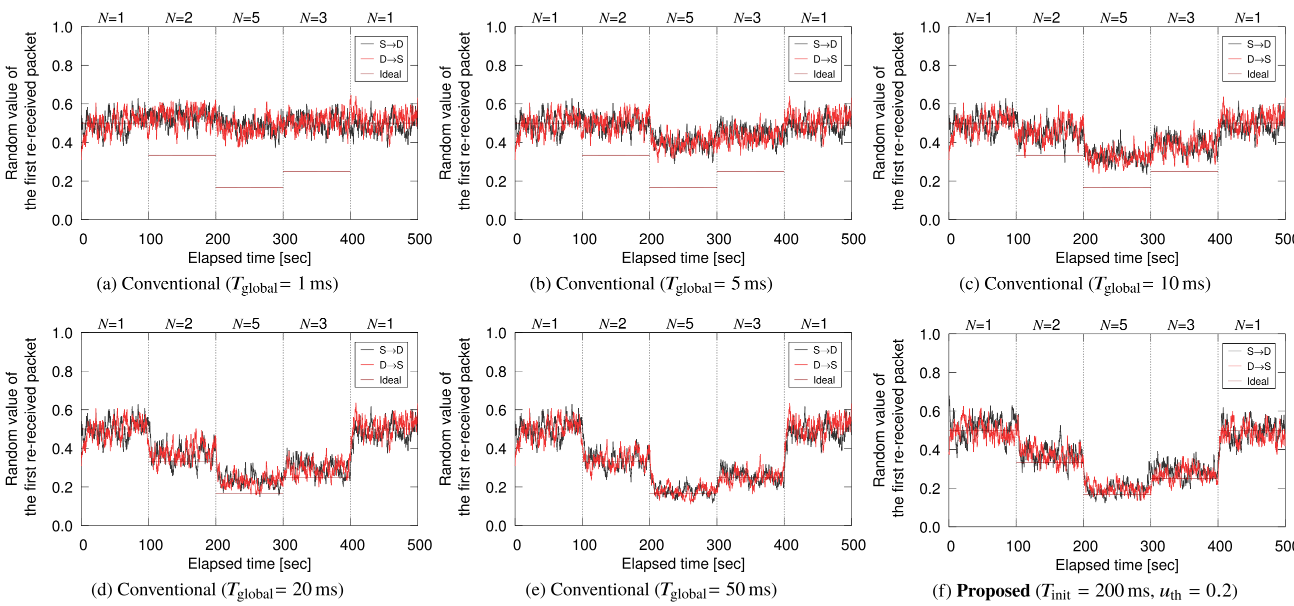
Fig. 7 Simulation 1: transition of the smoothed link round-trip time $\bar{R}_k$.



(a) Conventional (T_{global} = 1 ms) (b) Conventional (T_{global} = 5 ms) (c) Conventional (T_{global} = 10 ms)

(d) Conventional (T_{global} = 20 ms) (e) Conventional (T_{global} = 50 ms) (f) **Proposed** (T_{init} = 200 ms, u_{th} = 0.2)

Fig. 8 Simulation 1: distribution of the difference u_{diff} and transition of the smoothed difference $\bar{u}_{\mathrm{diff}}$.



(a) Conventional (T_{global} = 1 ms) (b) Conventional (T_{global} = 5 ms) (c) Conventional (T_{global} = 10 ms)

(d) Conventional (T_{global} = 20 ms) (e) Conventional (T_{global} = 50 ms) (f) **Proposed** (T_{init} = 200 ms, u_{th} = 0.2)

Fig. 9 Simulation 1: distribution of the random value in the first re-received packet u_1.

local scaling factor based on the binary state of receiving a single forwarded packet or multiple forwarded packets. Therefore, if there is a necessity of avoiding duplicate packet forwarding, the link round-trip time increases according to an increase of the local scaling factor. In contrast to the above situation, if there is no necessity of avoiding duplicate packet forwarding such as $N = 1$, the link round-trip time decreases according to the decrease in the local scaling factor. Namely, although the transmission delay of the conventional method depends on the value of the global scaling factor, that of the proposed method maintains the small transmission delay as possible based on local environments by automatically setting the local scaling factor.

Figure 8 (a)–(f) show the distribution of u_{diff} and the transition of $\bar{u}_{\mathrm{diff}}$ with the line of u_{th} as 0.2. Then, $\bar{u}_{\mathrm{diff}}$ ideally converges to $\frac{u_{\mathrm{th}}}{2}$ if the scaling factor is set to the ideal value. The proposed method decreases or increases the local scaling factor by ignoring duplicate packet forwarding when $\bar{u}_{\mathrm{diff}} < u_{\mathrm{th}}$. In other words, u_{th} is the same as the smallest amount of the sensed packet forwarding among potential forwarders. Then, if we assume the random value is uniformly distributed, the $\bar{u}_{\mathrm{diff}}$ will become $\frac{u_{\mathrm{th}}}{2}$. Hence, it is able to examine the validity of the proposed method to observe the transition of $\bar{u}_{\mathrm{diff}}$. In the conventional method, the potential forwarders cannot cancel duplicate packet forwarding appropriately with each other when the global scaling factor T_{global} is small since the difference of backoff time becomes small even though the random values are sufficiently different. As a result, the difference of the random values is widely distributed. When the scaling factor is increased, the distribution range of u_{diff} becomes narrower, and the number of duplicated forwarded packets is reduced. This is because the difference between the backoff times becomes larger as the increase of the global scaling factor T_{global}. Therefore, the difference is sufficient to avoid duplicate packet forwarding in this case even though the difference between the random values is small. In the proposed method, although the distribution range becomes sufficiently wide immediately after the change of the number of potential forwarders N from 1 to the other value, each forwarder can set the local scaling factor close to an appropriate value. Hence, they could cancel the duplicate packet forwarding after the convergence of the local scaling factor.

Figure 9 (a)–(f) show the transition of the random values of the first received packet $\bar{u}_1$. Here, note that $\bar{u}_1$ is related to the number of potential forwarders N. If the scaling factor is almost the same as or larger than the ideal value, the smoothed random value of the first received packet $\bar{u}_1$ converges to $\frac{1}{N+1}$ as red dotted lines shown in the figures. In contrast, when the scaling factor is smaller than the ideal value, $\bar{u}_1$ is close to 0.5 because the potential forwarder is randomly selected owing to inappropriate forwarder selection. In the light of the characteristics, when the global scaling factor becomes a smaller value without applying the proposed method, $\bar{u}$ is close to 0.5 due to the above reason. In contrast to the above, when the global scaling factor becomes large, $\bar{u}$ almost converges to $\frac{1}{N+1}$. Hence, it becomes sufficient to select an appropriate forwarder based on the random backoff time. The proposed method also adapts to the change of the number of potential forwarders changes without using the predetermined large scaling factor by calculating the local scaling factor of sender autonomously. However, the smoothed value is the slightly larger value than the ideal value since the proposed method always searches the appropriate value to increase or decrease the local scaling factor. Therefore, the smoothed value may not become the ideal value although the smoothed value is as close to the ideal value as much as possible.

Table 1 Simulation 1: end-to-end evaluation results.

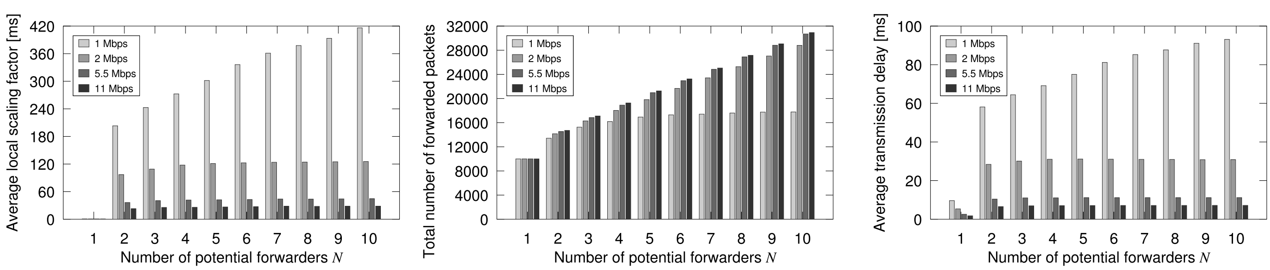
T_{global}	1 ms	5 ms	10 ms	20 ms	50 ms	**Proposed**
Packet transmission success rate [%]	99.60	99.80	99.90	99.96	99.92	**99.88**
Average transmission delay [ms]	1.722	2.669	3.837	6.175	13.159	**4.998**
Total number of forwarded packets	23,995	23,988	20,520	15,642	12,405	**14,756**

Table 1 shows the end-to-end packet transmission success rate, the average end-to-end transmission delay, and the total number of forwarded packets.

In the results of end-to-end packet transmission success rate, the difference is small when the global scaling factor is set to 20 ms or larger in the conventional method. However, the packet transmission success rate becomes slightly smaller when the global scaling factor is set to 1 ms. In this case, some potential forwarders forward packets at the same time since the global scaling factor is quite small. Therefore, packet collisions may occur among them even though the occurrence rate of collisions is considerably small. However, the proposed method maintains almost same or higher packet transmission success rate compared to the conventional method regardless of the global scaling factor.

In the results of the average end-to-end transmission delay, the average delay in the conventional method increases as the global scaling factor increases since the backoff time is redundantly imposed. In contrast, the proposed method could achieve almost the same average delay as the conventional method does at (T_{global} = 10 ms), and therefore we can conclude that the proposed method could reduce the transmission delay by adapting the local surrounding by changing the local scaling factor by each forwarder.

In the results of the total number of forwarded packets, the conventional methods using the predetermined global scaling factor decrease the total number of forwarded packets as the global scaling factor T_{global} increases since the potential forwarders could sense duplicated packet forwarding with each other. In the proposed method, the total number of forwarded packets is at the same level as or is less than that in the conventional method (T_{global} = 20 ms) without almost affecting the transmission delay. In the conventional method, it is necessary for reducing the transmission delay to explicitly decrease the global scaling factor. However, it causes unnecessary packet forwarding due to the small value of the global scaling factor, and hence network load



(a) Average local scaling factor (b) Total number of forwarded packets (c) Average transmission delay

Fig. 10 Simulation 2: evaluation results.

as well as the load of terminals may increase. The proposed method achieves both the small transmission delay and less total number of forwarded packets by automatically setting the local scaling factor by each forwarder even though the number of potential forwarders varies.

Simulation 2

Figure 10 (a) shows the average local scaling factor of each data rate. In the result, the proposed method almost stably adapts the local scaling factor to each local surrounding. In particular, the proposed method works well at data rates of 2 Mbps, 5.5 Mbps, and 11 Mbps. Then, the local scaling factor increases as the number of potential forwarders increases when the data rate is 1 Mbps. This is one of the reasons that T_{dec} is relatively small; thus, the local scaling factor converges to a high value. Therefore, there is a room fofr consideration to set T_{dec} because it correlates with a data rate. To precisely control the local scaling factor, one of the solutions is that an adaptive T_{dec} setting based on wireless airtime of packets if it would be possible to obtain the current data rate and this still remains as future work.

Figure 10 (b) shows the total number of forwarded packets of each data rate. In the result, the total number of forwarded packets increases as the number of potential forwarders increases. This is because the proposed method permits duplicate packet forwarding probabilistically based on the threshold u_{th}. However, when the data rate is 1 Mbps with several potential forwarders, the total number of forwarded packets is smaller than the remaining data rates due to the large local scaling factor.

Figure 10 (c) shows the average transmission delay of each data rate. In the result, the average transmission delay keeps the same value even when the number of potential forwarder increases since the proposed method permits a certain ratio of duplicate packet forwarding due to the same reason as described above. However, when the data rate is 1 Mbps, the average transmission delay increases to set the large local scaling factor although it can reduce the unnecessary packet forwarding.

5. Conclusion

This paper proposed a novel decentralized local scaling factor control for backoff-based opportunistic routing protocols. The proposed method consists of the following three mechanisms: (1) sender-centric local scaling factor setting mechanism; (2) adaptive scaling factor control mechanism based on a binary state of receiving a single forwarded packet or multiple forwarded packets; and (3) mitigation mechanism for excessive local scaling factor increases.

This paper conducted the computer simulations for observing and evaluating the performance of the proposed method. The results showed that the proposed method has been observed to obtain adaptive local scaling factor settings for adapting to the local surroundings of each forwarder. In addition, the computer simulations confirmed that the proposed method could set the scaling factor to an appropriate value when the number of potential forwarders varies dynamically.

Although this paper has evaluated the performance to clarify the primitive characteristics, simulations in realistic environments such as more dynamic situations should be conducted in the future.

In addition, the simulation has revealed that the unit of decreasing local scaling factor correlates with the data rate of forwarders. Therefore, an adaptive method to set the unit based on the wireless airtime should be discussed although the simulation has only used a fixed unit to decrease the local scaling factor. Furthermore, this paper treated a simple local scaling factor control based on a binary state inspired by the AIMD algorithms in case of TCP congestion control. Hence, we should consider and evaluate other algorithms such as the cubic function-based algorithm [44] and estimation-based algorithm [45], [46] in the future.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 17K12680.

References

[1] H. Hatamoto, S. Ano, N. Kikuchi, and S. Shimizu, "An evaluation of transmission performance for wireless harness systems using propagation models in an automobile engine compartment," Proc. 2014 IEEE 24th Annual Int. Symp. Personal, Indoor, and Mobile Radio Commun. (PIMRC 2013), London, UK, pp.117–121, Sept. 2013. DOI: 10.1109/PIMRC.2013.6666115

[2] G. Leen and D. Heffernan, "Vehicles without wires," Comput. Control Eng. J., vol.12, no.5, pp.205–211, Oct. 2001. DOI: 10.1049/cce:20010501

[3] T. Kunitachi, K. Kinoshita, and T. Watanabe, "Experimental study for reliable communications in vehicle," IEICE Trans. Commun. (Japanese Edition), vol.J99-B, no.4, pp.323–333, Jan. 2017.

[4] T. Kunitachi, K. Kinoshita, and T. Watanabe, "An experimental study of reliable wireless communications in vehicles," Proc. 2016 18th Asia-Pacific Netw. Operations and Manag. Symp. (APNOMS 2016), Kanazawa, Japan, Oct. 2016. DOI: 10.1109/APNOMS.2016.7737277

[5] C. Perkins and E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF RFC3561, July 2003. DOI: 10.17487/RFC3561

[6] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," IETF RFC4728, Feb. 2007. DOI: 10.17487/RFC4728

[7] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF RFC3626, Oct. 2003. DOI: 10.17487/RFC3626

[8] Y. Matsuzawa, Y. Yamao, K. Otake, and K. Ishibashi, "Multi-hop transmission performance of integrated dynamic multi-hopping for wireless ad hoc networks," Proc. 2014 IEEE 11th Consum. Commun. and Netw. Conf. (CCNC 2014), Las Vegas, United States, pp.635–640, Jan. 2014. DOI: 10.1109/CCNC.2014.6866639

[9] R. Yamamoto and T. Miyoshi, "Distributed retransmission method using neighbor terminals for ad hoc networks," Proc. 2008 14th Asia-Pacific Conf. Commun. (APCC 2008), Tokyo, Japan, Oct. 2008.

[10] K. Nagao and Y. Yamao, "Cognitive temporary bypassing for reliable multi-hop transmission in wireless ad hoc networks," IEICE Trans. Commun., vol.E93-B, no.12, pp.3391–3399, Dec. 2010. DOI: 10.1587/transcom.E93.B.3391

[11] Z. Wang, Y. Chen, and C. Li, "CORMAN: a novel cooperative op-

portunistic routing scheme in mobile ad hoc networks," IEEE J. Sel. Areas Commun., vol.30, no.2, pp.289–296, Feb. 2012. DOI: 10.1109/JSAC.2012.120207

[12] H. Liu, B. Zhang, H.T. Mouftah, X. Shen, and J. Ma, "Opportunistic routing for wireless ad hoc and sensor networks: present and future directions," IEEE Commun. Mag., vol.47, no.12 pp.103–109, Dec. 2009. DOI: 10.1109/MCOM.2009.5350376

[13] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," IEEE Commun. Surveys Tuts., vol.17, no.4, pp.2214–2241, March 2015. DOI: 10.1109/COMST.2015.2411335

[14] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," ACM SIGCOMM Comput. Commun. Rev., vol.34, no.1, pp.69–74, Jan. 2004. DOI: 10.1145/972374.972387

[15] 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. DOI: 10.1145/1090191.1080108

[16] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," ACM SIGCOMM Comput. Commun. Rev., vol.37, no.4, pp.169–180, Oct. 2007. DOI: 10.1145/1282427.1282400

[17] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, "SOAR: simple opportunistic adaptive routing protocol for wireless mesh networks," IEEE Trans. Mobile Comput., vol.8, no.12, pp.1622–1635, May 2009. DOI: 10.1109/TMC.2009.82

[18] Q. Shen, X. Fang, S. Kim, and R. He, "Combining opportunistic routing with dynamic source routing for wireless mesh networks," Proc. IET Int. Commun. Conf. on wireless Mobile and Comput. (CCWMC 2009), Shanghai, China, pp.61–65, Dec. 2009.

[19] D.S.J.D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Wireless Netw., vol.11, no.4, pp.419–434, July 2005. DOI: 10.1007/s11276-005-1766-z

[20] B. Karp and H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," Proc. 6th Annual Int. Conf. Mobile Comput. Netw. (MobiCom 2000), Boston, United States, pp.243–254, Aug. 2000. DOI: 10.1145/345910.345953

[21] M. Zorzi and R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," IEEE Trans. Mobile Comput., vol.2, no.4, pp.337–348, Oct.–Dec. 2003. DOI: 10.1109/TMC.2003.1255648

[22] M. Zorzi and R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," IEEE Trans. Mobile Comput., vol.2, no.4, pp.349–365, Oct.–Dec. 2003. DOI: 10.1109/TMC.2003.1255650

[23] B. Zhao, R.I. Seshadri, and M.C. Valenti, "Geographic random forwarding with hybrid-ARQ for ad hoc networks with rapid sleep cycles," Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2004), Dallas, United States, vol.5, pp.3047–3052, Nov.-Dec. 2004. DOI: 10.1109/GLOCOM.2004.1378912

[24] 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), Montreal, Canada, vol.3, pp.57–64, Aug. 2005. DOI: 10.1109/WIMOB.2005.1512885

[25] G.G. Chen, J.W. Branch, and B.K. Szymanski, "A self-selection technique for flooding and routing in wireless ad-hoc networks," J. Netw. and Syst. Manag., vol.14, no.3, pp.359–380, Sept. 2006. DOI: 10.1007/s10922-006-9036-7

[26] E. Gelenbe, P. Liu, B.K. Szymanski, and C. Morrell, "Cognitive and self-selective routing for sensor networks," Comput. Manag. Sci., vol.8, no.3, pp.237–258, Aug. 2009. DOI: 10.1007/s10287-009-0102-y

[27] 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.

[28] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," Proc. 5th ACM Int. Workshop on Mobility in the Evolving Internet Architecture (MobiArch 2010), Chicago, United States, pp.3–8, Sept. 2010. DOI: 10.1145/1859983.1859986

[29] T. Yamazaki, R. Yamamoto, T. Miyoshi, T. Asaka, and Y. Tanaka, "Opportunistic routing using prioritized forwarders with retransmission control," Image Processing & Commun., vol.21, no.1, pp.57–66, March 2016. DOI: 10.1515/ipc-2016-0005

[30] T. Yamazaki, R. Yamamoto, T. Miyoshi, T. Asaka, and Y. Tanaka, "PRIOR: Prioritized forwarding for opportunistic routing," IEICE Trans. Commun., vol.E100-B, no.1, pp.28–41, Jan. 2017. DOI: 10.1587/transcom.2016CQP0008

[31] R.N. Shorten, D.J. Leith, J. Foy, and R. Kilduff, "Analysis and design of AIMD congestion control algorithms in communication networks," Automatica, vol.41, no.4, pp.725–730, April 2005. DOI: j.automatica.2004.09.017

[32] K.K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer," ACM SIGCOMM Comput. Commun. Rev., vol.18, no.4, pp.303–313, Aug. 1988. DOI: 10.1145/52325.52355

[33] V. Jacobson, "Congestion avoidance and control," ACM SIGCOMM Comput. Commun. Rev., vol.18, no.4, pp.314–329, Aug. 1988. DOI: 10.1145/52325.52356

[34] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," Comput. Netw. ISDN Syst., vol.17, no.1, pp.1–14, June 1989. DOI: 10.1016/0169-7552(89)90019-6

[35] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," ACM SIGCOMM Comput. Commun. Rev., vol.26, no.3, pp.5–21, July 1996. DOI: 10.1145/235160.235162

[36] J. Postel, "Transmission control protocol," IETF RFC793, Sept. 1981. DOI: 10.17487/RFC0793

[37] S. Floyd, "Congestion control principles," IETF RFC2914, Sept. 2000. DOI: 10.17487/RFC2914

[38] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," IETF RFC2001, Jan. 1997. DOI: 10.17487/RFC2001

[39] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," IETF RFC5681, Sept. 2009. DOI: 10.17487/RFC5681

[40] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's retransmission timer," IETF RFC6298, June 2011. DOI: 10.17487/RFC6298

[41] "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11-2016, pp.1–3534, Dec. 2016. DOI: 10.1109/IEEESTD.2016.7786995

[42] QualNet Network Simulator Software, Scalable Network Technologies Inc., https://web.scalable-networks.com/qualnet-network-simulator-software, Nov. 2018.

[43] J. Postel, "User datagram protocol," IETF RFC768, Aug. 1980. DOI: 10.17487/RFC0768

[44] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," ACM SIGOPS Operating Syst. Rev., vol.42, no.5, pp.64–74, July 2008. DOI: 10.1145/1400097.1400105

[45] L.S. Brakmo, S.W. O'Malley, and L.L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," ACM SIGCOMM Comput. Commun. Rev., vol.24, no.4, pp.24–35, Oct. 1994. DOI: 10.1145/190809.190317

[46] L.S. Brakmo and L.L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet," IEEE J. Sel. Areas Commun., vol.13, no.8, pp.1465–1480, Oct. 1995. DOI: 10.1109/49.464716

Taku Yamazaki received his B.E. and M.S. degrees in electronic information systems from Shibaura Institute of Technology, Tokyo, Japan, in 2012 and 2014, respectively. He received his D.E. degree in computer science and communications engineering from Waseda University, Tokyo, Japan, in 2017. He was a research associate at the Department of Communications and Computer Engineering, School of Fundamental Science and Engineering, Waseda University, from 2015 to 2018. He is currently an assistant professor at the Department of Electronic Information Systems, College of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan. He is also currently an adjunct researcher at the Global Information and Telecommunication Institute, Waseda University, Tokyo, Japan. He received the IEICE Network Systems Research Award in 2014, the CANDAR/ASON Best Paper Award in 2014, the IEICE Young Researcher's Award in 2015, the IEICE Network Software Best Poster Award in 2016, and the IEICE Network Systems Young Researcher's Encouragement Award in 2018. He is a member of IEEE, IEICE, and JSEE.

Ryo Yamamoto received his B.E. and M.E. degrees in electronic information systems from Shibaura Institute of Technology, Tokyo, Japan, in 2007 and 2009. He received D.S. in global telecommunication studies from Waseda University, Tokyo, Japan, in 2013. He was a research associate at Graduate School of Global Information and Telecommunication Studies, Waseda University, from 2010 to 2014, and has been engaged in researching in wireless communication networks. He is presently an associate professor at Graduate School of Informatics and Engineering, The University of Electro-Communications. He received the IEICE young researcher's award in 2010, the IEICE Network System Research Award in 2014, the CANDAR/ASON Best Paper Award in 2014, IEICE Communications Society Distinguished Contributions Award in 2017, IEICE Information and Communication Management Distinguished Contributions Award in 2014 and 2017. His current research interests are ad hoc networks, sensor networks, IoT/M2M networks, and network protocols for the networks. He is a member of IEICE and IEEE.

Genki Hosokawa received the B.E. and M.E. degrees in mechanical engineering from Kansai University, Osaka, Japan, in 2014, 2016, respectively. He joined Yazaki Corporation in 2016, and has been engaged in the research and development of wireless communication control.

Tadahide Kunitachi received the B.E. and M.E. degrees in electronic engineering from Aichi Institute of Technology, Aichi, Japan, in 1994, 1996, respectively. He received the Ph.D. degree in Graduate School of Information Science and Technology from Osaka University, Osaka, Japan, in 2019. He joined Yazaki Corporation in 1996, and has been engaged in the research and development of wireless communication control.

Yoshiaki Tanaka received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1974, 1976, and 1979, respectively. He became a staff at Department of Electrical Engineering, the University of Tokyo, in 1979, and has been engaged in teaching and researching in the fields of telecommunication networks, switching systems, and network security. He is presently a professor at Department of Communications and Computer Engineering, Waseda University. He received the IEICE Achievement Awards in 1980, the Okawa Publication Prize in 1994, the Commendation by Minister for Internal Affairs and Communications in 2009, the IEICE Distinguished Achievement and Contributions Awards in 2013. He is an Honorary Member of IEICE.