Electronic Preprint for Journal of Information Processing Vol.30

Regular Paper

A Proposal of Communication Protocol to Improve the Throughput and Fairness of Multi-hop Wireless Networks and Its Evaluation

Taiki Morita[1,a)] Xuejun Tian[1,b)] Takashi Okuda[1,c)]

Received: May 10, 2021, Accepted: December 3, 2021

Abstract: WLANs have been used in a variety of places. The MAC protocol is an important item of WLANs, and directly affects the transmission efficiency. The distributed MAC protocol has the advantage of not requiring infrastructure such as access points, but it also has the problem that the total throughput decreases significantly when traffic is overloaded due to hidden node problems. In this paper, we focus on the MAC protocol and propose a new MAC protocol that provides a dynamically optimal back-off process in multi-hop wireless networks. To improve hidden node problems, we first conducted a theoretical analysis and found that the average idle slot spacing is a relevant indicator for traffic load. By using the average idle slot spacing and the number of neighbor nodes, the optimal CW (Contention Window) required to achieve high throughput can be configured. This paper compares the simulation results with those of conventional methods and evaluates them in terms of throughput, retransmission attempts, fairness, delay time, and number of collisions. The overall evaluation shows that the MAC protocol proposed in this paper has a better performance than the conventional method.

Keywords: WLANs, Multi-hop, MAC, backoff algorithm, throughput

1. Introduction

Wireless local area networks (WLANs) have been used in a variety of places. Out of two channel access methods DCF (Distributed Coordination Function) and an optional centralized PCF (Point Coordination Function), due to inherent simplicity and flexibility, the DCF is preferred in the case of no base station such as vehicle-to-vehicle (V2V) Communications. Since all the nodes share a common wireless channel with limited bandwidth in the WLANs, it is highly desirable that an efficient and fair Medium Access Control (MAC) scheme is employed. In the case of multi-hop wireless networks, where the transmission range of a node does not cover the entire network area, each node cannot know the current status of the network that the communication performance such as throughput will be degraded. One example of such a problem is the hidden node problem. Li et al. [1] compare various conventional backoff algorithms in various network topologies and show that none of these algorithms are suitable for the DCF wireless network in multi-hop wireless networks. This argues that a simple backoff algorithm, where individual nodes do not understand the network situation, cannot sufficiently mitigate the hidden node problem. Therefore, in order to adapt to the network at an arbitrary time, various researches have been done from various viewpoints to date. Du et al. [2] proposed a novel MAC protocol, Slotted Split-Channel MAC (SSC-MAC) protocol. It is a technique that greatly improves the efficiency of channel access by adjusting the relationship between control, data, and acknowledgement. This MAC protocol tries to improve the throughput by adjusting the slot of each channel, but this MAC protocol is complex. The ITMN (Improving Throughput of Multi-hop wireless networks by acquiring the Number of neighbor nodes) proposed in this paper is concise and effective, in which it adjusts only Contention Window to achieve a high performance. Shahin et al. [3] proposed a back-off mechanism for node congestion and depopulation. By this back-off, it was possible to reduce the unnecessary waiting time when the nodes were depopulated and to suppress the drastic decrease of the throughput when the nodes were dense. Although it is not possible to provide the optimal backoff for each number of nodes, it provides the appropriate backoff according to the traffic volume. There is also a backoff algorithm ACWB that assumes a wireless network with variable packet length [4]. ACWB determines the CW by keeping the number of idle slots within a certain range. ACWB also has the advantage of improving the CW diverging problem. Wu and Xu [5] proposed a Dynamic Adaptive Success-Collision Backoff Algorithm that dynamically adapts the backoff to the wireless network according to the number of successive transmissions and the number of successive collisions. The algorithm is designed to decrease CW when the number of successive transmissions exceeds a certain value and increase CW when the number of successive collisions exceeds a certain value. References [4] and [5] do not provide a backoff process that is optimized to the network conditions, but rather adjusts the frequency of transmission gradu-

1 Graduate School of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi 480–1198, Japan
a) im201012@cis.aichi-pu.ac.jp
b) tan@ist.aichi-pu.ac.jp
c) okuda@ist.aichi-pu.ac.jp

ally based on collisions and other factors. On the other hand, Lei et al. [6] provide a backoff process that is tailored to the network conditions. They propose a backoff algorithm adapted to DCF and EDCA using Access Points (AP). By using this backoff algorithm, the performance is overwhelmingly superior to other methods even at high node density. However, since this method uses APs to determine the number of active nodes, it cannot be implemented in an environment without APs. Therefore, it is necessary to understand the status of the wireless network without APs. Sanada et al. [7] proposed a novel MAC protocol that dynamically adapts the backoff process of each node in multi-hop wireless networks without APs, which is called OBEM. In OBEM, the number of neighbor nodes is dynamically estimated to perform a backoff that suits the network. However, the estimation accuracy of the number of neighbor nodes decreases as the node density increases in OBEM. On the other hand, ITMN can obtain the number of neighbor nodes regardless of the node density. In this paper, we propose a new novel MAC protocol ITMN that sets the optimal CW using the number of neighbor nodes based on OBEM. In proposing ITMN, we improved our model for throughput analysis in multi-hop wireless networks by referring to Refs. [8], [9], [10], [11].

The remainder of this paper is organized as follows. In Section 2, we present in detail our proposed ITMN scheme. Section 3 gives a performance evaluation and discusses the simulation results. Finally, concluding remarks are given in Section 4.

2. Proposed Protocol

In this section, we propose ITMN (Improving Throughput of Multi-hop wireless networks by acquiring the Number of neighbor nodes) that can improve throughput by acquiring the number of neighbor nodes to set CW in multi-hop networks. The proposed ITMN improves OBEM in Ref. [7]. ITMN can be applied to general networks, and the optimal CW can be set dynamically according to different environments.

In multi-hop wireless networks, individual nodes do not understand the status of the network, resulting in many collisions and a significant reduction in throughput. Therefore, it is necessary for each node to set an optimal CW to decrease collisions and perform efficient communication.

Each node just uses the number of intercepting RTS or CTS packets to calculate the number of neighbor nodes. It is a technique that enables high throughput communication by deriving and setting an optimal CW using the number of neighbor nodes.

Section 2.1 describes analysis of throughput in multi-hop networks, Section 2.2 describes setting CW, and Section 2.3 describes how to acquire the number of neighbor nodes.

2.1 Analysis of Throughput in Multi-hop Networks

In the analysis model, the following assumptions are made at each node without loss of generality.

- The transmission, interference and sensing ranges for all network nodes are the same value.
- The density of nodes in a concern area is the same in that, the number of neighbor nodes is the same value, and the number of hidden nodes is also the same value.
- The traffic of each node is saturated so that a node almost has to send requests.
- Each node competes for a channel with the same transmitting and receiving probability p which means transmitting and receiving probability in the idle state.

In addition, we focus on one arbitrary node and name it as a tag node, and its nodes existing within the transmission range of the tag node as a neighbor node. In any time slot, a tag node has three states shown below.

(1) Idle state
This state means a tag node is not sending a packet. When the channel is idle, the tag node counts down its own backoff timer. When the neighbor node transmits and the channel is busy, the tag node's backoff timer does not operate. When the channel is idle again, the countdown of backoff is resumed.

(2) Transmission success state
At the tag node, the backoff timer is 0, and the RTS packet and DATA packet have been successfully transmitted.

(3) Collision state
RTS packet transmission at the tag node fails. In other words, collisions occurred.

The throughput of the tag node can be calculated from the probabilities of idle state, transmission success state, and collision state, which is shown below.

Firstly, we show the probability of idle state. The fact that the tag node is idle means that the tag node neither send RTS packets or DATA packets. When the tag node is in the idle state, the state of the neighbor node can be observed. The state of each neighbor node and its probability are shown below.

(1) The state that neighbor nodes are also idle
All neighbor nodes of the tag node are not transmitting and receiving packets. If this probability is P_{nbr_idl}, it can be expressed as

$$P_{nbr\_idl} = (1 - p)^n \tag{1}$$

(2) The state that one neighbor node is transmitting
It means that tag node receives either RTS only, CTS only, or RTS/CTS pair sent from the neighbor node. If this probability is P_{nbr_snd}, it can be expressed as

$$P_{nbr\_snd} = np(1 - p)^{n-1} \tag{2}$$

(3) The state that two or more neighbor nodes are transmitting
It means that collisions occur between neighbor nodes. If this probability is P_{nbr_col}, it can be expressed as

$$P_{nbr\_col} = 1 - (P_{nbr\_idl} + P_{nbr\_snd}) \tag{3}$$

where n is the number of neighbor nodes around the tag node. When neighbor nodes are also idle, the tag node exists for the time of 1 slot time t_{slt}. Assuming that T_{nbr_snd} and T_{nbr_col} are the time when one neighbor node is transmitting and the time when two or more neighbor nodes are transmitting, respectively, it can be expressed as

$$T_{nbr\_snd} = T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 4\tau + 3SIFS + DIFS \tag{4}$$

$$T_{nbr\_col} = T_{RTS} + \tau + EIFS \tag{5}$$

where T_{RTS}, T_{CTS}, T_{DATA}, and T_{ACK} are the transmission duration for a RTS packet, a CTS packet, a DATA packet, and a ACK packet, and τ is maximum propagation delay between two nodes.

Second, we show the probability of transmission success state. In reality, there is a possibility of collision of DATA packets due to the transmission of hidden nodes [12]. To simplify this analysis, assume that the DATA packet was successfully transmitted when the RTS packet was successfully transmitted. In other words, in order for a tag node to transmit an RTS packet normally, it is assumed that the neighbor node of the tag node does not transmit in the same time slot, and that the hidden node of the tag node does not transmit in the η_{RTS} period, where η_{RTS} is the period from when the tag node sends an RTS packet until the destination node starts sending a CTS packet. Therefore, it can be expressed as Eq. (6), and the probability that the tag node exists in the transmission success state is shown in Eq. (7)

$$\eta_{RTS} = \left\lceil \frac{T_{RTS} + SIFS}{t_{slt}} \right\rceil \tag{6}$$

$$P_{tag\_suc} = p(1-p)^n (1-p)^{2\eta_{RTS} H(r)} \tag{7}$$

where $H(r)$ is the number of hidden nodes and can be expressed as

$$H(r) = \theta \left\{ \pi R_{cs}^2 - 2R_{cs}^2 \left(\cos^{-1} \frac{r}{2R_{cs}} - \frac{r}{2R_{cs}} \sqrt{1 - \left(\frac{r}{2R_{cs}} \right)^2} \right) \right\} \tag{8}$$

where θ, r and R_{cs} are the density of nodes, the distance between the transmitter and the receiver, the radius of the transmission range, respectively. Also, the time T_{tag_suc} until successful transmission is equal to T_{nbr_snd}.

Finally, we consider the collision state. Let P_{tag_col} be the probability that a tag node exists in a collision state. A collision state is when a tag node sends an RTS packet, and the same time, a collision occurs when a tag node's neighbor node or hidden node sends a packet. The probability at that time can be expressed as follows.

$$\begin{aligned} P_{tag\_col} &= p\{1-(1-p)^n\} \\ &\quad + p(1-p)^n \left\{1-(1-p)^{2\eta_{RTS} H(r)}\right\} \\ &= p - p(1-p)^n (1-p)^{2\eta_{RTS} H(r)} \end{aligned} \tag{9}$$

Also, the time T_{tag_col} until the collision is equal to T_{nbr_col}.

From the above, normalized throughput ρ per node can be obtained by using the probabilities of each state of neighbor nodes, the transmission success state, and the collision state. The equation is expressed as

$$\rho = \frac{T_{tag\_suc} P_{tag\_suc}}{E} \tag{10}$$

where E is an expected value and can be expressed as Eq. (11)

$$\begin{aligned} E &= T_{nbr\_idl} P_{nbr\_idl} + T_{nbr\_snd} P_{nbr\_snd} + T_{nbr\_col} P_{nbr\_col} \\ &\quad + T_{tag\_suc} P_{tag\_suc} + T_{tag\_col} P_{tag\_col} \end{aligned} \tag{11}$$

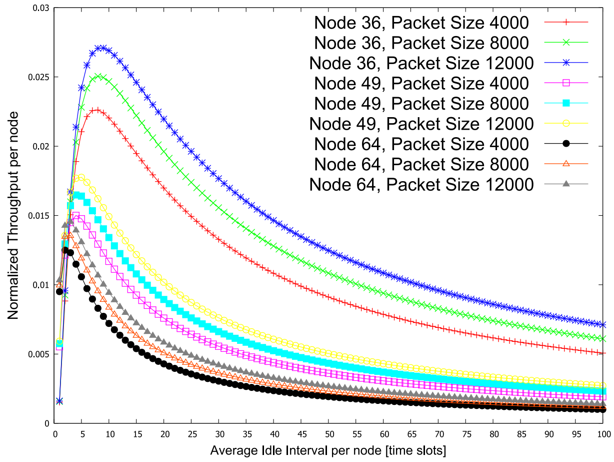

Fig. 1 Normalized throughput against average idle intervals.

2.2 Setting CW

CW has a significant effect on throughput. High-throughput transmission can be realized by setting an optimal CW according to the number of acquired neighbor nodes. This subsection describes how to calculate CW for high throughput. An optimal CW can be calculated from the relationship between the average idle slot interval and the throughput per node. Let L_{idl} be the average idle slot interval and the transmitting and receiving probability p is expressed in Eq. (12).

$$L_{idl} = \frac{P_{nbr\_idl}}{1 - P_{nbr\_idl}} \tag{12}$$

Also, by using Eqs. (10) and (12), we can express the relationship of throughput per node to the average idle slot interval per node L_{idl}/n, which is shown in **Fig. 1**.

The throughput is assumed to be IEEE 802.11. Equations (7) and (9) relate to the transmission/reception range. Equations (7) and (9) described in Section 2.1 relate to the distance r between the sending node and the receiving node. Figure 1 was created under the following assumptions.

- Numbers N of all nodes are arranged on A plane of 100 m in length and width so as to be $\sqrt{N} \times \sqrt{N}$ at equal intervals.
- The transmission node transmits to the farthest node in the transmission range.
- The transmission range is d.

The topology of the wireless network based on the above assumptions is shown in **Fig. 2**.

From Fig. 1, the graph lines are all similar, and the throughput per node increases when the average idle slot interval per node increases, but as soon as the maximum value is reached, the throughput is decreasing. From this, the average idle slot interval per node with the maximum throughput is different depending on the packet size and the number of neighbor nodes, but it can be regarded as insignificant. If γ is the value of the average idle slot interval per node at which throughput is maximized, it can be expressed as

$$\frac{L_{idl\_opt}}{n} = \gamma \tag{13}$$

where L_{idl_opt} is the optimal L_{idl}. Also, from Eq. (12), the optimal P_{nbr_idl} is Eq. (14), and the optimal p from Eqs. (13) and (14) is Eq. (15).

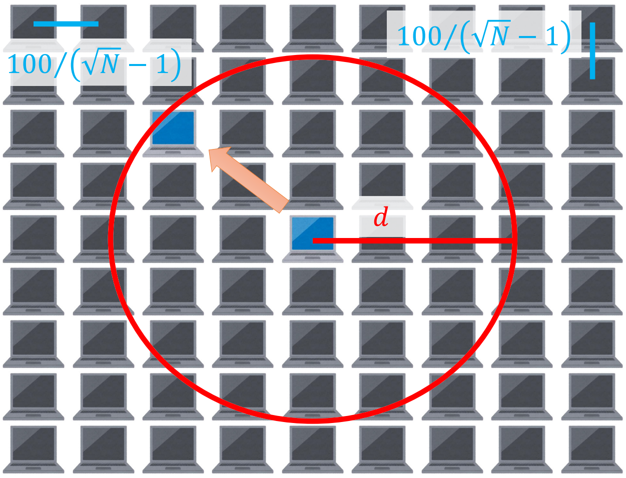

Fig. 2 Diagram of a wireless network based on assumptions.

$$P_{nbr\_idl\_opt} = \frac{L_{idl\_opt}}{L_{idl\_opt}+1} = \frac{n\gamma}{n\gamma+1} \tag{14}$$

$$p_{opt} = 1 - \sqrt[n]{P_{nbr\_idl\_opt}} = 1 - \sqrt[n]{\frac{n\gamma}{n\gamma+1}} \tag{15}$$

$P_{nbr\_idl\_opt}$ and p_{opt} are the optimal P_{nbr_idl} and the optimal transmitting and receiving probability, respectively.

Next, the calculation method of CW is described. First, let p be $P_{RTS} + P_{CTS}$, where P_{RTS} is the probability that the tag node will send an RTS packet, and P_{CTS} is the probability that the tag node will send a CTS packet. The tag node sends an RTS packet to the destination node, and when receiving a CTS packet from the destination node, the tag node starts sending data. Therefore, if we assume that the value of P_{RTS} is equal to P_{CTS}, then the value of P_{RTS} is equal to $p/2$. Since tag nodes transmit RTS packets when the neighbor nodes are also in an idle state, P_{RTS} can be expressed as

$$P_{RTS} = \frac{1}{\frac{CW_{opt}}{2}+1} P_{nbr\_idl\_opt} \tag{16}$$

where CW_{opt} is the CW at the maximum throughput. From the above, CW_{opt} can be expressed as

$$CW_{opt} = \frac{4P_{nbr\_idl\_opt}}{p_{opt}} - 2 \tag{17}$$

In ITMN, CW_{opt} can be obtained by calculating Eq. (17) using Eqs. (14) and (15), and the calculated CW is set in the tag node. The purpose of this section is to adjust the network for high-throughput wireless transmission, and this can be achieved by using Eq. (17). ITMN updates the CW every time each node sends 7 times.

2.3 How to Acquire the Number of Neighbor Nodes

The number of neighbor nodes can be acquired by intercepting RTS/CTS packets in the idle state. The source of received RTS/CTS packets is listed and managed. The flow until acquiring the number of neighbor nodes is shown below.

(i) Intercept RTS /CTS packets
(ii) Check the source of the intercepted packet
(iii) Check if the source is duplicated in the list
(iv) Return to (i) if it is a duplicate

Table 1 Network configuration.

Parameter	Value
Data Rate	11 Mbps
Buffer Size	256,000 bits
Slot Time	20 μs
SIFS	10 μs
EIFS	364 μs
DIFS	50 μs
Packet Size	4,000, 8,000, 12,000 bits
Inter-Arrival Time	0.01 s
Transmit Power	0.013 W
Max Number of Retransmissions	7
Radius of Transmission Range (d)	About 43 m

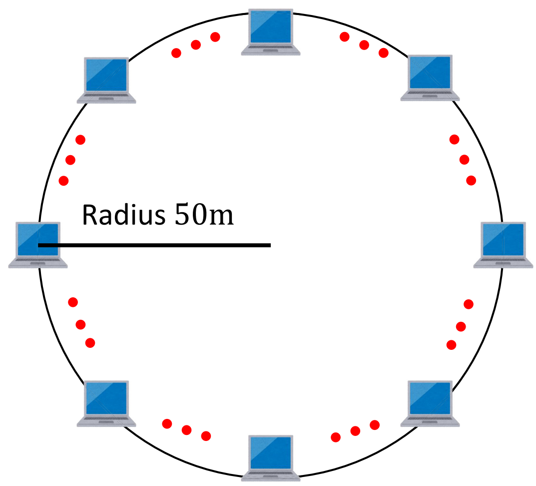

Fig. 3 Circular topology of a wireless network.

(v) Add it to the list
(vi) Increment the number of neighbor nodes and return to (i)

However, since the number of neighbor nodes swells in a network where nodes join or leave is remarkable, it is necessary to clear the contents of the list at a certain timing. In the simulation of this study, the list was initialized after calculating the optimal CW.

3. Performance Evaluation

In this section, we simulate various protocols in two different network topologies and evaluate them based on the results. The evaluation of the simulations for a circular topology are shown in Section 3.1, and the evaluation of the simulations for a square-shaped topology are shown in Section 3.2.

3.1 Simulation in the Case of Circular Network Topology

3.1.1 Simulation Environment

This subsection describes the simulation environment in the circular topology. The simulation software used is Riverbed Modeler 18.5 [13]. As a simulation result, IEEE 802.11 DCF and ITMN are compared and evaluated. The simulation parameters are shown in **Table 1**.

CW of 802.11 DCF has a maximum value 1,023. On the other hand, theoretically, there is no upper limit for CW in our proposal ITMN. So, the calculated CW value is used as it is. As a wireless network environment, nodes are arranged at equal intervals on a circle with a radius of 50 m. An image of the topology is shown in **Fig. 3**. As shown in Fig. 3, we place nodes evenly spaced on the circumference of a 50 m radius. The reason for running the simulation with a different topology from the theoretical analy-

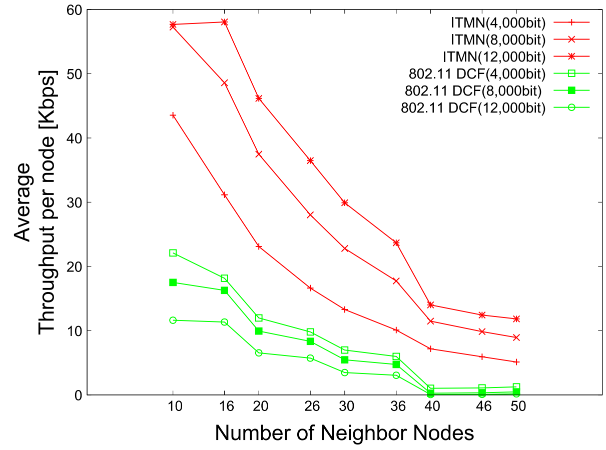

Fig. 4 Throughput against neighbor nodes.

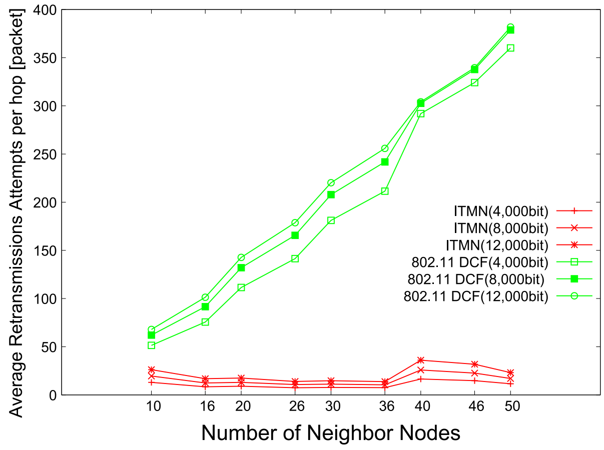

Fig. 5 Retransmission attempts against neighbor nodes.

sis is that we wanted the number of neighbor nodes acquired by any given node to be the same. Since the environment in which any given node transmits and receives is the same, it is easier to identify errors in node transmission and reception. This time, we increased the number of neighbor nodes even more than in our previous paper [14], and simulated the topology as shown in Fig. 3.

All nodes generate traffic according to the Poisson distribution at the same packet arrival. The packet arrival interval is set to a value that saturates the network. The packet size is 4,000 bits, 8,000 bits, 12,000 bits, and the MAC header is not included. The parameter γ in Eq. (13) is around about 5 when throughputs rise to peaks in Fig. 1.

3.1.2 Throughput

In this subsection, we evaluate ITMN from the simulation results. Throughput includes only the packet size that was successfully received and does not include other packets. First, the result is shown in **Fig. 4**.

In Fig. 4, the vertical axis is the average throughput per node, and the horizontal axis is the number of neighbor nodes. As the number of neighbor nodes increases, the average throughput per node decreases, which is thought to be due to the increase in hidden nodes and neighbor nodes. It can be seen that ITMN always has higher throughput than 802.11 in any number of neighbor nodes. Moreover, throughput of ITMN has higher than the one of 802.11 at any packet size.

3.1.3 Retransmission Attempts

In this subsection, **Fig. 5** shows the simulation results, and we compare retransmission attempts per hop with respect to the number of neighbor nodes of IEEE 802.11 DCF and ITMN. The retransmission attempt per hop is the total number of retransmissions performed by a tag node and neighbor nodes that exist within 43 m of the tag node transmission/reception range. In IEEE 802.11 DCF, retransmission attempts increase as the number of neighbor nodes increases. This is due to the hidden node problem with the increase in the number of neighbor nodes, and collisions are likely to occur. On the other hand, the retransmission attempts are reduced in ITMN. This is because the optimal CW is set for each number of neighbor nodes and the number of extra transmissions is reduced. Therefore, it can be seen that ITMN has significantly fewer retransmission attempts than 802.11. This reduces the waste of energy required for wireless networks.

3.1.4 Fairness

In this subsection, IEEE 802.11 DCF and ITMN are evaluated from the viewpoint of fairness. In order to evaluate fairness, we adopt a modified Fairness Index (FI) [15] that is generally used. FI can be expressed as

$$FI = \frac{\left(\sum_{i=1}^{n} T_i/\phi_i\right)^2}{n\sum_{i=1}^{n}(T_i/\phi_i)^2} \tag{18}$$

where T_i is throughput of flow i, ϕ_i is the weight of flow i (normalized throughput requested by each node). Here, $\phi_1 = \phi_2 = \cdots = \phi_n$ because the simulation assumes that all nodes have the same weight. So, if the weight of flow is ϕ, then Eq. (19) is obtained.

$$\begin{aligned} FI &= \frac{\left(\sum_{i=1}^{n} T_i/\phi\right)^2}{n\sum_{i=1}^{n}(T_i/\phi)^2} \\ &= \frac{\left(\sum_{i=1}^{n} T_i\right)^2/\phi^2}{n\sum_{i=1}^{n}T_i^2/\phi^2} \\ &= \frac{\left(\sum_{i=1}^{n} T_i\right)^2}{n\sum_{i=1}^{n}T_i^2} \end{aligned} \tag{19}$$

According to Eq. (19), $FI \leq 1$, and the equality is true when $T_1 = T_2 = \cdots = T_n$. Usually, the higher the FI, the higher the fairness.

A graph of fairness between ITMN and IEEE 802.11 DCF is shown in **Fig. 6**. The horizontal axis is the number of neighbor nodes, and the vertical axis is the fairness index. In 802.11, fairness decreases with the number of neighbor nodes. In addition, after 40 nodes, fairness is greatly reduced and there are many nodes with 0 bps (value of throughput). On the other hand, ITMN has a slight change, but not a big difference. Similarly, in the packet size, the fairness decreases in 802.11, whereas ITMN does not change significantly. Furthermore, in ITMN, the fairness exceeded 0.9 for any number of neighbor nodes and packet sizes.

From the above results, it can be seen that ITMN is fairer and does not change significantly due to the number of neighbor nodes or the packet size.

3.1.5 Delay

In this subsection, we show the delay time of IEEE 802.11

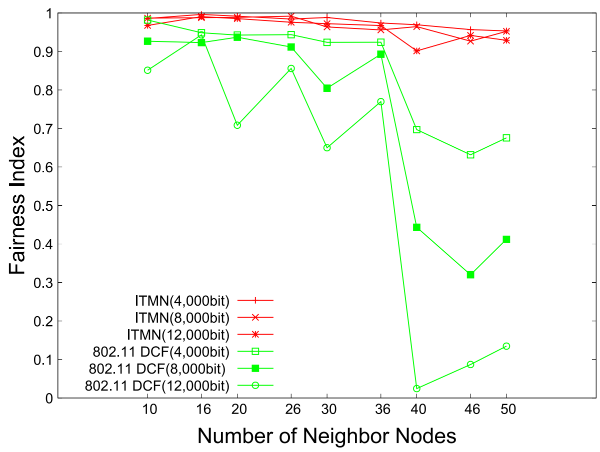

Fig. 6 Fairness Index.

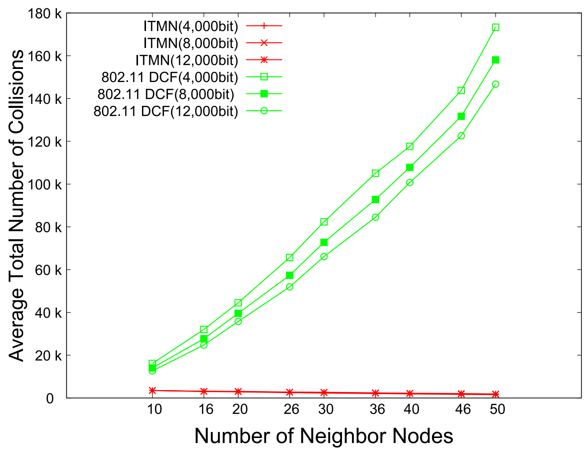

Fig. 8 Average total number of collisions.

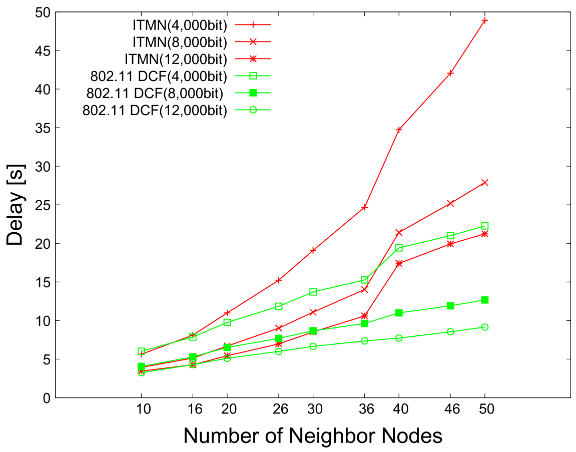

Fig. 7 Delay.

DCF and ITMN. Each delay time is shown in **Fig. 7**. The delay time usually tends to be higher as the throughput is increased. The reason for this is that all the nodes are competing with each other and it takes time to ensure that they can communicate. Also, if a node discards a packet after multiple failed transmissions, the time associated with that communication is not included in the calculation of the delay time. In other words, the delay time is small in this case because it is only counted when the transmission is successful.

According to this graph, the delay time of ITMN is longer than that of 802.11 DCF. The graph shows that the delay time of ITMN increases as the number of neighbor nodes increases and that of 802.11 DCF increases slightly. However, due to the relationship between delay time and throughput, the increase in delay time of ITMN is within an acceptable range. On the other hand, 802.11 has a numerically smaller delay because of repeated transmission failures and many packets are discarded. However, in the previous discussion, it can be said that the performance of 802.11 is poor because the throughput is very low.

3.1.6 Collisions

In this section, we compare the total number of collisions of each node in IEEE 802.11 DCF and ITMN from simulation results. First, the results are shown in **Fig. 8**.

According to the graph, the difference in the number of collisions is obvious: the number of collisions for 802.11 DCF increases significantly as the number of neighbor nodes increases, while the number of collisions for ITMN is very small regardless of the number of neighbor nodes. By minimizing the number of useless transmissions, we succeeded in reducing the number of collisions among nodes.

3.2 Simulation in the Case of Square-shaped Topology

In Section 3.1, we evaluated the ITMN by simulating it in a circular network topology. In this section, we evaluate ITMN by simulating them in a more realistic environment. In this case, we can find different behavior for the nodes in different places such as the center and corner.

3.2.1 Simulation Environment

This subsection describes the simulation environment in the Square-shaped topology. The simulation software and parameters are basically the same as in Section 3.1, Section 3.1.1. In this section, we evaluate the ITMN by simulating it together with IEEE 802.11 DCF and OBEM [7]. Detailed network configuration is given in Table 1 in Section 3.1.1.

In the conventional method IEEE 802.11 DCF, there is a maximum *CW* value, but in ITMN and OBEM there is no upper limit for *CW*, so the calculated *CW* value is set as it is.

We assume that the topology of a network in simulation, is same as that used in theoretical analysis, as shown in Fig. 2. The communication distance d is about 43 m and the number of nodes N is 100. We focus on two nodes. The node with the highest node density is called the D (densest) node, and the node with the lowest node density is called the S (sparsest) node. The D node is located at the center and the S node is located at the corner.

All nodes generate traffic according to the Poisson distribution at the same packet arrival. The packet arrival interval is set to a value that saturates the network. The packet size is 4,000 bits, 8,000 bits, 12,000 bits, and the MAC header is not included.

3.2.2 Throughput

In this subsection, we evaluate ITMN from the simulation results. Throughput includes only the packet size that was successfully received and does not include other packets. First, the result is shown in **Fig. 9**.

In Fig. 9, the vertical axis is the average throughput per node, and the horizontal axis is the packet size. In all communication methods, the throughput increases as the packet size increases. The throughput of ITMN is higher than that of 802.11 DCF.

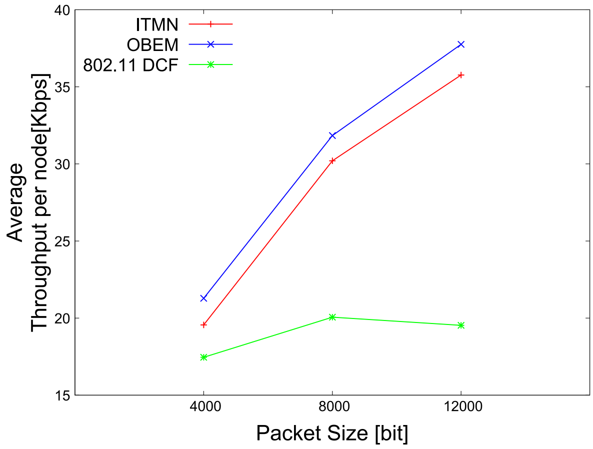

Fig. 9 Throughput against neighbor nodes.

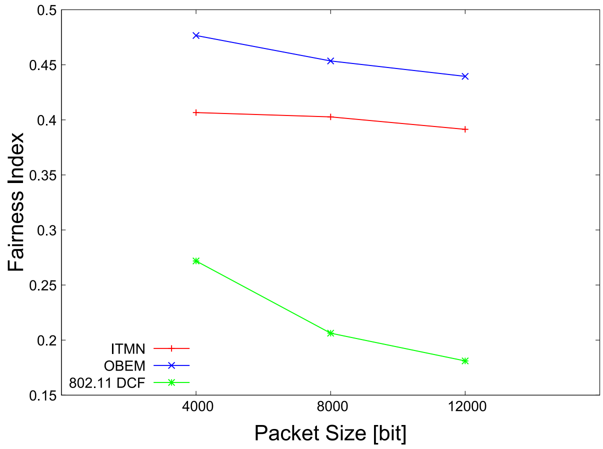

Fig. 11 Fairness Index at D node.

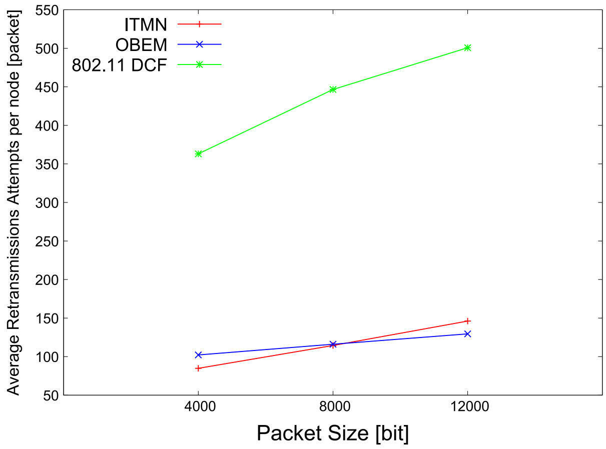

Fig. 10 Retransmission attempts against neighbor nodes.

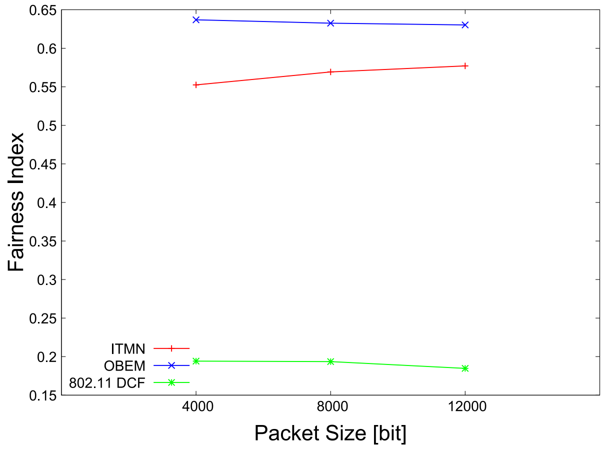

Fig. 12 Fairness Index at S node.

However, the throughput of ITMN was slightly lower than that of OBEM. After analyzing related simulation results, we found, in Fig. 14, the number of collisions of ITMN is much lower than that of OBEM, which means that if increasing the sending probability in ITMN, we can obtain a higher throughput than OBEM.

3.2.3 Retransmission Attempts

In this subsection, **Fig. 10** shows the simulation results, and we compare average retransmission attempts per node of ITMN, 802.11 DCF, and OBEM. The number of retransmission attempts of ITMN was much lower than that of 802.11 DCF and slightly lower than that of OBEM. Since 802.11 DCF is not a communication method that dynamically adapts to any given network, it is inevitably prone to collisions. On the other hand, ITMN is dynamically adaptive to arbitrary networks, so it can reduce the number of extra transmissions. This reduces the waste of energy required for wireless networks.

3.2.4 Fairness

In this subsection, ITMN, OBEM, and 802.11 DCF are evaluated from the viewpoint of fairness. The calculation to derive the fairness is done using Eq. (19).

A graph of fairness among ITMN, OBEM, and 802.11 DCF is shown in **Figs. 11** and **12**. The horizontal axis is the packet size, and the vertical axis is the fairness index. In order to investigate the fairness of nodes in different places, we chose two nodes, D node and S node. Regardless of the communication method, the fairness of S nodes is higher than that of D nodes. This can be attributed to the fact that there are fewer nodes in the vicinity and thus less interference among them. The fairness of ITMN was higher than that of 802.11 DCF regardless of nodes, but slightly lower than that of OBEM. However, the fairness of ITMN is comparable to that of OBEM. In addition, ITMNs maintain a certain degree of impartiality in an environment that is similar to reality and is poor for wireless communication.

3.2.5 Delay

In this subsection, we show the delay time of ITMN, OBEM, and 802.11 DCF. Each delay time is shown in **Fig. 13**. The delay time is the same specification as in Section 3.1.5. Throughput and delay time are generally proportional. The reason for this is that all the nodes are competing with each other, and it takes time to ensure that they can communicate. Also, if a node discards a packet after multiple failed transmissions, the time associated with that communication is not included in the calculation of the delay time. In other words, the delay time is small in this case because it is only counted when the transmission is successful.

The vertical axis is the delay time [s], and the horizontal axis is the packet size [bit]. According to this graph, the delay time of ITMN is larger than that of 802.11 DCF by about 4 seconds, but it is almost the same as that of OBEM. In this graph, the delay time decreases as the packet size increases. On the other hand, 802.11 has a numerically smaller delay because of repeated transmission failures and many packets are discarded. However, in the previous discussion, it can be said that the performance of 802.11 is

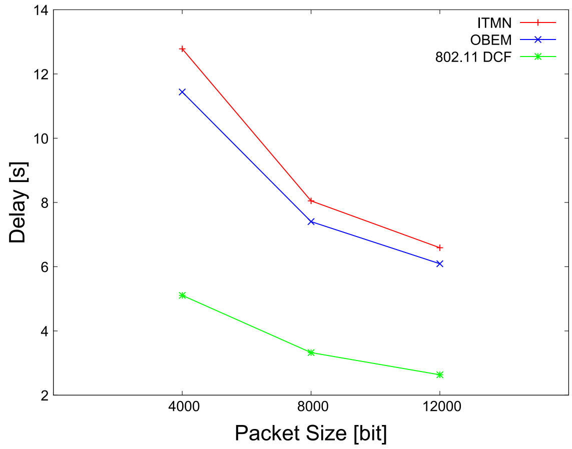

Fig. 13 Delay.

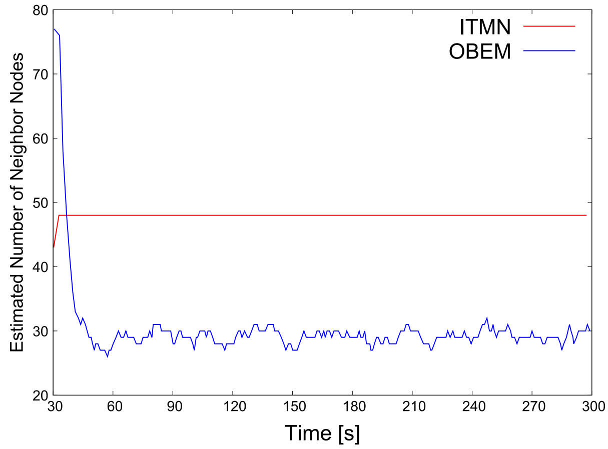

Fig. 15 Estimated number of neighbor nodes at D node.

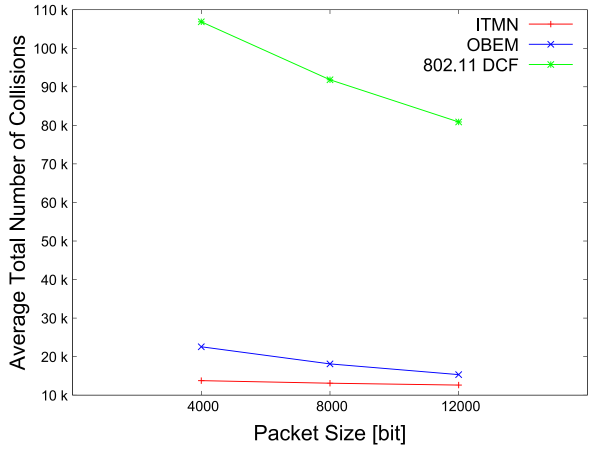

Fig. 14 Average total number of collisions.

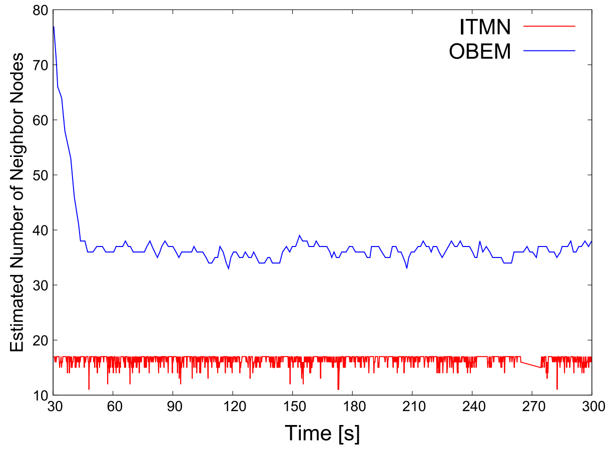

Fig. 16 Estimated number of neighbor nodes at S node.

poor because the throughput is very low. The delay time of the ITMN is almost equal to that of the OBEM, so the ITMN is not degraded compared to the OBEM.

3.2.6 Collisions

In this subsection, we compare the total number of collisions of each node between ITMN, OBEM, and 802.11 DCF from simulation results. First, the results are shown in **Fig. 14**.

The vertical axis is the number of collisions, and the horizontal axis is the packet size [bit]. The graph clearly shows the difference in the number of collisions. The number of collisions of ITMN is very much lower than that of 802.11 DCF and about 3,000 to 10,000 lower than that of OBEM. By minimizing the number of useless transmissions, we succeeded in reducing the number of collisions among nodes.

3.2.7 Estimation Accuracy of the Number of Neighbor Nodes

In this subsection, we compare the estimates of neighbor nodes and show their accuracy. 802.11 DCF does not estimate the number of neighbor nodes, so we compare only ITMN and OBEM. The number of neighbor nodes of D node is 48, and that of S node is 17. In our simulation, we assume that the packet size is 4,000 bit. The reason is that the estimation accuracy does not change regardless of the packet size. **Figures 15** and **16** show the actual number of neighbor nodes shown by ITMN and OBEM.

The vertical axis is the estimated number of neighbor nodes, and the horizontal axis is the simulated time [s]. According to the graph, the line representing the number of nodes for D nodes is smooth, while the line for S nodes is rugged. Normally, when a node is idle, it observes the surrounding nodes. The longer the idle time, the longer the observation time, and the number of neighbor nodes is updated using the obtained data. In other words, the longer the observation time, the more stable and smooth the line of the graph becomes. The number of ITMN neighbor nodes in D node is almost 48. In addition, the number of OBEM neighbor nodes at D node is always more than 10 smaller than the actual number. Next, we focus on the number of neighbor nodes at S node. The number of neighbor nodes for ITMN is 17, but the ones for OBEM is very large. These solve the problem of inaccurate estimation of the number of neighbor nodes in a dense node environment.

3.2.8 Changes in CW

In this subsection, we compare the change of CW between ITMN and OBEM. the stability of *CW* means the stability of the network. We measured *CW* at two different nodes, D node and S node, and we showed in **Figs. 17** and **18**. As in Section 3.2.7, the simulation was performed with a packet size of 4,000 bit.

The vertical axis is the Contention Window [Kslot], and the horizontal axis is the simulated time [s]. Both ITMN and OBEM calculate CW using the number of neighbor nodes. In other words, if the line of the graph of the number of neighbor nodes is rugged, the line of the graph of CW is also rugged. Both ITMN and OBEM calculate CW using the number of adjacent nodes.

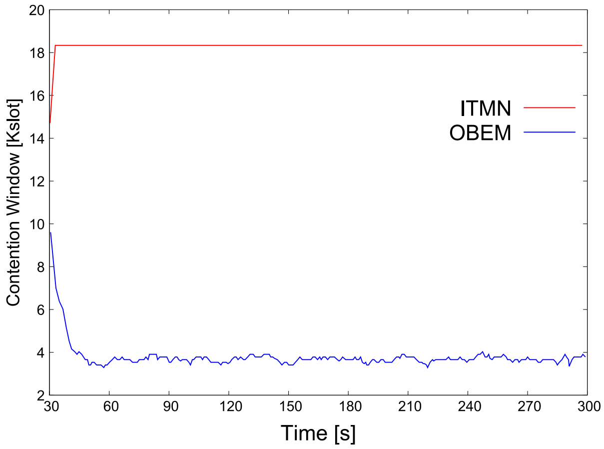

Fig. 17 Changes in CW at D node.

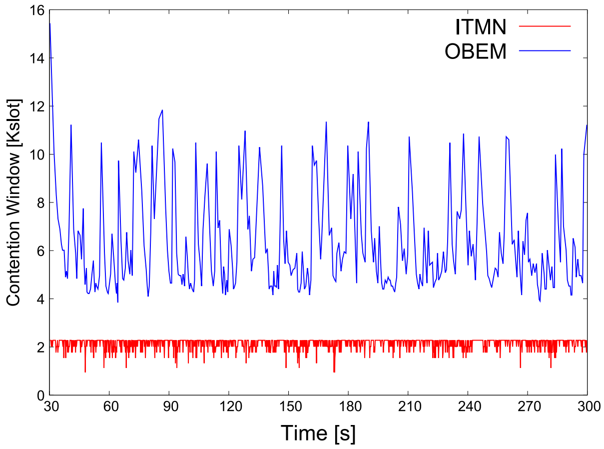

Fig. 18 Changes in CW at S node.

In other words, if the line of the graph of the number of neighbor nodes is rugged, the line of the graph of CW is also rugged. However, the smaller the amplitude of the graph line, the better. According to this graph, ITMN does not have much change in CW regardless of node type. On the other hand, for OBEM, CW of D node is slightly up and down, and that of S node is significantly up and down.

In other words, we can say that the change of CW is constant for ITMN regardless of the density of nodes.

4. Conclusion

In this paper, we proposed a novel protocol ITMN, where a tag node intercepts RTS/CTS packets to acquire the number of neighbor nodes and adjusts network parameters to be optimal values. As a result, we were able to achieve high throughput and low collision wireless transmission by setting the optimal CW according to the number of neighbor nodes acquired by the tag node.

In order to evaluate the proposed ITMN, we carried out simulations in the cases of two different network topologies. In the circular topology, every node has the same number of neighbor nodes. On the other hand, in the square-shaped topology, every node has the different number of neighbor nodes, which is similar to the realistic environment. In the square-shaped topology, nodes near the boundary have a smaller number of neighbor nodes and are beneficial to obtain transmission opportunities with a lower average CW, so the fairness of the square-shaped topology became lower than that of the circular topology.

Theoretical analysis and simulation results show that ITMN is effective and can dynamically adapt to changes in the network. Furthermore, ITMN mitigates the hidden node problem, achieves higher communication performance than IEEE 802.11 DCF, and solves the problem of inaccurate estimation of the number of neighbor nodes in OBEM while maintaining the same performance as OBEM. In future work, we need to evaluate the effectiveness of ITMN by verifying them in real environments.

Acknowledgments This work was supported by the Nitto Foundation.

References

[1] Li, J., Zeng, X. and Su, Q.: Performance Investigation of Backoff Algorithms in Multihop Wireless Networks, *2008 9th International Conference for Young Computer Scientists*, pp.564–569 (online), DOI: 10.1109/ICYCS.2008.289 (2008).

[2] Du, S., Du, X. and Liu, K.: An efficient split-channel MAC protocol for multihop wireless networks, *9th International Conference on Communications and Networking in China*, pp.357–360 (online), DOI: 10.1109/CHINACOM.2014.7054317 (2014).

[3] Shahin, N., Ali, R., Kim, S.W. and Kim, Y.-T.: Cognitive backoff mechanism for IEEE802.11ax high-efficiency WLANs, *Journal of Communications and Networks*, Vol.21, No.2, pp.158–167 (online), DOI: 10.1109/JCN.2019.000022 (2019).

[4] Zhou, X., Zheng, C. and Liao, M.: Full-feedback contention window adaption for IEEE 802.11 WLANs, *Journal of Systems Engineering and Electronics*, Vol.27, No.1, pp.90–98 (2016).

[5] Wu, G. and Xu, P.: Improving Performance by a Dynamic Adaptive Success-Collision Backoff Algorithm for Contention-Based Vehicular Network, *IEEE Access*, Vol.6, pp.2496–2505 (online), DOI: 10.1109/ACCESS.2017.2783909 (2018).

[6] Lei, J., Tao, J., Huang, J. and Xia, Y.: A Differentiated Reservation MAC Protocol for Achieving Fairness and Efficiency in Multi-Rate IEEE 802.11 WLANs, *IEEE Access*, Vol.7, pp.12133–12145 (online), DOI: 10.1109/ACCESS.2019.2892760 (2019).

[7] Sanada, T., Tian, X., Okuda, T. and Horie, R.: A Novel MAC Protocol in Multi-hop Wireless Networks Through Dynamically Optimizing Backoff, *10th EAI International Conference on Wireless and Satellite Systems* (2019).

[8] Abdullah, A.A., Gebali, F. and Cai, L.: Modeling the throughput and delay in wireless multihop ad hoc networks, *Proc. 28th IEEE Conference on Global Telecommunications*, pp.5711–5716 (online), DOI: 10.5555/1811982.1812327 (2009).

[9] Li, P. and Fang, Y.: Saturation throughput of IEEE 802.11 DCF in multi-hop ad hoc networks, *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pp.1–7 (online), DOI: 10.1109/MILCOM.2008.4753624 (2008).

[10] Gao, Y., Chiu, D.-M. and Lui, J.C.: Determining the end-to-end throughput capacity in multi-hop networks: Methodology and applications, *Proc. Joint International Conference on Measurement and Modeling of Computer Systems*, pp.39–50 (online), DOI: 10.1145/1140103.1140284 (2006).

[11] Wang, Y., Yan, N. and Li, T.: Throughput Analysis of IEEE 802.11 in Multi-Hop Ad Hoc Networks, *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pp.1–4 (online), DOI: 10.1109/WiCOM.2006.239 (2006).

[12] Xu, K., Gerla, M. and Bae, S.: How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks, *Global Telecommunications Conference, GLOBECOM '02*, Vol.1, pp.72–76, IEEE (online), DOI: 10.1109/GLOCOM.2002.1188044 (2002).

[13] Riverbed: Riverbed Modeler, Riverbed (online), available from ⟨https://www.riverbed.com/jp/products/steelcentral/steelcentral-riverbed-modeler.html⟩ (accessed 2020-07-30).

[14] Morita, T., Tian, X. and Okuda, T.: ITMN: A New MAC Protocol for Improving Throughput of Multihop Wireless Networks Considering Number of Neighbor Nodes, *Advanced Information Networking and Applications*, pp.153–163 (online), DOI: 10.1007/978-3-030-44041-1_14 (2020).

[15] Chiu, D.-M. and Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks, *Computer Networks and ISDN Systems*, Vol.17, pp.1–4 (online), DOI: 10.1016/0169-7552(89)90019-6 (1989).

Taiki Morita was born in 1997. He received his B.S. School of Information Science and Technology from Aichi Prefectural University in 2020. He is in the master's program in the Graduate School of Information Science and Technology, Aichi Prefectural University. His research interests include medium access control and QoS in wireless networks.

Xuejun Tian graduated from Hebei University, China in 1985. He received his M.S. degree from the Department of Electrical and Mechanical Engineering, Tianjin Institute of Technology, China in 1991, and his Ph.D. degree from the Department of Intelligence and Computer Science, Nagoya Institute of Technology, Japan, in 1998, respectively. Since 1998, he has been an Assistant Professor and Associate Professor from 2008 in the Department of Information Systems, Faculty of Information Science and Technology, Aichi Prefectural University, Japan. From July 2003 to June 2004, he was a Visiting Assistant Professor in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville, FL. Dr. Tian is a member of IEICE, IEEJ, IPSJ respectively. His research interests include QoS, wireless networks, mobile communications and ubiquitous computing.

Takashi Okuda received his B.S., M.S., and Ph.D. degrees in Engineering from Toyohashi University of Technology, Japan, in 1985, 1987, and 1992, respectively. He is a professor in the School of Information Science and Technology, Aichi Prefectural University, Nagakute, Japan. He joined Toyohashi University of Technology as a research associate and Asahi University as an Associate Professor in 1988 and 1993, respectively. He worked as a Visiting Professor at Information Systems and Technologies Department, Weber State University, UT, from December 1994 to August 1995 and as a Visiting Scholar at the Computer Engineering Department, Duke University, NC, from July 2002 to January 2003. His research interests include traffic engineering over communication networks, and information systems and technologies for business and society. He is a member of IEEE, IEICE, IPSJ, ORSJ, and JSET.