

## PAPER

# A Fixed Backoff-Time Switching Method for CSMA/CA Protocol in Wireless Mesh Networks

Sritrusta SUKARIDHOTO<sup>†</sup>, *Student Member*, Nobuo FUNABIKI<sup>†a)</sup>, Toru NAKANISHI<sup>†</sup>, Kan WATANABE<sup>†</sup>, and Shigeto TAJIMA<sup>††</sup>, *Members*

**SUMMARY** As a flexible and cost-efficient scalable Internet access network, we studied architectures, protocols, and design optimizations of the *Wireless Internet-access Mesh NETWORK (WIMNET)*. WIMNET is composed of multiple access points (APs) connected through multihop wireless communications on *IEEE 802.11 standards*. The increasing popularity of real-time applications such as IP-phones and IP-TV means that they should be supported in WIMNET. However, the contention resolution mechanism using a random backoff-time in the *CSMA/CA protocol* of 802.11 standards is not sufficient for handling real-time traffic in multihop wireless communications. In this paper, we propose a *Fixed Backoff-time Switching (FBS) method* for the CSMA/CA protocol to improve the real-time traffic performance in WIMNET by giving the necessary activation chances to each link. We implement our proposal on the *QualNet* simulator, and verify its effectiveness through simulations on three network topologies with four scenarios.

**key words:** wireless mesh network, CSMA/CA protocol, fixed backoff-time switching (FBS), real-time application, QualNet

## 1. Introduction

The *wireless mesh network* has recently been studied as a promising network technology for a flexible and cost-efficient solution to expand the communication service area by distributing wireless mesh routers on a network field [1]–[3]. The mesh routers are connected with each other through multihop wireless communication links using *IEEE 802.11 standards*, in addition to wireless links between client hosts and routers. Then, as a scalable Internet access network based on this technology, we studied architectures, protocols, and design optimizations of the *Wireless Internet access Mesh NETWORK (WIMNET)* [3]. For a simple architecture, WIMNET is composed of only *access points (APs)* as mesh routers as shown in Fig. 1. At least one AP acts as a *Gateway (GW)* to the Internet. Any host in WIMNET can be connected to the Internet through multihop communications between APs and the GW after establishing an association with one neighbor AP.

Like conventional wireless networks, WIMNET adopts the *CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol* of the *IEEE802.11 MAC (Media Access*

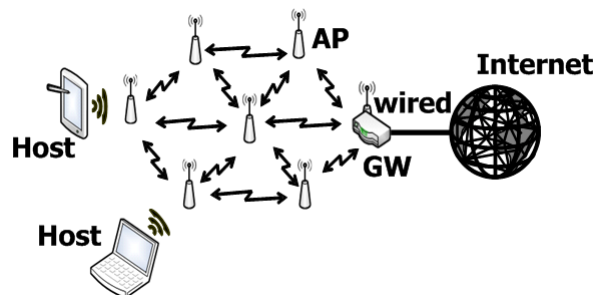


Fig. 1 Outline of WIMNET.

*Control*) for the shared communication media access by resolving contentions among interfered wireless links [4]. In the CSMA/CA protocol, any wireless node holding a transmission packet is on standby for a random time called the *backoff-time* before starting the data frame transmission, to avoid frame collisions among contending nodes while providing their fairness. At each transmission chance, a random value within a size called the *Contention Window (CW)* is selected for the backoff-time. When a node fails at a transmission, the CW size is doubled to reduce the probability of the collision occurrence in the retransmission, which is called the *binary exponential backoff*. When the node succeeds at a transmission, it resets the CW size to the initial one.

Unfortunately, this conventional CSMA/CA protocol is not sufficient for multihop communications in WIMNET. Firstly, heavy congestions of links around the GW can be bottlenecks for all WIMNET communications, because these links have to handle a lot of packets to/from the GW for the Internet access. Thus, they should be activated with much higher priorities than other links. Secondly, interferences among these congested links may not be resolved by a random backoff-time in the CSMA/CA protocol because of the limited CW size. Here, we note that the initial CW size is small, and even the maximum CW size is limited. Then, multiple conflicting links can be activated simultaneously by generating the same or similar backoff-times at their transmitting nodes. As a result, no link can complete packet transmission successfully, and needs a retransmission that may cause further conflicts. Hence, using the conventional CSMA/CA protocol, WIMNET can cause a lot of packet losses and intolerable delays, which cannot afford real-time applications such as IP-phones and IP-TVs, although their popularity has been increased with the advancement of dig-

Manuscript received June 15, 2012.

Manuscript revised November 13, 2012.

<sup>†</sup>The authors are with the Graduate School of Natural Science and Technology, Okayama University, Okayama-shi, 700-8530 Japan.

<sup>††</sup>The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan.

a) E-mail: funabiki@cne.okayama-u.ac.jp

DOI: 10.1587/transcom.E96.B.1019

ital communication technologies.

In this paper, we propose a *Fixed Backoff-time Switching (FBS) method* for the CSMA/CA protocol to solve above-mentioned problems in WIMNET, as extended works in [5], [6]. In the FBS method, the two constant parameters, namely the *active backoff-time* and the *passive backoff-time*, must be assigned to every link before starting communications when the topology is changed. Here, the *link activation* means that the transmitting node of the corresponding wireless link sends out a signal for a frame transmission. The *active backoff-time* represents a shorter waiting time for the link to be activated preferentially when it holds packets for transmissions. The *passive backoff-time* represents a longer waiting time for the link to be activated only if the contending links using the active backoff-time are not activated, where a larger value than any active backoff-time is used. Besides, for any backoff-time of any link, a different value is assigned from each other to avoid simultaneous link activations as best as possible, and the magnitude follows the descending order of expected traffic loads of links so that congested links can be activated more frequently.

During communications, the *target link activation rate* and the *actual link activation rate* are calculated at every link activation chance. The former one represents the required rate of activating the corresponding link to handle its traffics properly, and is calculated from the required total bit rate of the link, the average frame size, and the transmission error rate. The latter one represents the rate of actually activating the corresponding link, and is calculated by counting the numbers of link activation chances and actually activated times for each link. If the actual link activation rate is smaller than the target activation rate, the active backoff-time is selected for the preferential activation of the link. Otherwise, the passive backoff-time is selected. Because different values are assigned them, contentions among interfered links are expected to be resolved.

In summary, in the *fixed backoff-time switching* method, the value for any backoff-time is calculated when the topology is changed, and one of the *fixed* values for *backoff-times* is selected or *switched* on for each link activation during communications.

For evaluations, we implement the FBS method on a well-known network simulator *QualNet* [10]. *QualNet* adopts a more realistic physical model than other network simulators such as *ns-2* [11]. Before implementing the proposal on hardware, evaluations on such a realistic network simulator are significant to refine the details. Using *QualNet*, we verify the effectiveness of the FBS method through simulations in three network topologies with four scenarios.

The rest of this paper is organized as follows: Sect. 2 reviews the CSMA/CA protocol in IEEE 802.11 standards and related studies. Section 3 proposes the FBS method. Section 4 describes our implementation on *QualNet*. Section 5 shows evaluations through simulations. Section 6 concludes this paper with some future works.

## 2. CSMA/CA Protocol and Related Studies

In this section, we briefly review the CSMA/CA protocol in IEEE 802.11 standards, and the related studies to this paper.

### 2.1 Overview of CSMA/CA Protocol

The CSMA/CA protocol in IEEE802.11 standards allows multiple wireless nodes to use the same physical medium or communication channel for data transmissions by detecting and/or avoiding data frame collisions among interfered links. Figure 2 illustrates the timing chart for the data frame transmission on this protocol. When a channel becomes free, a transmitting node waits for a constant *DIFS* period and a backoff-time that is randomly selected between 0 and the *CW* size. During this waiting period, if the node does not detect any transmission from other node, it starts the transmission. This random backoff-time is used to stagger the transmission starting timing among ready nodes to avoid their collisions. Nevertheless, if a collision happens, the *CW* size is doubled as the binary exponential backoff to avoid further collisions, and the transmission procedure is applied again. If a transmission succeeds, the *CW* size is reset to the initial one  $CW_{min}$ .

### 2.2 Related Studies of CSMA/CA Modifications

In [12], Xu et al. raised a question: can the IEEE 802.11 work well in wireless ad hoc networks? They concluded that the protocol was not designed for multihop networks. Although it can support some ad hoc network architecture, it is not intended to support wireless multihop networks including ad hoc networks and wireless mesh networks, because *RTS/CTS* exchanges will block all the wireless nodes in the neighborhood.

In [13], Nakamura et al. examined a fixed backoff-time for wireless local area networks. Through simulations, they showed that it can improve the throughput and delay performance by reducing collisions and idling periods. However, their method is based on the *PCF* scheme, whereas WIMNET is based on *DCF*.

In [14], Minooei et al. proposed an efficient backoff model for ad hoc networks using *DCF* that modifies the backoff-time by considering the frame collision probability of each node in multihop wireless networks. For this purpose, the backoff-time  $bt^m$  is given by:

$$bt^m = rand [CW_{min} \times 2^{m-1}, CW_{min} \times 2^m] \quad (1)$$

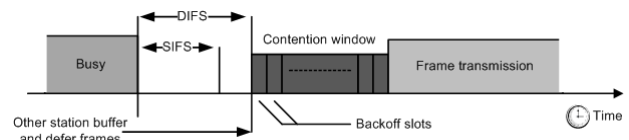


Fig. 2 Timing chart for data frame transmission.

where  $CW_{min}$  represents the initial CW size,  $m$  does the number of consecutive transmission failures (retry counter), and the function  $rand[x, y]$  returns a uniformly randomized integer between  $x$  and  $y$ . Their simulation results show that it provides higher end-to-end throughputs than the conventional one. We adopted this backoff-time model and modified it into our fixed backoff-time one, where the performance is compared in Sect. 5. However, their method does not consider the traffic feature of a wireless mesh network, where links closer to the GW need to transmit more packets than links farer from it. Thus, our FBS method incorporates a mechanism of activating each link depending on its traffic volume to be transmitted.

In [15], Wu et al. presented an improvement of the method in [14], whereas our simulation results show that it gives the worse performance than [14]. This reason may come from the longer backoff-time in [15] by adopting an additional term considering the frame collision probability of the node. The longer backoff-time can suppress collisions more, but prolongs idling time, which decreases the throughput. A proper backoff-time is important to resolve the trade-off of suppressing collisions and idling time at the same time. Thus, we did not include it in the performance comparison.

In [16], Zhu et al. proposed *T-MAC* as an enhancement of the TDMA-like IEEE802.11e MAC protocol. Through simulations, they showed that this TDMA approach with loosely synchronized clocks can give a better performance. However, the implementation of clock synchronizations among many nodes is difficult in real devices.

In [17], [18], Furukawa et al. proposed an intermittent periodic transmission method to improve the performance of a wireless mesh network by avoiding collisions between the interfered links along a routing path. To block the interference from the outside of the path, the interfered links are sustained during the intermittent periodic transmissions. However, this method can be effective in a limited situation where only one communication request of transmitting packets from a source to a destination appears in a wireless mesh network. When two or more interfered requests arrive at a network at the same time, they need to be handled sequentially. Besides, the proper transmission period is hard to be set or cannot be fixed in real networks, because conditions of wireless links are often affected by interferences from surrounding objects. Thus, their method is not suitable for realistic situations in this paper where multiple requests for real-time applications arrive at the same time.

In [19], Yamada et al. proposed the *IEEE 802.11e EDCA (Enhanced Distributed Channel Access)* based congestion control method for a wireless mesh network, where the waiting time before the backoff called *AIFS (Arbitration Inter Frame Space)* of a node is dynamically changed so that the receiving traffic and the transmitting traffic can be balanced there. Unfortunately, like [17], [18], this method can be effective in a limited situation with only one request.

In [20], [21], Kishida et al. proposed a MAC protocol to suppress collisions and improve the fairness among the

hosts associated with the same AP by assigning different initial backoff-times and the same cyclic backoff-time to them. This method assumes that every host in a wireless network is associated with one AP and has the same traffic. Thus, it cannot be directly applied to WIMNET, where hosts are associated with different APs that are performing multihop communications, and their traffics are usually different from each other.

### 3. Proposal of FBS Method

In this section, we propose the FBS method for the CSMA/CA protocol in WIMNET.

#### 3.1 Overview of FBS Method

The FBS method uses the *active backoff-time* and the *passive backoff-time* for each link, and selects either of them as a backoff-time at a frame transmission by comparing the *target link activation rate* and the *actual link activation rate*. Any backoff-time is assigned a different value from each other so that no pair of the conflicting links may be activated simultaneously. Besides, the backoff-time for a link with larger traffic is assigned a smaller value than that for a link with smaller one, so that congested links can be activated preferentially. Furthermore, any active backoff-time is assigned a smaller value than a passive one, so that links using active ones have higher priorities in activations than links using passive ones.

During communications, every time a node holding packets detects that the channel for transmissions becomes free, it updates both the target activation rate and the actual activation rate. If the actual one is smaller than the target one, it selects the active backoff-time to let the link be activated, because the current activation rate of the link is not sufficient to handle its traffic. On the other hand, if it is larger, it selects the passive backoff-time to let other links with active backoff-times be activated with higher priorities. A link with the passive backoff-time can be activated only if any conflicting link with the active backoff-time does not hold packets. The following subsections describe how to calculate the three parameters in the FBS method.

Here, we discuss the design goal of the FBS method from a different point of view. To improve the throughput of a wireless network using the CSMA/CA protocol, it is necessary to reduce both the collisions among interfered links and the idling time caused by backoff-time. However, to avoid collisions, a longer backoff-time is more effective than a shorter one. Thus, the tradeoff problem between them should be considered properly. The FBS method has been designed to solve this tradeoff problem by assigning different backoff-times to the links to avoid collisions at any retry counter, and shorter backoff-times to links with heavier traffics than those to links with lighter traffics to reduce the idling time. Because shorter backoff-times can cause more collisions, we adopt two backoff-times for each link, and switch them depending on whether the corresponding

link transmits the demanded traffic or not, which is judged by comparing two activation rates in the FBS method.

### 3.2 Target Link Activation Rate

For a wireless link  $l_{ij}$  transmitting packets from  $AP_i$  to  $AP_j$  for  $i = 1, \dots, N$  and  $j = 1, \dots, N$  where  $N$  is the number of APs, the target link activation rate  $rt_{ij}$  can be calculated by:

$$rt_{ij} = \frac{tn_{ij}}{an_{ij}} \quad (2)$$

where  $tn_{ij}$  represents the target number of activating link  $l_{ij}$  per second, and  $an_{ij}$  does the average number of link activations per second.  $tn_{ij}$  can be given from the requested bit rate by:

$$tn_{ij} = \frac{rb_{ij}}{fb_{ij}} \times \frac{1}{1 - fe_{ij}} \quad (3)$$

where  $rb_{ij}$  represents the number of bits per second that link  $l_{ij}$  needs to be transmitted,  $fb_{ij}$  does the average number of bits in one transmitted frame, and  $fe_{ij}$  does the rate of causing the frame transmission error.  $an_{ij}$  can be given by:

$$an_{ij} = \frac{1}{ft_{ij}} \quad (4)$$

where  $ft_{ij}$  represents the average duration time of one frame transmission.

Among the parameters for the target link activation rate,  $rb_{ij}$  should be calculated by taking the summation of the bit rates requested by the applications using link  $l_{ij}$  in the routing path of WIMNET. The others,  $fb_{ij}$ ,  $fe_{ij}$ , and  $ft_{ij}$ , should be updated during communications by the following equations:

$$fb_{ij} = \frac{sb_{ij}}{sf_{ij}} \quad (5)$$

$$fe_{ij} = \frac{ff_{ij}}{sf_{ij} + ff_{ij}} \quad (6)$$

$$ft_{ij} = \frac{t}{sf_{ij} + ff_{ij} + of_{ij}} \quad (7)$$

where  $sb_{ij}$ ,  $sf_{ij}$ ,  $ff_{ij}$ , and  $of_{ij}$  represent the total number of successfully transmitted bits by link  $l_{ij}$ , the total number of successfully transmitted frames, the total number of failed frames, and the total number of transmitted frames of the interfered links with link  $l_{ij}$ , when  $t$  seconds have passed since the communication started in WIMNET, respectively.

### 3.3 Actual Link Activation Rate

The *actual link activation rate*  $ra_{ij}$  for link  $l_{ij}$  is obtained by dividing the number of successfully transmitted frames with the number of possibly activating chances for the link:

$$ra_{ij} = \frac{sf_{ij}}{ac_{ij}} \quad (8)$$

where  $ac_{ij}$  represents the number of possibly activating chances of link  $l_{ij}$ .

In the CSMA/CA protocol,  $ac_{ij}$  is hard to be obtained. Unlike the TDMA protocol where the link activations are synchronized by a single clock, the timing of counting the number of activating chances is not clear in the CSMA/CA protocol. Besides, the link activation chances resulting in transmission failures must be considered. In this paper,  $ac_{ij}$  is counted every time  $AP_i$  detects that the channel becomes free.

### 3.4 Active/Passive Backoff-Time

The *active backoff-time*  $ta_{ij}^m$  and the *passive backoff-time*  $tp_{ij}^m$  for link  $l_{ij}$  are calculated by the following procedure, where  $m$  represents the number of consecutively failed transmissions (or retry counter) due to heavy traffics and is saturated by 6. These backoff-times are updated every time the routing path is changed due to the topology change by adding a new AP or removing an existing AP and the host distribution change by a host join or leave to WIMNET. Then, they are fixed during communications.

1. Calculate the number of bits to be transmitted per second  $rb_{ij}$  for link  $l_{ij}$  by taking the summation of the bit rates for all the communication requests by the hosts using  $l_{ij}$ :

$$rb_{ij} = \sum_{k \in H_{ij}} hr_k \quad (9)$$

where  $H_{ij}$  represents the set of the host indices using link  $l_{ij}$  in the routing path, and  $hr_k$  does the requested bit rate (bps) of host  $k$ .

2. Sort every link in descending order of  $rb_{ij}$ , where the tiebreak is resolved by the number of hosts using this link for the routing path.
3. Set this sorted order to the link priority  $p_{ij}$  for  $l_{ij}$ .
4. Calculate the active/passive backoff-times for  $l_{ij}$ :

$$\begin{aligned} tamin_{ij}^m &= CW_{\min} \cdot \left(2^{m-1} + 2^{m-2} \cdot \frac{p_{ij}-1}{P}\right), \\ tamax_{ij}^m &= CW_{\min} \cdot \left(2^{m-1} + 2^{m-2} \cdot \frac{p_{ij}}{P}\right), \\ ta_{ij}^m &= rand \left[ tamin_{ij}^m, tamax_{ij}^m \right], \end{aligned} \quad (10)$$

where  $tamin_{ij}^m$  and  $tamax_{ij}^m$  represent the minimum and maximum values for the active backoff-time for  $l_{ij}$  when the retry counter is  $m$ ,  $CW_{\min}$  does the initial CW size, and  $P$  does the largest priority among the links. In our simulations,  $CW_{\min} = 31$  is used in any case.

$$\begin{aligned} tpm_{ij}^m &= CW_{\min} \cdot \left(2^{m-1} + 2^{m-2} \cdot \frac{P+p_{ij}-1}{P}\right), \\ tpmax_{ij}^m &= CW_{\min} \cdot \left(2^{m-1} + 2^{m-2} \cdot \frac{P+p_{ij}}{P}\right), \\ tp_{ij}^m &= rand \left[ tpm_{ij}^m, tpmax_{ij}^m \right]. \end{aligned} \quad (11)$$

where  $tpm_{ij}^m$  and  $tpmax_{ij}^m$  represent the minimum and maximum values for the passive backoff-time for  $l_{ij}$  when the retry counter is  $m$ .

### 3.5 Backoff-Time Switching at Link Activation Chance

In the FBS method, either of the active backoff-time or the passive backoff-time is selected for use at each link activation chance by the following procedure:

if  $rt_{ij} > ra_{ij}$  then select  $ta_{ij}^m$  else select  $tp_{ij}^m$ , for the backoff-time of  $l_{ij}$

where  $m$  represents the retry counter. Thus, from (10) and (11), the backoff-time is increased exponentially as the number of retransmissions is increased.

## 4. Implementation on QualNet

In this section, we present our implementation of the FBS method on the *QualNet* simulator.

### 4.1 Modified Functions in QualNet

For the implementation of the FBS method, we modified the following five functions in QualNet:

- *MacDot11StationProcessAck* deals with a reception of the ACK frame at a node when it can send a packet to the destination successfully.
- *MacDot11StationRetransmit* deals with a packet retransmission procedure at a node when it fails in the previous packet transmission due to collisions.
- *MacDot11StationProcessNotMyFrame* deals with transmitted frames that can be detected by a node, which indicates that the frames are transmitted by its interfered links.
- *MacDot11StationSetBackoffIfZero* deals with the procedure of setting a backoff-time, where the main function of the FBS method in calculating the target/actual link activation rates and setting a backoff-time for a link is implemented.
- *MacDot11AttemptToGoIntoWaitForDifsOrEifsState* deals with the waiting procedure for DIFS when the channel becomes clear.

### 4.2 Target Link Activation Rate

For the target link activation rate, the initial values of  $rb_{ij}$ ,  $fb_{ij}$ ,  $fe_{ij}$ , and  $ft_{ij}$  should be set before communications. Here, we note that each link in WIMNET has the maximum bit rate for transmissions. When  $rb_{ij}$  exceeds this capacity  $LC_{ij}$  for link  $l_{ij}$ , it should be saturated there to avoid the serious degradation of the performance of WIMNET.  $LC_{ij}$  can be calculated by:

$$LC_{ij} = \frac{NC \times \alpha}{\sum_{pq \in l_{ij}} rb_{pq}} \quad (12)$$

where  $NC$  represents the link capacity in WIMNET that must be specified beforehand in the *QualNet* simulation,  $\alpha$

does the coefficient to calculate the effective network capacity by considering the guard time and the acknowledgement response of a wireless link, and  $l_{ij}$  does the set of the interfered links with link  $l_{ij}$ . From our preliminary experiments using  $NC = 2(Mbps)$ , we set  $\alpha = 0.6$  in our simulations. Besides, the initial values for the number of bits per frame  $fb_{ij} = 2,272$ , the frame transmission error rate  $fe_{ij} = 0.1$ , and the frame duration time  $ft_{ij} = 0.02$  are used.

During communications, these parameter values should be automatically updated by using the obtained values of  $sb_{ij}$ ,  $sf_{ij}$ ,  $ff_{ij}$ , and  $of_{ij}$  in Sect. 3. For this purpose, the function *MacDot11StationProcessAck* in *QualNet* is modified to obtain  $sb_{ij}$  and  $sf_{ij}$  when  $AP_j$  receives an ACK message. Besides,  $ff_{ij}$  is increased when  $AP_i$  retransmits a failed frame by the function *MacDot11StationRetransmit*. To obtain  $of_{ij}$ , we use the NAV counter in the function *MacDot11StationProcessNotMyFrame*. In the function *MacDot11StationSetBackoffIfZero*, the target link activation rate  $rt_{ij}$  is updated by using the latest parameter values.

### 4.3 Actual Link Activation Rate

In our implementation,  $ac_{ij}$  is counted every time  $AP_i$  detects that the channel becomes free where no node is occupying the same channel, and  $sf_{ij}$  is counted every time link  $l_{ij}$  starts a data frame transmission.

In *QualNet*, one necessary variable *dot11->chance* for  $sf_{ij}$  is added into the function: *MacDot11AttemptToGoIntoWaitForDifsOrEifsState*, and the value is increased every time this function is called. Then, for  $ac_{ij}$ , the value is obtained from the variable *dot11->pktsToSend*. By using two variables *node->nodeId* and *dot11->currentNextHopAddress*, the link index  $ij$  of link  $l_{ij}$  is obtained.

For the implementation in *QualNet*, the abovementioned procedures are added in the function *MacDot11StationSetBackoffIfZero*.

## 5. Evaluation by Simulations

In this section, we discuss simulation results on *QualNet* to evaluate our proposal in this paper.

### 5.1 Simulation Environment

In our simulations, three network topologies with the static routing in Fig. 3 are examined, namely, a) Line, b) Grid, and c) Random. The topology for Random is given from [22]. The nominal bit-rate of each node is set 2 Mbps for  $NC$ , and the nominal wireless range is 250 m. In Line, the host and the GW act as both the source and destination of two data flows in opposite directions. For Grid and Random, every host sends data to the GW.

Each host performs CBR as a real-time UDP application, and FTP as a TCP application. For CBR, one packet is transmitted from each host to the GW at every 0.05 sec. (= 20 fps), where the packet size is changed from 160 bytes

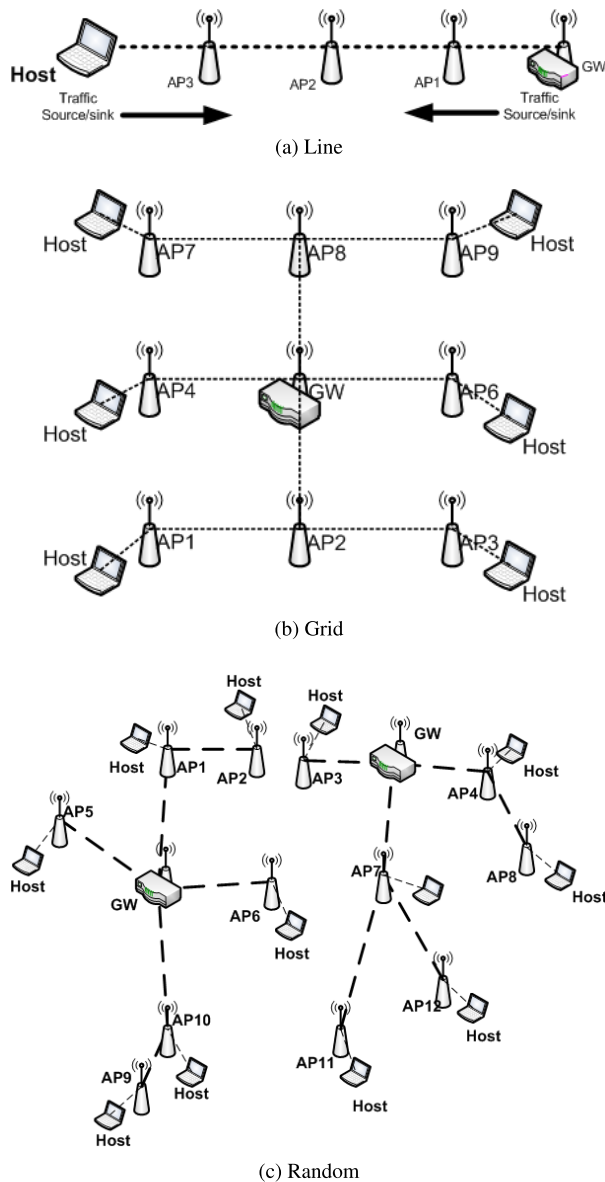


Fig. 3 Simulated network topology.

to 1280 bytes to examine the performance change under different traffic loads. For FTP, one file of different sizes from 160 bytes to 1280 bytes is transmitted from each host to the GW at every 0.05 sec. Each network simulation is carried on for 30 min., and the average result throughout the simulation is used in evaluations. The simulation environment is summarized in Table 1. Because the distance between adjacent APs is usually larger than that between a host and its associated AP in WIMNET, we selected 2 Mbps for an AP and 11 Mbps for a host as the nominal bit-rate.

To evaluate the performance of the FBS method through comparisons, the conventional CSMA/CA, the IEEE 802.11e protocol, and the method by Minooei et al. in [14] are together simulated. The 802.11e protocol has been defined to enhance the QoS (Quality of Service) of real-time UDP applications by adopting the EDCA (Enhanced Dis-

Table 1 Simulation environment.

Parameter	Value
Interface	IEEE802.11b
Nominal bit-rate	AP: 2 Mbps Host: 11 Mbps
Channel frequency	AP: 2.484 GHz Host: 2.412, 2.437, 2.462 GHz
Network simulation time	30 min.
Application	CBR for UDP FTP Generic for TCP
Packet rate for CBR	20 packets/sec.
Packet size for CBR	160, 320, 640, 1280 bytes
File rate for FTP	20 files/sec.
File size for FTP	160, 320, 640, 1280 bytes
Maximum retry counter	6
$CW_{min}$	31

tributed Channel Access) that allows different priorities to network services [23]. For the 802.11e protocol, we select priority 1 (Background) for CBR and priority 0 (Best effort) for FTP because we found that this priority selection gives the best performance in our preliminary experiments.

## 5.2 Evaluation for Real-Time Application

As the first scenario, we evaluate the performance in real-time applications using CBR.

### 5.2.1 Throughput

We compare the throughput when only CBR is adopted at any host. Figure 4 shows throughput results for different packet sizes in three topologies. They indicate that when the packet size is 160 or 320 bytes, all of the four methods for Line and the three methods except CSMA/CA for Grid, provide the same or similar highest throughput. However, in other cases, the FBS method always provides the best throughput among them.

### 5.2.2 Successful/Failed Link Activations

To investigate the cause of throughput results in Fig. 4, we examine the total number of successful and failed link activations during communications by the four methods. In the simulated instances, the number of packets is increased in order of Line, Grid, and Random, and one packet size is increased from 160 bytes to 1280 bytes, whereas the network bandwidth is fixed for any case. Thus, both successful and failed link activations can be increased there.

Figure 5 shows the total numbers of successful and failed link activations for each case. Here, we regard packet receptions as successful link activations, and ACK timeouts and RTS retransmissions as failed link activations. The results indicate that for the cases where several methods provide the same highest throughput, all of the necessary link activations are successfully completed, which indicates the sufficiency of the network bandwidth for them. For other cases, the number of successful link activations is the largest

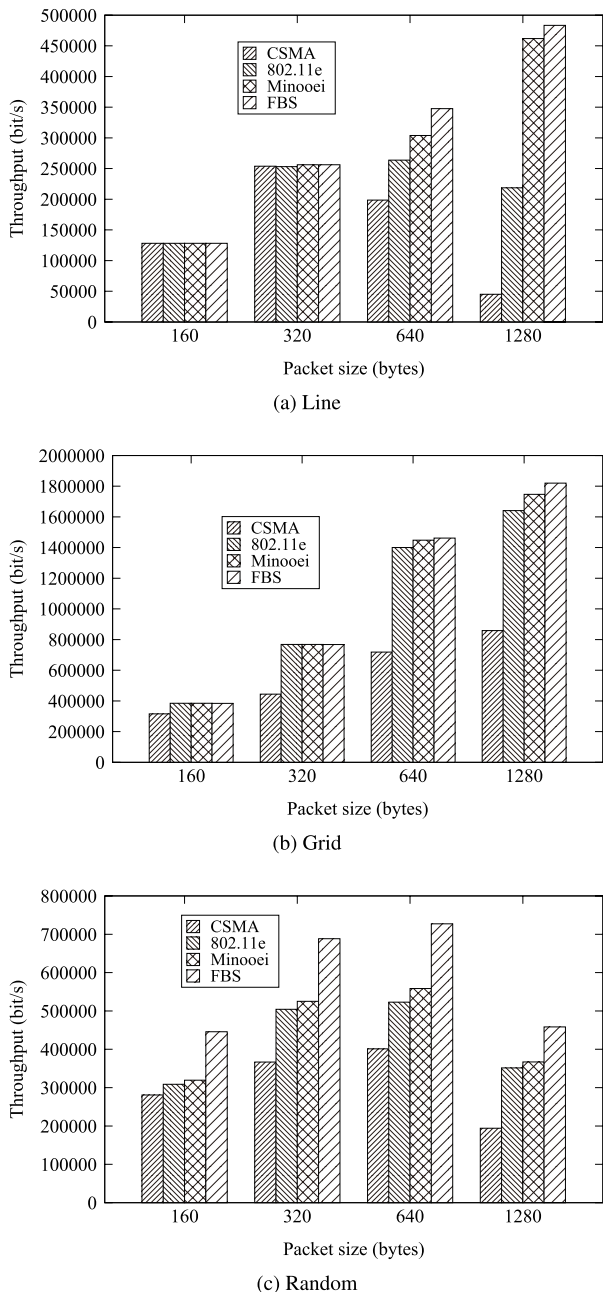


Fig. 4 Throughput for CBR (CSMA, 802.11e, Minooei, FBS).

and that of failed ones is the smallest for the FBS method among the four methods. Thus, as we discussed in Sect. 3.1, the FBS method can increase the number of successful link activations by assigning shorter backoff-times to congested links, and can reduce the number of failed link activations by assigning different backoff-times to the links.

5.2.3 Packet Loss

Then, we examine the packet loss probability during communications by the four methods. Figure 6 shows the ratio of the number of lost packets to the total number of transmitted packets from the transmitting nodes as the packet loss

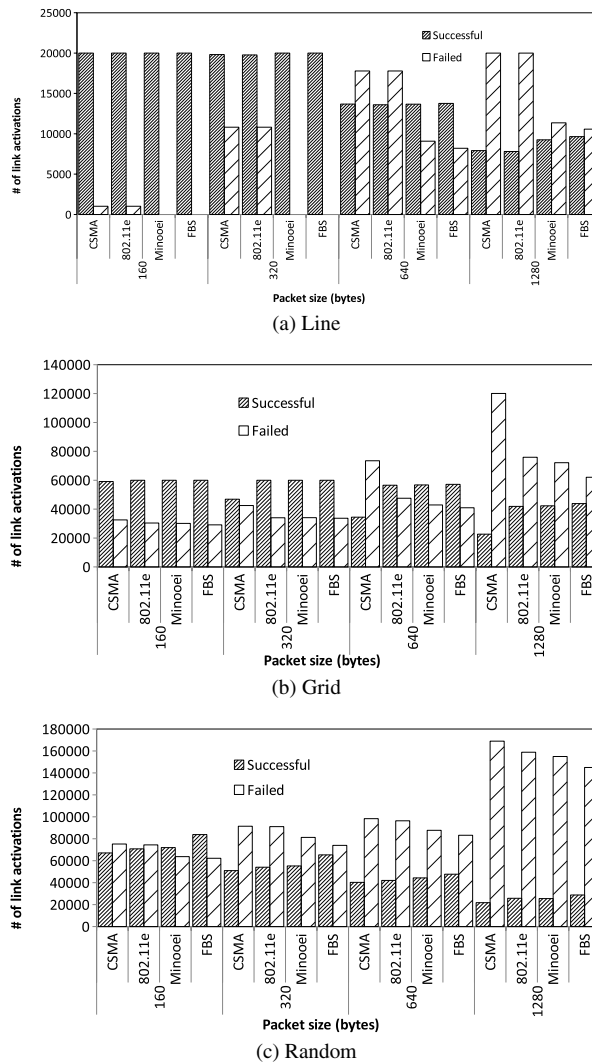


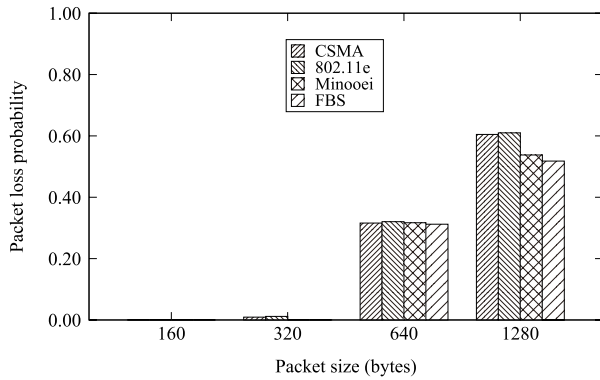
Fig. 5 Number of successful and failed link activations for CBR (CSMA, 802.11e, Minooei, FBS).

probability for each case. These results indicate that again, the FBS method generates the smallest packet loss.

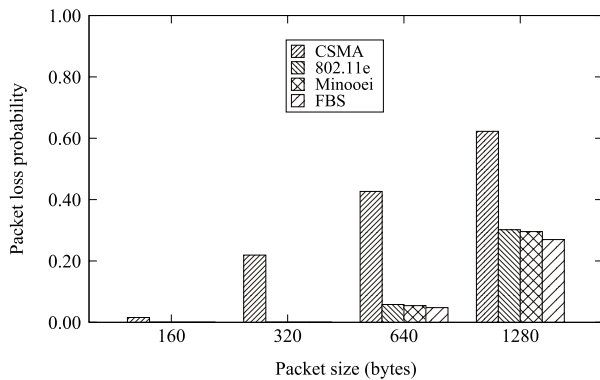
5.3 Evaluation for TCP Application

As known in [7], [8], a TCP application has the tendency of dominating the whole network bandwidth when the conventional CSMA/CA protocol is used. As a result, it has been observed in WIMNET that hosts closer to the GW in terms of hop counts consume the network bandwidth whereas hosts farer from the GW cannot take it. Thus, a host may receive unfair services in TCP applications depending on the location in WIMNET. The solution to this unfairness problem is another important issue in WIMNET, in addition to the total throughput improvement.

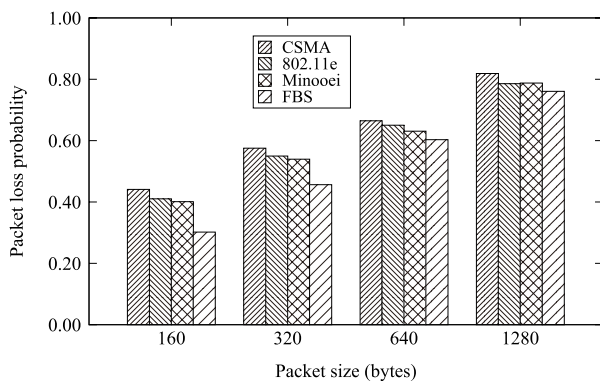
As the second scenario, to evaluate the contribution in solving this unfairness problem by the FBS method, we observe the difference between the highest and lowest throughputs among the hosts in addition to the average throughput,



(a) Line



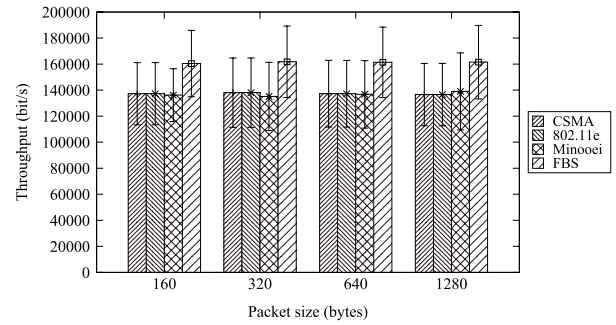
(b) Grid



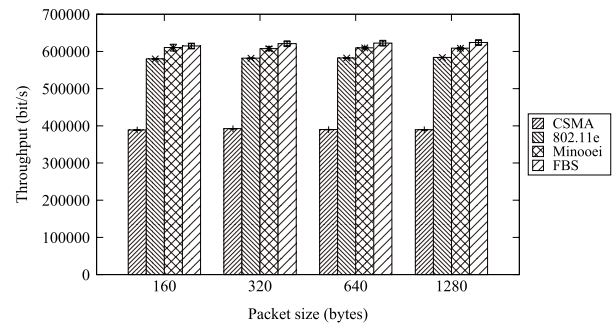
(c) Random

**Fig. 6** Packet loss probability for CBR (CSMA, 802.11e, Minooei, FBS).

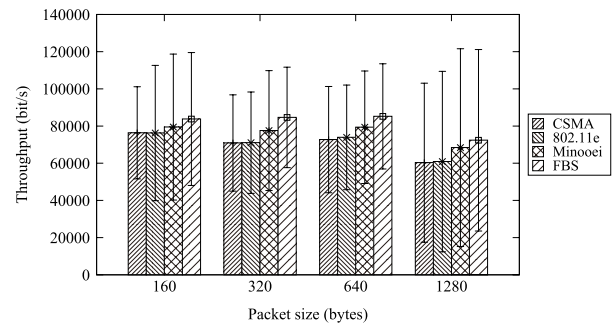
when every host executes only FTP. Figure 7 shows the results, where the thick bar represents the average throughput among the hosts, and the top/bottom edge of the thin line represents the highest and lowest throughput of one host. They indicate that the FBS provides the highest average throughput for any case, and generally reduces the difference between the highest and lowest throughputs. However, the further improvement of the fairness is necessary. The analysis of the reason causing the unfairness and its provision will be in our future studies.



(a) Line



(b) Grid



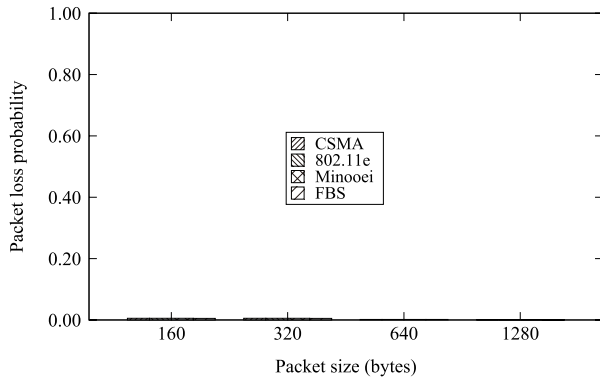
(c) Random

**Fig. 7** Throughput for TCP (CSMA, 802.11e, Minooei, FBS).

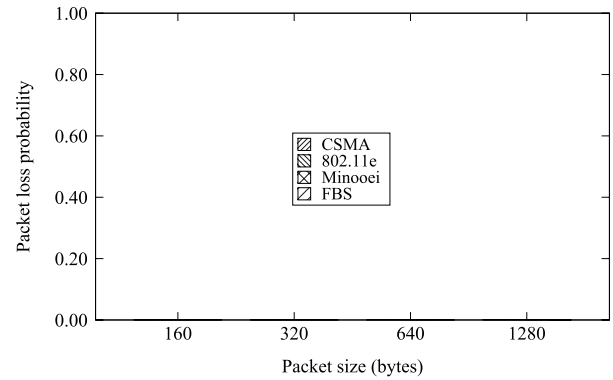
### 5.4 Evaluation for Mixture of Real-Time and TCP Applications

As the third scenario, we evaluate the performance when both real-time UDP applications and TCP applications appear in WIMNET as a typical situation in an access network. For the IEEE 802.11e, we set 1 for the priority class of real-time applications, and 0 for that of TCP applications. In Line, the host sends CBR packets to the GW, and the GW sends FTP packets to the host. In Grid, two hosts associated with AP4 and AP6 send FTP packets whereas the others send CBR packets. In Random, two hosts associated with AP6 and AP7 send FTP packets whereas the others send CBR packets. Figure 8 shows the packet loss probability for CBR by the four methods for different packet sizes, while any packet of FTP is received by the GW. This figure indicates that the FBS method can reduce the packet loss prob-

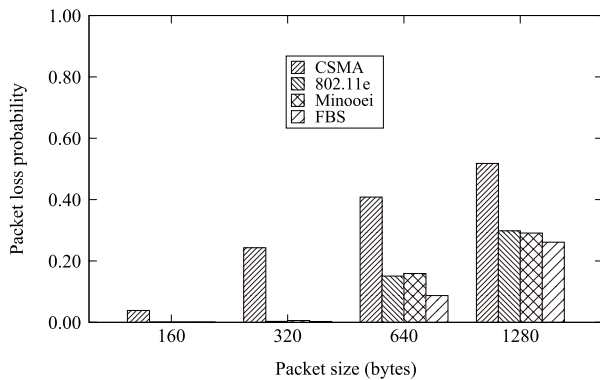




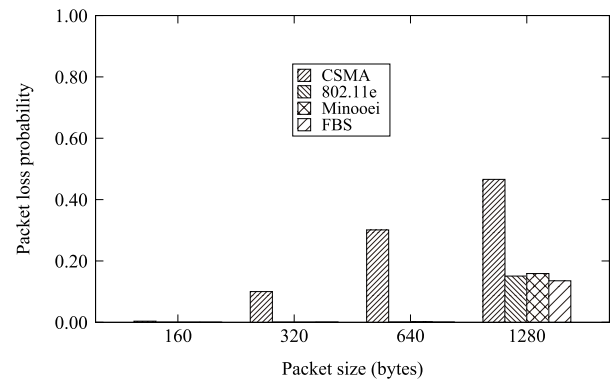
(a) Line



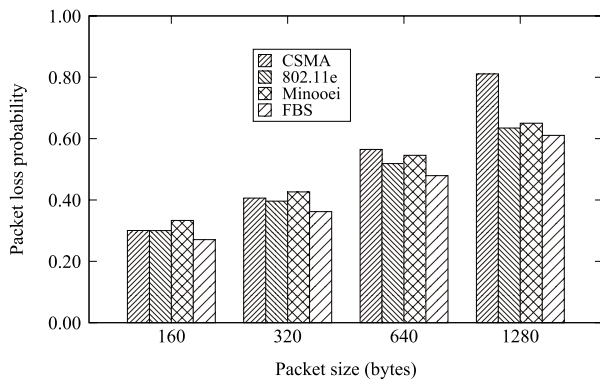
(a) Line



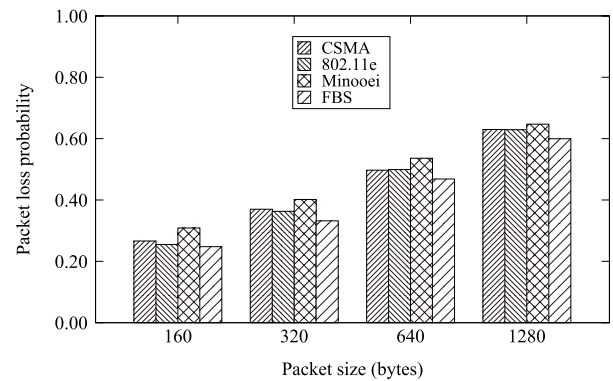
(b) Grid



(b) Grid



(c) Random



(c) Random

**Fig. 8** Packet loss probability for mixture of CBR and FTP (CSMA, 802.11e, Minooei, FBS).

**Fig. 9** Packet loss probability for mixture of CBR and FTP with leaky bucket traffic shaping (CSMA, 802.11e, Minooei, FBS).

ability more successfully than the other methods.

### 5.5 Evaluation with Leaky Bucket Traffic Shaping

As the last scenario, the *leaky bucket traffic shaping* [9] is adopted together at the GW to further reduce lost packets by avoiding the bandwidth domination of TCP applications, when both real-time and TCP applications appear. In our simulations, we used 32 kbps for the token rate and 10 kbps for the bucket size. Figure 9 shows the packet loss probability for CBR in this scenario. The comparison between the results in Figs. 8 and 9 indicate that for any method,

the application of the leaky bucket traffic sharing can reduce the packet loss probability to almost 0 for Line and by about 40% for Grid. However, it is not effective for Random except for the conventional CSMA/CA whose packet loss probability can be reduced by about 20%. This reason can be considered that the current link capacity is too small to handle the high traffic load in Random. Either of the increase of the link capacity or the rejection of some communication requests from hosts will be necessary in this case.

## 5.6 Discussion

The performance comparisons of the four methods in three topologies with four application scenarios show that the FBS generally provides better results than other methods. Thus, we conclude that the proposed FBS method is more suitable for WIMNET, whereas the advantage can be different and depend on the network topology and the traffic load. We recognize that the weak point of the FBS method is in the complexity of the parameter adjustment. The parameters need to be readjusted every time the routing path between the GW and the APs or the host location is changed. The autonomous adjustment mechanism will be in our next studies.

## 6. Conclusion

This paper presented the *Fixed Backoff-time Switching (FBS) method* for the CSMA/CA protocol in the *Wireless Internet-access Mesh Network (WIMNET)*, and its implementation on the *QualNet* simulator. The results of simulations of three network topologies with four real-time/TCP application scenarios using CBR and FTP confirmed the effectiveness of our proposal in terms of throughput, packet loss, and fairness among hosts. Our future works may include an autonomous parameter adjustment mechanism for the FBS method, further improvements in the unfairness problem for TCP applications, and evaluations in different network topologies and levels of traffic.

## Acknowledgment

This work is partially supported by KAKENHI (22500059).

## References

- [1] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Comput. Netw. ISDN Syst.*, vol.47, no.4, pp.445–487, March 2005.
- [2] Y. Zhang, J. Luo, and H. Hu, *Wireless mesh networking: Architectures, protocols and standards*, Auerbach Pub., 2006.
- [3] N. Funabiki, ed., *Wireless mesh networks*, InTech—Open Access Pub., 2011, <http://www.intechopen.com/books/show/title/wireless-mesh-networks>
- [4] Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11, 1999.
- [5] S. Tajima, N. Funabiki, and T. Higashino, "A proposal of fixed backoff-time switching method by link activation rate for wireless mesh networks," *Proc. Int. Conf. Complex, Intell., Software Intensive Sys. (CISIS)*, pp.647–652, 2011.
- [6] N. Funabiki, S. Sukaridhoto, Z. Wang, T. Nakanishi, K. Watanabe, and S. Tajima, "An implementation of fixed backoff-time switching method on IEEE 802.11 MAC protocol for wireless Internet-access mesh network," *Proc. Int. Work. Smart Info-Media Sys. Asia (SISA 2011)*, pp.67–72, Oct. 2011.
- [7] R. Morris, "TCP behavior with many flows," *Proc. Int. Conf. Netw. Protocols*, pp.205–211, Oct. 1997.
- [8] A. Vishwanath, V. Sivaraman, and G.N. Rouskas, "Anomalous loss performance for mixed real-time and TCP traffic in routers with very small buffers," *IEEE/ACM Trans. Netw.*, vol.19, no.4, pp.933–946, Aug. 2011.
- [9] A.S. Tanenbaum, *Computer Networks*, 5th ed., Prentice Hall, 2010.
- [10] *QualNet simulator*, Scalable network tech., <http://www.scalable-networks.com>
- [11] G.A.D. Caro, "Analysis of simulation environments for mobile ad hoc networks," *Tech. Rep.*, no.IDSIA-24-03, Dec. 2003.
- [12] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks," *IEEE Commun. Mag.*, vol.39, no.6, pp.130–137, June 2001.
- [13] K. Nakamura, S. Satake, and H. Inai, "An access method using fixed backoff for wireless LANs," *Electrical Eng. Japan*, vol.173, no.3, pp.30–39, 2010.
- [14] H. Minooei and H. Nojumi, "Performance evaluation of a new backoff method for IEEE 802.11," *Comp. Commun.*, vol.30, no.18, pp.3698–3704, 2007.
- [15] C.-M. Wu, H.-K. Su, and W.-H. Yang, "Efficient backoff algorithm in wireless multihop ad hoc networks," *Int. J. Advance. Comput. Tech.*, vol.3, no.1, pp.87–96, Feb. 2011.
- [16] R. Zhu, W. Shu, T. Mao, and T. Deng, "Enhanced MAC protocol to support multimedia traffic in cognitive wireless mesh networks," *Multi. Tool. App.*, pp.1–20, Dec. 2011.
- [17] H. Furukawa, "Hop count independent throughput realization by a new wireless multihop relay," *IEEE VTC2004-Fall*, pp.2999–3003, Sept. 2004.
- [18] Y. Tohzaka, K. Maruta, Y. Higa, and H. Furukawa, "Evaluations of wireless multihop network incorporating intermittent periodic transmit and packet forwarding path reservation," *IEICE Trans. Commun. (Japanese Edition)*, vol.J91-B, no.10, pp.1287–1298, Oct. 2008.
- [19] A. Yamada, A. Fujiwara, L.L. Yang, and B. Sadeghi, "EDCA based congestion control for WLAN mesh networks," *IEEE VTC2006-Spring*, pp.1288–1292, May 2006.
- [20] A. Kishida, M. Umeuchi, T. Onizawa, M. Iwabuchi, and H. Shibayama, "Centralized and distributed MAC protocol for IEEE802.11 WLAN," *IEICE Technical Report, RCS2011-35*, June 2011 (in Japanese).
- [21] A. Kishida, M. Iwabuchi, T. Shintaku, F. Nagase, T. Sakata, and T. Onizawa, "Fairness control method for centralized and distributed MAC protocol using CSMA/CA," *IEICE Technical Report, RCS2011-60*, Nov. 2011 (in Japanese).
- [22] S. Das, K. Papagiannaki, S. Banerjee, and Y.C. Tay, "SWARM: The power of structure in community wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol.19, no.3, pp.760–773, June 2011.
- [23] *IEEE Std 802.11e-2005*.



**Sritrusta Sukaridhoto** received the B.S. degree in Electrical Engineering from Institute Technology Sepuluh November of Surabaya, Indonesia, in 2002. He joined the Department of Telecommunication Technology at Electrical Engineering Polytechnic Institute of Surabaya, Indonesia, as a lecturer in 2002. He is currently a Ph.D. candidate in Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include computer networks. He is a student member of IEEE.



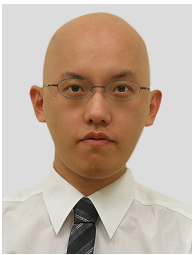
**Nobuo Funabiki** received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became

an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000–2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, and Web technology. He is a member of IEEE and IPSJ.



**Toru Nakanishi** received the M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000, respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor in 2003 and an associate professor in 2006, respectively. His research interests include cryptography, information security, and network protocol. He is a member of IPSJ.

He is a member of IPSJ.



**Kan Watanabe** received the B.S., M.S., and Ph.D. degrees in information technology from Tohoku University, Japan, in 2006, 2008, and 2011, respectively. In 2011, he joined the Department of Electrical and Communication Engineering at Okayama University, Japan, as an assistant professor. His research interests include distributed systems and wireless networks. He is a member of IEEE and IPSJ.



**Shigeto Tajima** received the B.S. degree in industrial engineering from Osaka Electro-Communication University, Japan, in 1992, and the Ph.D. degree in information and computer sciences from Osaka University, Japan, in 2010. From 1992 to 1995, he was with NS and I System Service Corp., Japan. In 1995, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an experimental officer, and became an assistant professor in 1996. His research interests include computer networks and optimization algorithms. He is a member of IPSJ.

He is a member of IPSJ.