IEICE TRANSACTIONS

on Communications

VOL. E100-B NO. 10
OCTOBER 2017

A PUBLICATION OF THE COMMUNICATIONS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. COMMUN., VOL.E100–B, NO.10 OCTOBER 2017

PAPER

Improving Fairness in Wireless Ad Hoc Networks by Channel Access Sensing at Link Layer and Packet Rate Control

Nguyen Minh TUAN[†a)], *Nonmember*, **Kohei WATABE**[†], *Member*, **Pham Thanh GIANG**[††], *Nonmember*, *and* **Kenji NAKAGAWA**[†], *Member*

SUMMARY Wireless Ad hoc networks have been rapidly developed in recent years since they promise a wide range of applications. However, their structures, which are based on the IEEE 802.11 standard, cause a severe unfairness problem in bandwidth sharing among different users. This is an extreme drawback because in wireless ad hoc networks, all users need to be treated fairly regardless of their geographical positions. In this paper, we propose a method to improve the fairness among flows by sensing channel access of other nodes based on the information obtained at the link layer and then, controlling the packet sending rate from the link layer to the MAC layer and the dequeue rate from the queue. Simulation results show that the proposed method achieves a better fairness with a good total throughput compared to conventional methods.

key words: *fairness, throughput, wireless ad hoc network, channel utilization, dequeue rate, Round Robin, link layer*

1. Introduction

Recently, wireless ad hoc networking has emerged and been largely studied with many interesting problems such as routing, Quality of Service and security. Although wireless ad hoc networks are now diverged to many new research directions [2], the fairness problem, on which we focus in this paper, is still important and need to be addressed.

Most of wireless ad hoc network architectures are currently based on the random access method of IEEE 802.11 [5] Distributed Coordination Function (DCF) in CSMA/CA. DCF enables to distribute channel bandwidth between nodes, and prevent the channel from collision. However, IEEE 802.11 is not really well-suited for wireless ad hoc networks [6], [8] since it often causes unfairness situations and throughput degradation. In particular, it usually leads to the unfairness where a few flows tend to dominate and occupy almost all the channel bandwidth. As a result, the throughputs of these flows are remarkably high while other flows only receive extremely low throughputs.

There are many factors causing this problem. For instance, at the MAC layer, the existence of transmission range (TR) and carrier sensing range (CR) causes a problem, which is called a large-EIFS problem [4] where nodes that are far from the destination sometimes must wait longer time Extended Interframe Space (EIFS) than DCF Interframe Space (DIFS). As a result, the throughput of flows transmitted from these far nodes are lower than that of flows from nodes which are geographically near to the destination.

In addition, the link layer buffer management plays an important role in fairness achievement. In wireless ad hoc network architecture, at the link layer, First In First Out (FIFO) queue is used by default. At an intermediate node, the queue at the link layer is responsible for sending both packets of the direct flow generated by the node itself and also that of the forwarding flows generated by other nodes. A packet of forwarding flow needs some time to arrive at the queue from its source while a packet of the direct flow can be enqueued immediately as soon as it is generated. As a result, packets of the direct flow may occupy almost all the buffer space, then packets of forwarding flows are dropped due to buffer overflow.

Besides, the fairness considerably depends on the type of topology, or the positions of nodes. A typical example is a three-pair topology [3] (also known as the "flow in the middle" problem [7]) as in Fig. 1. In this topology, nodes S1, S2 and S3 transmit their packets to R1, R2 and R3, represented by flow 1, flow 2 and flow 3, respectively. In Fig. 1, (0, 250) written above the node R1 shows that the X-coordinate of R1 is 0 [m], and the Y-coordinate is 250 [m], further, G [Mbps] denotes the offered load generated by each

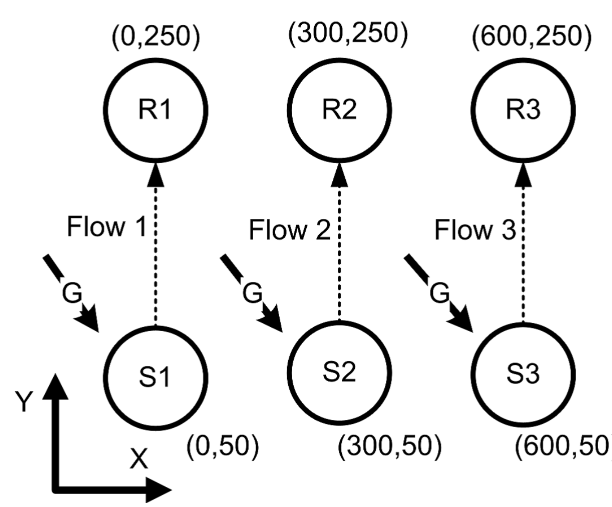

Fig. 1 Three-pair topology.

Manuscript received September 20, 2016.
Manuscript revised February 17, 2017.
Manuscript publicized April 14, 2017.

[†]The authors are with the Graduate School of Engineering, Nagaoka University of Technology, Nagaoka-shi, 940-2188 Japan.
[††]The author is with the Department of Telematics, Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam.
a) E-mail: tuannmdt2k52@gmail.com
DOI: 10.1587/transcom.2016EBP3359

node.

In the topology of Fig. 1, the unfairness of throughputs occurs as follows. If we set the channel bandwidth $B = 2$ [Mbps] and $G = 1.6$ [Mbps], then the network becomes the saturation state, i.e., every node always has a packet to transmit. The simulation result shows that the throughputs of flow 1 and flow 3 are 1.42[Mbps] while that of flow 2 is only 0.01[Mbps]. This unfairness is caused by the asynchronous transmission between two exterior flows, flow 1 and flow 3. In particular, link S1-R1 and S3-R3 are active by turns and hence, S2 always senses a busy channel and has little chance to transmit its packets. In this case, the Binary Exponential Back-off (BEB) has no effect.

In this paper, we will propose a method for achieving a good fairness of throughputs among flows sharing the same channel. Our method consists of two algorithms, Algorithms 1 and 2, both of which work on the link layer. In Algorithm 1, a channel access of other node is sensed by using the link layer information to control the packet sending rate to the MAC layer. In Algorithm 2, the packet dequeue rate from the link layer queue to the MAC layer is controlled by the packet dequeue rate of each flow.

The rest of the paper is organized as follows. In Sect. 2, we briefly show some related works in the literature. In Sect. 3, the protocols of our proposed method are given, followed by the simulation results in Sect. 4. Finally, Sects. 5 and 6 give a conclusion and future work.

2. Related Work

Various reasons that cause the unfairness problem at the MAC layer of IEEE 802.11 have been studied extensively [6], [8], [10]. One of the well known problems is the hidden terminal problem. RTS/CTS handshake can greatly reduce the influence of hidden terminal problem, however, this problem cannot be solved completely in some situations where the interference range is very large [9]. The radius of interference range is not a fixed value and it changes according as the transmitter-receiver distance. The effectiveness of RTS/CTS is not good when the transmitter-receiver distance is larger than a certain value.

In [4], Li, et al. discovered that the fixed EIFS delay leads to unfairness for a wireless node that is far from the destination. Then, they proposed a dynamic EIFS adjustment based on the location of a node to deal with such situation. However, their core idea stands on the length of sensing range (SR) that cannot always be identified correctly due to the mobility and collision among wireless nodes.

Several schemes for improving the fairness of MAC protocols have been proposed. For instance, in [11], Chakraborty, et al. introduced a new method in which Contention Window is dynamically changed based on the difference between the actual channel share and the required channel share in order to achieve the proportional fairness. Nevertheless, the actual share of a node is calculated from the information of overhearing data and control frames which are not stable in the multi-hop environment. In [12], Razafindralambo, et al. proposed a novel scheme to prevent from collision and monopolization by inserting a waiting time along with the standard back-off algorithm based on sensing activities of a node and collision experience. However, in their proposed method, it is difficult to detect node's monopolization in some configurations. They only can solve it by transmitting a packet with a large contention window after many consecutive packets were transmitted successfully. Note that all above methods [4], [11], [12] need to modify the existing IEEE 802.11 MAC layer, which is not favored from the viewpoint of implementation.

The unfairness problem at the link layer is also severe and thus addressed in the literature. In [13], Nandiraju, et al. studied the weakness of the FIFO queueing, then they suggested a fair scheduling algorithm called Dual Queue Service Differentiation (DQSD). DQSD is implemented at intermediate nodes where the direct flow and forwarding flows coexist. It uses two queues for handling the packets from these flows. Nevertheless, since there is only one queue used for the forwarding packets, it is not ensure the equal handling for packets from a distant node. Furthermore, Shagdar, et al. [14] and Jun, et al. [15] also investigated the unfairness of the direct flow and forwarding flows. They proposed a scheduling algorithm by using Round Robin (RR) scheduling. However, they assumed ideal MAC layer fairness which cannot be really achieved.

Giang, et al. [16] modified the RR mechanism. They pointed that using just the RR mechanism cannot result in good fairness. Then, they proposed Probabilistic Control on Round Robin Queue (PCRQ), in which the RR is applied in combination with three algorithms for queue control. We show the detailed algorithm of PCRQ, because we will compare our proposed method with PCRQ.

2.1 Algorithms of PCRQ

There are three algorithms in PCRQ.

The first algorithm is applied to control packets for enqueueing to the link layer queue. An arrived packet is enqueued with probability

$$P_i^{input} = \begin{cases} 1, & \text{if } qlen_i \le ave, \\ 1 - \psi_1 \dfrac{qlen_i - ave}{(N-1)ave}, & \text{if } qlen_i > ave, \end{cases}$$

where ψ_1 is a constant, N is the number of flows, $qlen_i$ is the queue length of the flow i, and ave is the average of $qlen_i$, $i = 1, \ldots, N$. By this algorithm, packets of flows with heavy offered load, usually the direct flow, are likely to be dropped. Therefore, this algorithm maintains a good balance of the queue length of flows.

The second algorithm controls the turn of queue reading. The reading pointer is hold at queue i with probability

$$P_i^{turn} = \begin{cases} \psi_2 \dfrac{N \times ave}{qmax}, & \text{if } qlen_i = 0, \\ 0, & \text{if } qlen_i > 0, \end{cases}$$

where ψ_2 is a constant, $qmax$ is the maximum queue length.

The delay time δ[s] is given as a holding time of the reading pointer at an empty queue. By this algorithm, many packets from a distant node can arrive at the queue during the holding time δ, hence it solves the unfairness of bandwidth among flows at the MAC layer.

The third algorithm controls an advantageous flow not to send too many packets to the MAC layer by decreasing the number of output packets from RR queues. A packet at the head of the queue of flow i is sent to the MAC layer at probability

$$P_i^{output} = \begin{cases} 1, & \text{if } qlen_i \leq ave, \\ 1 - \psi_3 \dfrac{qlen_i - ave}{(N-1)ave}, & \text{if } qlen_i > ave, \end{cases}$$

where ψ_3 is a constant. Then, if a packet is not sent to the MAC layer, it will be delayed for δ, which is the same value as in the second algorithm. By this algorithm, the forwarding flows have more chance of sending packets.

By these three algorithms, PCRQ works effectively at the link layer. Nonetheless, the PCRQ cannot solve the unfairness problem in some cases where the link layer congestion does not occur such as in the case of three-pair topology (Fig. 1). Moreover, to determine the optimal values of the parameters ψ_1, ψ_2, ψ_3 is not an easy task.

3. Proposed Method

We propose a method to achieve a good fairness among flows by using only local information obtained at the link layer of each node. Our method consists of two algorithms, Algorithm 1 and Algorithm 2, both of which work on the link layer.

In Algorithm 1, a channel access of other node is sensed by the link layer information to control the packet sending rate to the MAC layer by giving a delay if the node is an advantageous node.

In Algorithm 2, the packet dequeue rate from the link layer queue to the MAC layer is controlled by the packet dequeue rate of each flow by skipping the service of a packet at the queue of an advantageous flow.

3.1 Algorithm 1 - Channel Access Sensing and Sending Rate Control

3.1.1 Channel Access Sensing

We first consider the sensing of channel access by other nodes based on the channel utilization ρ of a node that is defined by

$$\rho = \frac{\lambda}{\mu}, \tag{1}$$

where λ[1/s] is the packet transmission rate and μ[1/s] is the service rate. μ is assumed to be a constant $\mu = B/L$, with the channel bandwidth B[Mbps] and the constant packet length L[bit]. λ is nearly equal to the rate that packets are

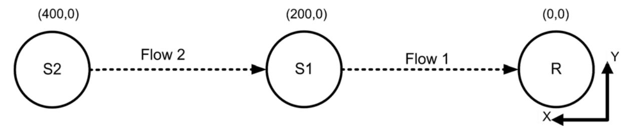

Fig. 2 A three-node chain topology.

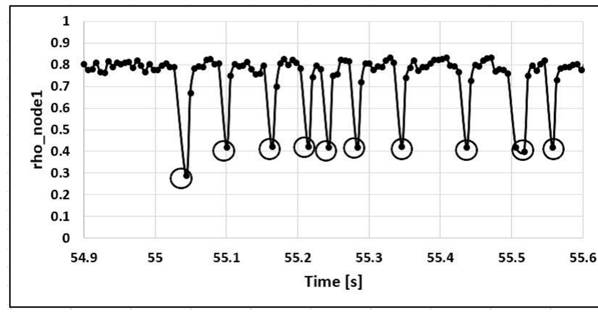

Fig. 3 The change of ρ at S1 in three-node chain topology.

sent from the link layer to the MAC layer, hence, λ can be approximated by

$$\lambda \simeq \frac{1}{\Delta t}, \tag{2}$$

where Δt[s] is the average time between two consecutive packets sent to the MAC layer at the node, which is calculated by

$$\Delta t = \alpha \Delta t(last) + (1 - \alpha)(t(cur) - t(last)), \tag{3}$$

where α is a constant with $0 < \alpha < 1$, $\Delta t(last)$ is the last value of Δt, $t(cur)$ is the time that the current packet is sent, $t(last)$ is the time that the last packet was sent.

Thus, from (1), (2) and (3), we have

$$\rho \simeq \frac{1}{\mu \Delta t}. \tag{4}$$

The channel utilization ρ reflects the channel access ability of a node. The higher the channel utilization at a node is, the more smoothly the packets at that node are transmitted.

For three-node chain topology of Fig. 2, we show the change of ρ at node S1 in Fig. 3. As we can observe from Fig. 3, because of the MAC layer problem, S1 can transmit packets continuously, leading to a high value of ρ that is about 0.8. However, there are some points of time that the ρ at S1 suddenly decreases. At this point, the disadvantageous node S2 accessed the channel and caused a decrease in ρ at S1. Hence, catching this moment, the advantageous node S1 can detect the appearance of other node, S2 in this case.

As shown in the above example, each time a node detects a decrease in its utilization, the node notices that other node accessed the channel and transmitted a packet. Hence, this node should defer in sending its packets so that other nodes have chance to transmit their packets. In addition to the above example, we also show in Fig. 4 the change of ρ at node S1 in the case of three-pair topology (Fig. 1). As shown

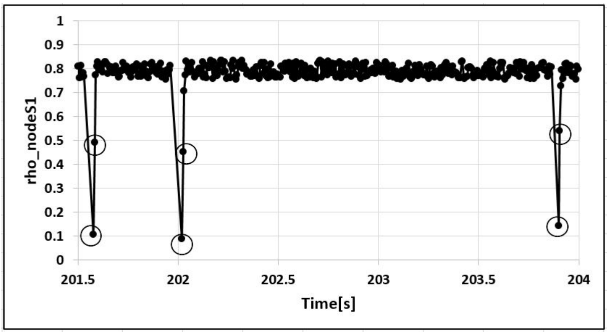

Fig. 4 The change of ρ at S1 in three-pair topology.

Table 1 Algorithm 1—Channel access sensing and sending rate control.

Algorithm 1 Channel Access Sensing and Sending Rate Control

when a packet is sent to the MAC Layer
calculate Δt by (3)
Let *safe_interval* = T_{DIFS}, which is the period of *DIFS*
N is the number of flows
if $\Delta t > \Delta t(last) + safe\_interval$
  give delay $\Delta t/N$
  send the packet to the MAC Layer
else
  immediately send the packet to the MAC Layer
end if

in Fig. 4, although the flow 2 is a disadvantageous flow, it sometimes can transmit packets. The packet transmission of flow 2 causes a sudden decrease in ρ at node S1 and S3. Therefore, S1 and S3 notice that there is another flow that is contending the channel with them.

3.1.2 Sending Rate Control

Based on the above analysis, the operation of Algorithm 1 is shown in Table 1.

A (sudden) decrease of ρ is detected by a large increase of Δt, which is decided by $\Delta t > \Delta t(last) + safe\_interval$ in Algorithm 1. We set $safe\_interval = T_{DIFS}$, where T_{DIFS} is the period of DIFS. The reason of *safe_interval* is explained as follows. At the end of a packet transmission cycle, when a packet is successfully transmitted and an ACK is received at the sender, this sender will perform a back-off process followed by a DIFS time before starting a new packet transmission. Hence, when comparing between $\Delta t(last)$ and the new value of Δt, it is necessary to add $safe\_interval = T_{DIFS}$.

If an increase of Δt is detected, we give a delay before sending the packet to the MAC layer by the command "give delay $\Delta t/N$" in Algorithm 1. The delay value $\Delta t/N$ aims to the case of long chain topology where a node may be responsible for packets from many different nodes. In this case, a node with a heavier load should defer a smaller time than the others. For example, in four-node chain topology, a node with three flows should defer smaller time than a node with only two flows. In addition, the division by N avoids too much delay time, that may waste bandwidth. Note that this is just a relative comparison because Δt is not same among

Table 2 Algorithm 2—Dequeue rate control.

Algorithm 2 Dequeue Rate Control

Each time the reading pointer points to queue i
Let T = the time that the reading pointer points to queue i,
calculate η_i by (7)
$\bar{\eta}$ by (6) and $variance = \frac{1}{N}\sum_{i=1}^{N}(\eta_i - \bar{\eta})^2$ where N is the number of flows
if $\eta_i < \bar{\eta}$ and $(\eta_i - \bar{\eta})^2 > variance$
  flow i is considered to be an advantageous flow
  skip this queue without serving a packet
  throw away the value η_i
else
  flow i is considered to be a disadvantageous flow
  dequeue a packet from queue i
  put $t_i(cur) = T$
  update η_i
  put $\eta_i(last) = \eta_i$
  put $t_i(last) = t_i(cur)$
end if
point the reading pointer to the next queue

nodes.

3.2 Algorithm 2 - Dequeue Rate Control

Algorithm 2 operates based on the dequeue rate control for Round Robin queue.

Buffer queue at the link layer is Round Robin mechanism where each queue serves packets from each flow separately. The dequeue rate is determined by the dequeue interval. The *dequeue interval* η_i[s] is defined as follows. At every departure epoch of a packet from the queue i, the dequeue interval η_i of a queue i is defined by

$$\eta_i = \beta\eta_i(last) + (1-\beta)(t_i(cur) - t_i(last)), \tag{5}$$

where β a constant with $0 < \beta < 1$, $\eta_i(last)$ is the last value of η_i, $t_i(cur)$ is the time that the current packet is dequeued at queue i, $t_i(last)$ is the time that the last packet was dequeued at queue i. Then, the *average dequeue interval* $\bar{\eta}$ is defined by

$$\bar{\eta} = \frac{1}{N}\sum_{i=1}^{N}\eta_i, \tag{6}$$

where N is the number of queues, η_i is the value calculated by (5).

The operation of Algorithm 2 is shown in Table 2. The Round Robin queues are used in this Algorithm. Let T be the time that the reading pointer points to queue i, then η_i is temporally calculated by

$$\eta_i = \beta\eta_i(last) + (1-\beta)(T - t_i(last)). \tag{7}$$

Then we check whether η_i is smaller than the average dequeue interval $\bar{\eta}$ of (6). If $\eta_i < \bar{\eta}$, we see that the dequeue rate of this queue is larger than the average. Further, for the sake of a reliable decision, we add one more condition $(\eta_i - \bar{\eta})^2 > variance$, which implies that the dequeue rate of the queue i is large enough. If the above two conditions are satisfied, we consider that the flow of this queue i is an

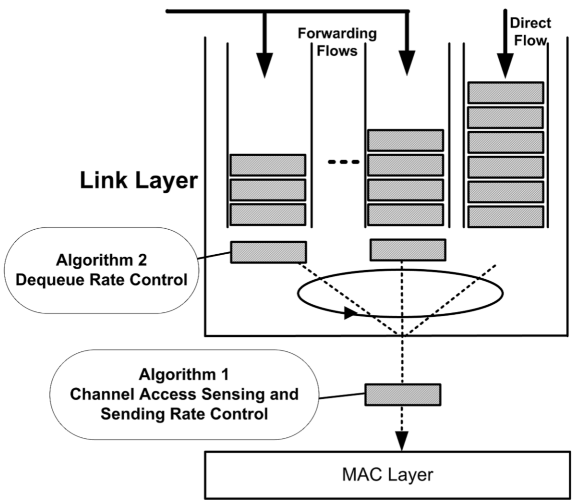

Fig. 5 System model of the proposed method.

advantageous flow, so reading this queue is skipped without serving a packet to reduce the dequeue rate for achieving the fairness, and throw away the temporal value η_i of (7).

If $\eta_i < \bar{\eta}$ or $(\eta_i - \bar{\eta})^2 > variance$ are not satisfied, we consider that the flow of queue i is a disadvantageous flow. Then dequeue a packet of queue i immediately at time T, and put $t_i(cur) = T$, update η_i, and put $\eta_i(last) = \eta_i$, $t_i(last) = t_i(cur)$.

As is known, in multi-hop ad hoc network, if the queue is FIFO, flows from neighboring nodes (including the direct flow) usually occupy almost all the buffer space and as a result, packets from the far distant nodes are frequently dropped due to buffer overflow. Hence, it is necessary to apply the Round Robin queue technique to separate packets from different flows. Moreover, unlike packets from the direct flow that can be enqueued immediately after generated, packets from forwarding flows need some time to arrive at the queue. This causes unfairness when reading the queue even if the queue is modeled as the Round Robin mechanism. Algorithm 2 is implemented to solve this problem. This algorithm ensures the fairness among flows based on the control of the dequeue rate of flows at intermediate nodes, and this is done by controlling the dequeue interval of each queue as shown in Table 2.

The system model of Algorithms 1 and 2 is illustrated in Fig. 5.

4. Performance Evaluation

In this section, we compare our proposed method with PCRQ [16] and standard 802.11 by the simulator NS-2 version 2.35 [17]. NS-2 simulation parameters are shown in Table 3. In the simulation, the channel bandwidth is set to 2[Mbps], and the effective bandwidth becomes about 1.4[Mbps] due to the overhead in IEEE 802.11 [4]. For each flow, we gradually increase the number of packets per second from 1 to 250 packets. In other words, the offered load at a node increases from 8[kbps] to 2[Mbps]. If the offered load is large enough,

Table 3 NS-2 parameters configuration.

Channel bandwidth	2 Mbps
Antenna type	Omni direction
Radio Propagation	Two-ray ground
Transmission range	250 m
Carrier sensing range	550 m
MAC protocol	IEEE 802.11b (RTS/CTS is enabled)
Routing protocol	DSDV
Connection type	UDP/CBR and TCP
Queue type	FIFO, PCRQ, proposal
Maximum Queue length	100 packets
Packet's size	1024 Bytes
Slot time	20 μs
DIFS	50 μs
EIFS	314 μs
Simulation time	300 s

the network is in the saturated state and a single flow may occupy all the channel bandwidth. We start the evaluation from 50 seconds to ensure that the simulation is in the stable state. The parameter values α and β are set to $\alpha = 0.1$ and $\beta = 0.6$, which are determined by extensive simulations. The metrics used to evaluate are the throughput and fairness index. The fairness index (FI) [18] is defined by

$$FI = \frac{\left(\sum_{i=1}^{N} Th_i\right)^2}{N\sum_{i=1}^{N}(Th_i)^2}, \tag{8}$$

where N is the number of flows, Th_i is the throughput of flow i. The fairness index ranges from $1/N$ to 1, where $FI = 1/N$ indicates the worst fairness and $FI = 1$ the perfect fairness. We also evaluate the total throughput $\sum_{i=1}^{N} Th_i$.

In the following subsections, we will show the results of performance evaluation for four fundamental scenarios. Scenario 1 is a typical MAC unfairness topology called "flow in the middle" that is studied in [3], [12]. Scenario 2 is a well-known multi-hop wireless network called "chain topology" that is studied in [4], [12], [13], [14], [16]. Actually, it is a typical topology of wireless network where a destination station acts as a gateway connecting to the Internet. Scenario 3 is a mixed topology of above scenarios 1 and 2. Scenario 4 is evaluated to show the effectiveness of our proposed method in case of chain topology with the mixture of UDP and TCP flows.

4.1 Scenario 1

Scenario 1 is the three-pair topology shown in Fig. 1. In this type of topology, the central flow is starved due to the asynchronous transmission of two external flows. The fairness and throughput results are presented in Fig. 6, Fig. 7 and Fig. 8.

In scenario 1, the channel access contention occurs but there is no buffer contention, thus only Algorithm 1 works in our method. Fig. 6 shows that our proposed method yields

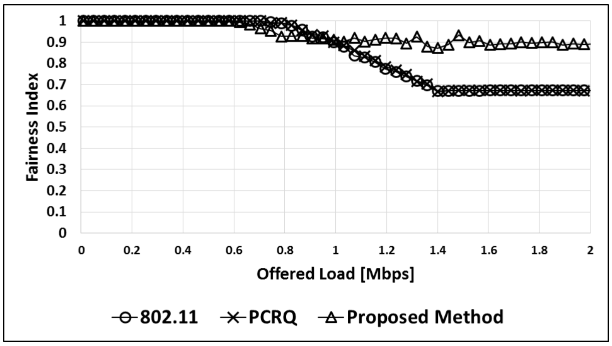

Fig. 6 Fairness index—Three-pair topology.

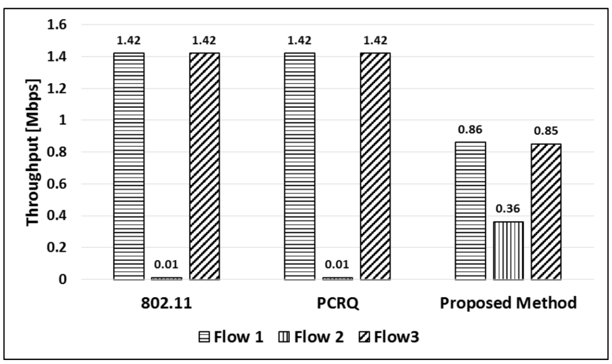

Fig. 7 Per-flow throughput—Three-pair topology.

larger fairness index compared to both PCRQ and 802.11 standard. When the offered load is large, that is, the network is in the saturation state, in PCRQ and 802.11, the node S2 rarely senses the idle state of channel because of the continuous transmission of nodes S1 and S3. In particular, in this scenario, PCRQ doesn't work because forwarding flows do not exist. Hence, the throughput of flow 2 is almost zero.

On the other hand, in our proposed method, catching the moments that S2 transmits packets, S1 and S3 can detect the existence of S2 by the decrease of their channel utilization. Then, S1 and S3 decrease their own utilization by giving a delay before sending their packets to the MAC layer so that S2 can get more chance to access the channel. As a result, in our proposed method, the throughput of flow 2 increases, leading to a better fairness index. This is also confirmed by the per-flow throughput in Fig. 7. In this figure, the offered load G is set to the channel bandwidth 2[Mbps]. From this figure, the central flow's throughput in the case of PCRQ and 802.11 standard are almost zero, while in our proposed method, it can be improved remarkably.

The total throughput is shown in Fig. 8. From this figure, it can be seen that the total throughput in our proposed method is smaller than the others. This can be explained as follows. Assume the channel bandwidth is B. Then, in the worst case of fairness, i.e., the two external flows transmit at the maximum throughput that equals B and the central flow 0 throughput. Hence, in this case, the total throughput is $2B$.

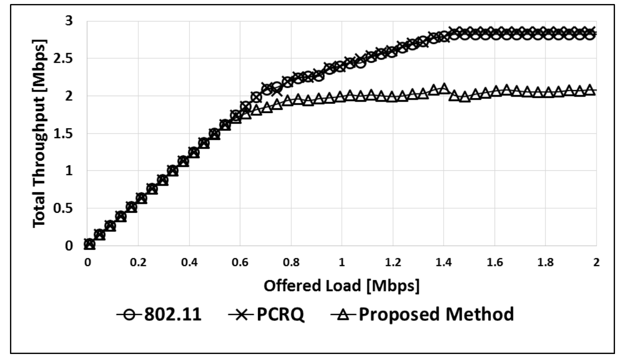

Fig. 8 Total throughput—Three-pair topology.

On the other hand, in the case of the perfect fairness where 3 flows share the channel bandwidth equally, each flow can obtain a throughput of $B/2$. Thus, in this case, the total throughput is $3B/2$, which is smaller than $2B$. So, we can see that there exists the tradeoff between the fairness and the total throughput.

Now, let us compare the throughput reduction ratio of IEEE 802.11 with that of our proposed method. In the case of IEEE 802.11, the ideal total throughput is 4[Mbps] ($= 2B = 2 \times 2$[Mbps]) and the experimental result (in Fig. 7) is $1.42 + 1.42 + 0.01 = 2.85$[Mbps]. Hence, the throughput reduction ratio is $2.85/4 = 0.71$. Then, in the case of proposed method, the ideal total throughput is 3[Mbps] ($= 3B/2 = 1.5 \times 2$[Mbps]) and the experimental result is $0.86 + 0.85 + 0.36 = 2.07$[Mbps]. Thus, the throughput reduction ratio is $2.07/3 = 0.69$, which is similar to that of IEEE 802.11. Therefore, there is no degradation of the throughput reduction ratio in our method.

4.2 Scenario 2

Scenario 2 is a five-node chain topology shown in Fig. 9. Four nodes S1-S4 generate UDP packets to the destination R. This type of topology causes the unfairness at both MAC and link layers. Thus, the far distant flows suffer very small throughput.

The fairness index and throughput comparison are shown in Fig. 10, Fig. 11 and Fig. 12. It can be clearly seen from Fig. 10 that in our proposed method the fairness index is greatly improved compared to both PCRQ and 802.11 standard. In the conventional protocol 802.11, because of the weakness of FIFO queue as analyzed in [16], in the FIFO queue at S1, packets of the direct flow occupy almost all the buffer space. Hence, packets from other flows cannot be enqueued because the queue is full of packets of the direct flow. In addition, due to the effect of MAC unfairness problems like large-EIFS [4], the nodes S2, S3, and S4 only have a little chance for packet transmission.

On the other hand, PCRQ works by comparing queue lengths of the flows, then it can achieve better fairness than 802.11. However, when the offered load value is from 0.2 to 1.2 in Fig. 10, i.e., the link utilization is small, the queue lengths are short, thus the difference of queue lengths among

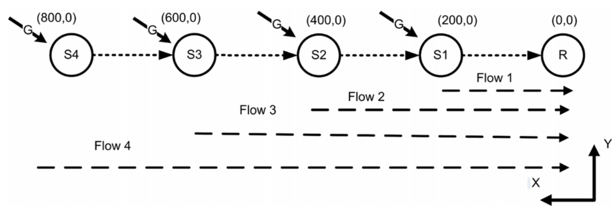

Fig. 9 A five-node chain topology.

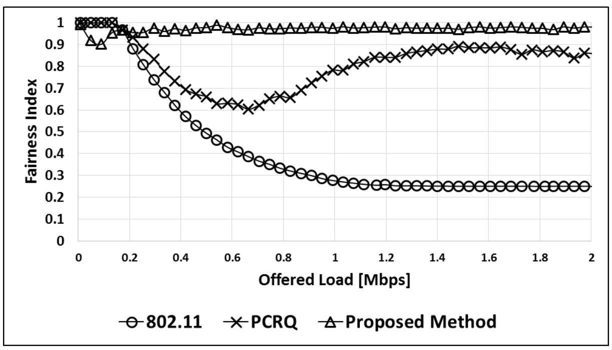

Fig. 10 Fairness index—Five-node chain topology.

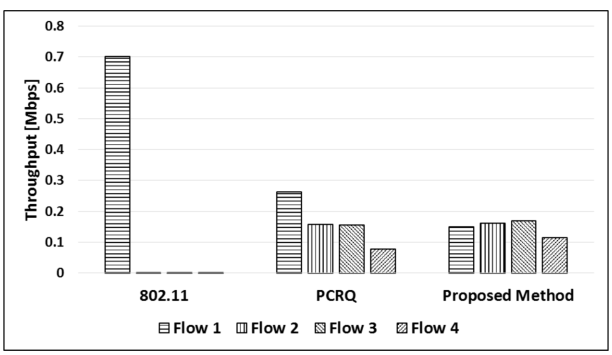

Fig. 11 Per-flow throughput—Five-node chain topology.

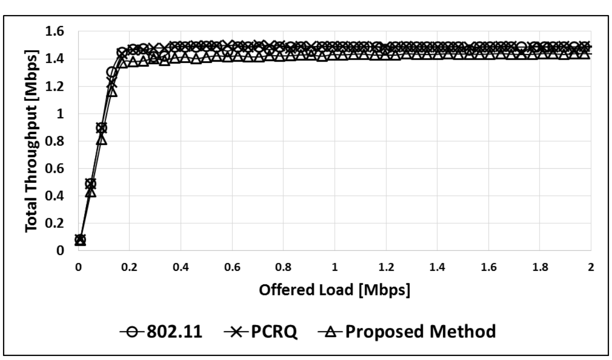

Fig. 12 Total throughput—Five-node chain topology.

flows is not so large, then PCRQ results in a medium level of fairness. It takes time to reflect the change of the state of the MAC layer to the change of the queue length. Therefore, a real time control is difficult by queue length.

When the offered load is larger than 0.2 in Fig. 10, our

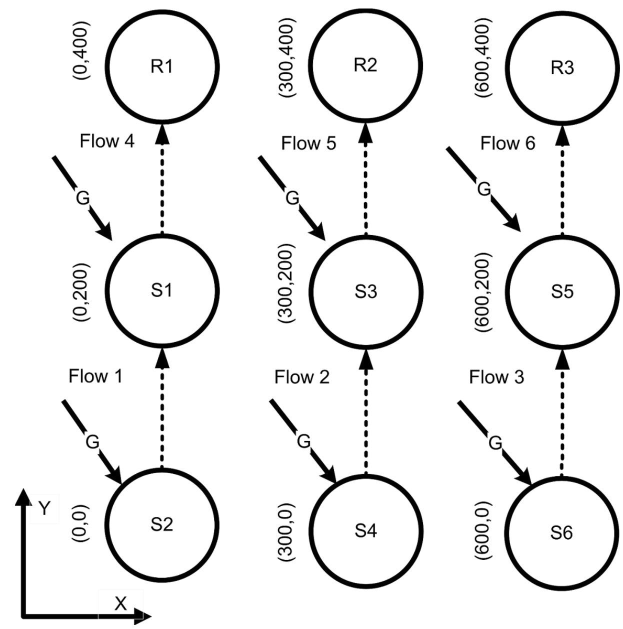

Fig. 13 A grid topology.

proposed method achieves good fairness. This result comes from a good combination of Algorithm 1 and Algorithm 2. Algorithm 1 contributes to the fairness among nodes by channel utilization adjustment. In addition, in Algorithm 2, we use packet dequeue rate that can reflect the MAC layer state more quickly and more accurately than queue length.

Fig. 11 shows that our proposed method achieves better per-flow fairness than the others. In Fig. 12, the total throughput in our proposed method is slightly smaller than the others. The reason for this is because our algorithm gives some delay time which causes a slight reduction in bandwidth efficiency.

4.3 Scenario 3

Scenario 3 in Fig. 13 is a combination of three-pair topology and chain topology. This topology is called a grid topology. In this topology, the distance between columns is 300[m] which is larger than the radius of transmission range 250[m] but smaller than the radius of carrier sensing range 550[m]. Nodes S1-S6 generate the same offered load G. This scenario has both MAC layer problems (caused by both three-pair topology and multi-hop topology) and link layer problem (caused by multi-hop topology).

The performance results are shown in Fig. 14, Fig. 15 and Fig. 16. As shown in Fig. 14, the fairness index in the case of 802.11 is worst due to unfairness at both MAC and link layers. The fairness index is slightly improved by PCRQ, yet, PCRQ cannot solve the unfairness problem caused by three-pair topology. The proposed method achieves better fairness index than PCRQ and 802.11 standard. In Fig. 14, we also show the result of “Algorithm 2 Only”, which means only Algorithm 2 is used and Algorithm 1 is not used. We can see from Fig. 14 that the fairness index of our proposed

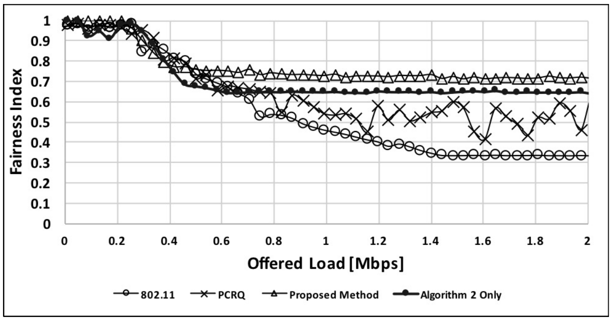

Fig. 14 Fairness index—Grid topology.

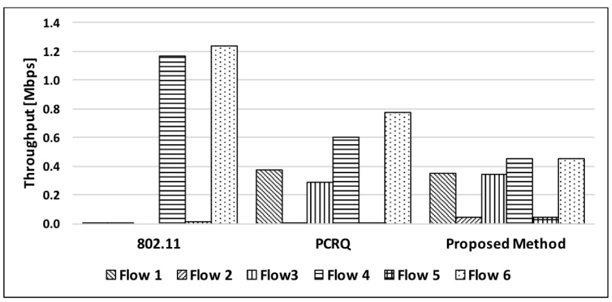

Fig. 15 Per-flow throughput—Grid topology.

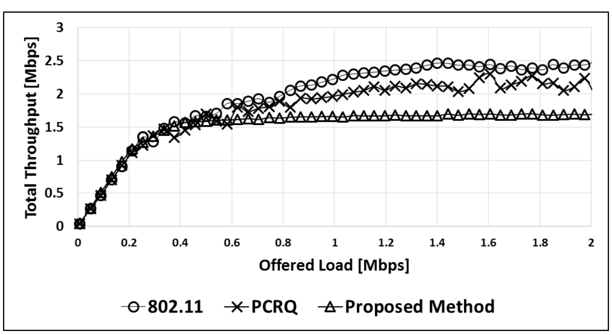

Fig. 16 Total throughput—Grid topology.

method is better than "Algorithm 2 Only". Scenario 3 has both MAC and link layer problems, then Algorithm 1 works effectively for solving the MAC layer problem and Algorithm 2 works for the link layer problem. This result proves the good combination of Algorithms 1 and 2. In addition, we show in Fig. 15 the per-flow throughput result at saturation state when the offered load G equals the effective bandwidth (1.4[Mbps]), which shows the fairness improvement of the proposed method. The total throughput results are shown in Fig. 16. From this figure, we can see that the throughput in the proposed method slightly decreases. This is because of the trade-off between fairness and throughput that was discussed at the end of 4.1 as well as the delay time caused by Algorithm 1.

4.4 Scenario 4

In scenario 4, we consider a topology in Fig. 17 where both

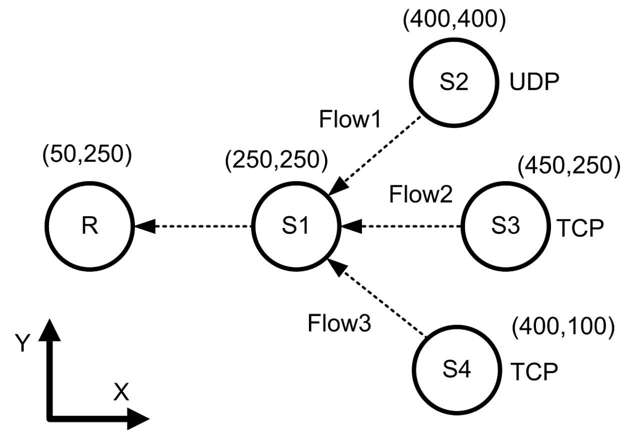

Fig. 17 Scenario 4.

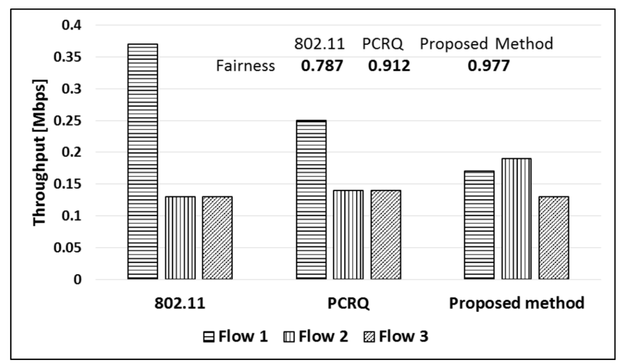

Fig. 18 Throughput and fairness results—Scenario 4.

UDP and TCP flows co-exist. In this topology, one UDP flow from S2 and two TCP flows from S3 and S4 contend for channel access. The UDP flow's offered load is set equal to the channel bandwidth 2[Mbps]. The performance results are shown in Fig. 18. Flow 2 is the central flow and hence less advantageous than Flow 1, but Flows 2 and 3 are TCP flows, so packet transmission control is done by TCP, then the result of 802.11 in Fig. 18 shows that Flows 2 and 3 got the same throughputs. Our proposed method results in a better fairness index than PCRQ and 802.11 standard, and the total throughput of our method is almost equal to the others. In other words, our proposed method is effective also for the case of mixture of UDP and TCP flows.

5. Conclusion

In this paper, we proposed a method to solve the unfairness problem in wireless ad hoc networks by using only local information at the link layer of each node. Our method consists of two algorithms. Algorithm 1 is effective for the MAC layer fairness, and Algorithm 2 is effective for the link layer fairness. The effectiveness of our proposed method was verified by various simulation results.

6. Future Work

In this paper, we investigated the unfairness problem which occurs at the MAC and link layers. However, the same problem occurs at other layers like transport layer [19], [20]. Therefore, it is necessary to study the unfairness problem at those layers to build a good algorithm.

References

[1] N.M. Tuan, K. Nakagawa, and K. Watabe, "A method at link layer to improve the fairness in multi-hop wireless ad hoc networks," SoICT 2015, pp.135–142, Dec. 2015.
[2] M. Conti and S. Giordano, "Mobile ad hoc networking: Milestones, challenges, and new research directions," IEEE Commun. Mag., vol.52, no.1, pp.85–96, Jan. 2014.
[3] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., vol.1, pp.23–34, March 2005.
[4] Z. Li, S. Nandi, and A.K. Gupta, "ECS: An enhanced carrier sensing mechanism for wireless ad hoc networks," Comput. Commun., vol.28, no.17, pp.1970–1984, Oct. 2005.
[5] I.S. Department, IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.
[6] C. Chaudet, D. Dhoutaut, and I.G. Lassous, "Performance issues with IEEE 802.11 in ad hoc networking," IEEE Commun. Mag., vol.43, no.7, pp.110–116, July 2005.
[7] M. Garetto, T. Salonidis, and E.W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA Multi-hop wireless networks," IEEE/ACM Trans. Netw., vol.16, no.4, pp.864–877, Aug. 2008.
[8] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," Proceedings of the 6th annual international conference on Mobile computing and networking, pp.97–98, Aug. 2000.
[9] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," Ad Hoc Networks, vol.1, no.1, pp.107–123, July 2003.
[10] 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.
[11] S. Chakraborty, P. Swain, and S. Nandi, "Proportional fairness in MAC layer channel access of IEEE 802.11s EDCA based wireless mesh networks," Ad Hoc Networks, vol.11, no.1, pp.570–584, Jan. 2013.
[12] T. Razafindralambo and I. Guérin-Lassous, "Increasing fairness and efficiency using the MadMac protocol in ad hoc networks," Ad Hoc Networks, vol.6, no.3, pp.408–423, May 2008.
[13] D. Nandiraju, N. Nandiraju, and D.P. Agrawal, "Service differentiation in IEEE 802.11s mesh networks: A dual queue strategy," MILCOM 2007-IEEE Military Communications Conference, pp.67–72, Oct. 2007.
[14] O. Shagdar, K. Nakagawa, and B. Zhang, "Achieving per-flow fairness in wireless ad hoc networks," Electronics and Communications in Japan (Part I: Communications), vol.89, no.8, pp.37–49, Aug. 2006.
[15] J. Jun and M.L. Sichitiu, "Fairness and QoS in multihop wireless networks," IEEE Vehicular Technology Conference, vol.5, pp.2936–2940, Oct. 2003.
[16] P. Giang and K. Nakagawa, "Achieving fairness over 802.11 multihop wireless ad hoc networks," IEICE Trans. Commun., vol.E92-B, no.8, pp.2628–2637, Aug. 2009.
[17] The Network Simulator: ns-2. http://www.isi.edu/nsnam/ns/
[18] R. Jain, D.-M. Chiu, and W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, vol.38, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984.
[19] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen, "TCP performance issues over wireless links," IEEE Commun. Mag., vol.39, no.4, pp.52–58, April 2001.
[20] A.M. Al-Jubari, M. Othman, B.M. Ali, and N.A.W.A. Hamid, "TCP performance in multi-hop wireless ad hoc networks: Challenges and solution," EURASIP Journal on Wireless Communications and Networking 2011, no.1, pp.1–25, Dec. 2011.

Nguyen Minh Tuan received his B.E. degree from Hanoi University of Science and Technology, Vietnam, in 2012. He worked for Samsung Electronics Vietnam from 2012 to 2014. He achieved his M.E. degree from Nagaoka University of Technology in 2016. His research interests are wireless ad hoc network, network architecture and routing protocols.

Kohei Watabe received his B.E. and M.E. degrees in Engineering from Tokyo Metropolitan University, Tokyo, Japan, in 2009 and 2011, respectively. He also received the Ph.D. degree from Osaka University, Japan, in 2014. He was a JSPS research fellow (DC2) from April 2012 to March 2014. He has been an Assistant Professor of Department of Electrical Engineering, Nagaoka University of Technology since April 2014. He is a member of the IEEE and the IEICE.

Pham Thanh Giang received his B.E. degree from Hanoi University of Technology, Vietnam, in 2002. He achieved M.E. and D.E. degrees from Nagaoka University of Technology, Japan, in 2007 and 2010, respectively. He is currently a researcher at Institute of Information Technology, Vietnam Academy of Science and Technology. His interests are security, mobile ad hoc networks, the Internet architecture in mobile environments and Internet traffic measurement.

Kenji Nakagawa received the B.S., M.S. and D.S. degrees from Tokyo Institute of Technology, Tokyo, Japan, 1980, 1982 and 1986, respectively. In 1985, he joined NTT (Nippon Telegraph and Telephone Corp.). In 1992, he became an associate professor of Nagaoka University of Technology, and since 2012 he has been a professor. He received 2012 IEICE Paper Award. His research interests include performance evaluation of network, queueing theory, and information geometry. He is a member of IEEE, Operations Research Society of Japan, Mathematical Society of Japan and Japanese Society of Engineering Education.