社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

マルチホップ無線アドホックネットワークにおけるスループットとその公平性の改善に関する研究

グェン　ミン　トゥアン†　中川　健治†　渡部　康平†

† 長岡技術科学大学 大学院 工学研究科　〒 940–2188 新潟県 長岡市 上富岡町 1603–1

あらまし　マルチホップ無線アドホックネットワークにおいて，IEEE802.11 プロトコルではネットワーク特性，特に，スループットや公平性などの特性の低下が避けられない。ネットワークの負荷が大きくなり，システムが飽和状態に達すると，長距離フローはスループットが大きく低下する。これらの問題は，MAC 層における媒体の競合に起因するものとリンク層におけるキューの競合に起因するものがある。本論文では，リンク層において各フロー間を公平に扱う方法を与え，スループットの低下や不公平性の問題を解決する方法を提案する。提案手法ではリンク層でラウンドロビンキューを使用する。パケットエンキューの平均間隔の推定値を使用して，公平なスケジューリングアルゴリズムを与える。提案法によって MAC 層およびリンク層の両方で不公平問題を軽減することができる。いくつかのシミュレーション結果によって，提案法は標準的な IEEE802.11 と比較してより良好なスループットと公平性を達成できることを示す。

キーワード　公平性, マルチホップ無線

Improvement for Throughput and Fairness in Multi-hop Wireless Ad Hoc Networks

Nguyen MINH TUAN†, Kenji NAKAGAWA†, and Kohei WATABE†

† Graduate School of Engineering, Nagaoka University of Technology
Kamitomiokamachi 1603–1, Nagaoka, Niigata, 940-2188 Japan

Abstract　Using the protocol IEEE 802.11, multi-hop wireless ad hoc networks yield only a poor performance, especially in throughput and fairness. When the offered load becomes large, i.e. the system is in saturation state, long-distance flows suffer a high degree of throughput deterioration. These problems not only come from medium contention at the MAC layer but are due to the link layer. In this paper, we propose a method to solve the throughput degradation and unfairness problem by providing fair treatment between flows at the link-layer. In our proposed method, a fair scheduling algorithm using round robin queue and the estimation for the average interval of packet enqueueing is applied to alleviate the unfairness problem at both MAC and link layer. The simulation results reveal that our proposed method is able to achieve better throughput and fairness compared to the standard IEEE 802.11.

Key words　fairness, multihop wireless

1. Introduction

An ad hoc wireless network is an infrastructureless wireless network that consists of a set of wireless stations. Each station is able to play a role of both a router and an endpoint, forming a multi-hop wireless ad hoc network. Some attractive features of wireless adhoc networks such as self-organisation, flexibility and simple deployment offer a wide array of applications.

In wireless ad hoc network, the IEEE 802.11 [9] protocol, in which Distribution Coordinated Function (DCF) performs a vital role, is de facto standard used at the MAC layer. DCF prevents channel from colliding, endeavors the fairness in sharing channel resource between stations, and facilitates asynchronous data transfer. However, IEEE 802.11 protocol is known to have throughput unfairness problem, both short-term and long-term fairness [10]. In particular, in a multi-

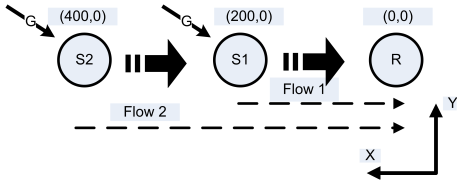

Fig. 1: A basic multihop network model

hop ad hoc network where direct flows and forwarding flows coexist, the throughput of many hop flows decrease dramatically. To illustrate this, we carried out an experiment with a simple topology as Fig. 1 by network simulator 2 (NS-2) [6]. The channel bandwidth is set to 2 Mbps. Each station generates a 1.6 Mbps offered load with 1∼Kb payload per packet, enough to make the network in the saturation state, i.e. each node always has a packet to transmit. The result discloses that the flow 1, or direct flow has a throughput of about 1.25 Mbps while the throughput of flow 2, the forwarding flow, just reaches at 0.0013 Mbps.

There are many factors causing this unfairness problem [11]. At the MAC layer, one of well-known problems is hidden terminal problem. Although, using RTS/CTS handshake can greatly reduce the effect of hidden terminal problem, this problem can not be solved completely. As shown in the Fig. 1, node R is always ready to receive a packet, i.e. it will immediately reply a CTS message as soon as a RTS frame comes. Assume that node S1 and node S2 would like to transmit a packet, then they transmit a RTS frame to node R and S1 respectively. Node R can immediately answer with a CTS frame while node S1 cannot because of its busy state. Finally, node S1 succeeds in transmitting its packet and reset its CW back to its minimum value. Thus, it is clear that node S1 gains more advantages in the next channel contention. In addition, the standard protocol cannot deal with the exposed node problem [15] which has injurious effects in a multi-hop network. Another problem is the large-EIFS problem [8].

At the link layer, Nandiraju et al. [5] proves that the link layer buffer management plays an important role for achieving good throughput and fairness. In multi-hop wireless network, the default queue management is FIFO queue. At an intermediate node, the queue at the link layer is responsible for both packets of forwarding flows from other sources and packets of the direct flow from its-own. Unfortunately, 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 a direct flow may occupy almost all the buffer space, then packets of forwarding flows are dropped due to buffer overflow.

In this paper, based on Probabilistic Control on Round robin Queue (PCRQ) [1], we propose a queue management scheme for multi-hop wireless networks at the link layer to solve the unfairness problem and throughput degradation. The algorithm is formed of Round Robin queue mechanism in combination with an estimation of the packet enqueue. According to the packet enqueuing rate of each flow, an arrival packet can be enqueued or dropped. In addition, if the reading turn comes to an empty queue, we will make a delay time calculated by the packets enqueue interval of that queue. This facilitates the fair serving between flows at link layer and a time delay at an advantageous station will create a chance for other stations to obtain the channel and transmit its packets. Our scheme requires no modifications to the MAC layer, thus in the practical, there is no need to do hardware modification.

The rest of the paper is organised as follows. In section 2 and 3, we briefly present some related works including PCRQ. Our proposed method is given in section 4. The numerical results are shown in section 5. Finally, section 6 commits a short conclusion.

2. Related Work

In reference [12], Nandiraju et al. suggested a fair scheduling algorithm called Dual Queue Service Differentiation(DQSD). DQSD uses two separate queues for forward flow and direct flows and manages to provide fair service to these two queues. However, since packets from all forwarding flows are put into one queue, it is not ensure the justice for packets from large distance. Shagdar et. al. [13] and Izumikawa et. al. [14] also investigated the unfairness of direct flow and forwarding flows. They proposed a scheduling algorithm by using Round Robin scheduling. Nevertheless, their proposal considers an ideal MAC layer that is difficult to obtain. Giang [1] shows that only RR mechanism cannot result in a per-flow fairness.

3. PCRQ

Giang [1] researched the unfairness problem at both MAC layer and link layer in multi-hop networks. He introduced a method called probabilistic control on round robin queue (PCRQ) to lessen the unfairness problem and improve the network performance. PCRQ is composed by three algorithms at the link layer.

The first algorithm is used for controlling the number of input packets to the queue. An arrival packet is put to the queue with the probability calculated by

$$P_{i\_input} = \begin{cases} 1 & \text{if } qlen_i \leqq ave \\ 1 - \alpha\frac{qlen_i - ave}{(n-1)ave} & \text{if } qlen_i > ave \end{cases}$$

where α 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 the $qlen_i$, $i = 1, \ldots, n$.

The second algorithm is used for controlling the turn of reading queues by giving waiting time δ for the empty queue. The queue's turn of a flow i is kept at the following probability

$$P_{i\_turn} = \begin{cases} \beta\frac{n \times ave}{qmax} & \text{if } qlen_i = 0 \\ 0 & \text{if } qlen_i > 0 \end{cases}$$

where β is a constant, $qmax$ is the maximum queue lengh.

The third algorithm is used to decrease the number of packets from heavy offered load flows to MAC layer. The idea is controlling the number of output packets from RR queues. A packet at the head of the queue of flow i is sent to MAC at probability calculated by

$$P_{i\_output} = \begin{cases} 1 & \text{if } qlen_i \leqq \text{ave} \\ 1 - \gamma \frac{qlen_i - ave}{(n-1)ave} & \text{if } qlen_i > \text{ave} \end{cases}$$

where γ is a constant.

By three above algorithms, PCRQ operates for discarding packets before entering to a queue or giving delay to packets in the queue basing on some probability. Hence, PCRQ results in an equal packet distribution between flows. However, within a short-time scale, PCRQ does not always provide a fairness of short-term throughput. In addition, the determination of values α, β, γ in PCRQ are not easy and require much experimental effort.

4. Proposed Method

Inheriting the basic idea of PCRQ in [1], we offer a modification at link layer to cope with the poor performance in term of fairness in multi-hop wireless network. Our method consists of two algorithms: the algorithm 1 controls queue input and and the algorithm 2 controls the queue reading . The details are as follows.

The time interval between the two consecutive enqueued packets at a queue of a flow i, called packet enqueue interval σ_i, is calculated as below.

Each time a packet is enqueued to the queue i, calculate

$$\sigma_i = \begin{cases} \sigma & \text{if } k=1 \\ \dfrac{t_i(k) - t_i(1)}{k-1} & \text{if } k \geqq 2 \end{cases} \tag{1}$$

where σ is a pre-defined value, k is the number of packets, $t_i(k)$ indicates the time that a $k^{th} - packet$ is enqueued. Then, *average packet enqueue interval* $\bar{\sigma}$ is calculated as

$$\bar{\sigma} = \frac{\sum_{i=1}^{N} \sigma_i}{N} \tag{2}$$

where N is the number of queues, σ_i is the value calculated by (1).

4.1 Algorithm 1 (modified)

As analysis, in a multi-hop wireless ad hoc network, packets from the direct flow come to the queue much faster, and thus the queue of the direct flow has a higher enqueue rate. Algorithm 1 is proposed to prevent this problem. It takes a decision on enqueueing or dropping an arrived packet. Each time a packet arrives to a queue, if the packet enqueue interval of the queue is less than the average packet enqueue interval between all queues, it is likely that this packet will be dropped.

By this algorithm, packets of a flow with high arrival rate, i.e. heavy offered load are likely to be dropped. This algorithm keeps the balance between queues. Hence, the queue length of each flow is kept equivalently small.

Algorithm 1 Packet Drop Decision

```
when a packet arrives
check source_node i of this packet
if there is no queue for this source_node then
  create a new queue;
  enqueue the packet;
else
  calculate average packet enqueue interval σ̄ by (2)
  if σ_i < σ̄ − η (η is a constant that approximates to 0) then
    drop the packet;
  else
    enqueue the packet to the queue;
  end if
end if
```

4.2 Algorithm 2 (modified)

Unlike packets from the direct flow that can be enqueued immediately when generating, packets from forwarding flows need sometime to reach at the intermediate node. Hence, usually, the queues of forwarding flows are empty. This causes the unfairness when reading queue. Algorithm 2 is proposed to deal with this problem. It gives a delay time in reading queue if the queue is empty. If the reading pointer points to an empty queue, we will give a delay time for packet traversal from the remote node. However, different from PCRQ where in an empty queue, a delay action is determined by a probability based on the queue length, and the period of delay time is fixed in a δ time, our new algorithm tends to calculate how much time the reading pointer should wait when pointing to an empty queue.

Algorithm 2 Delay Time Control

```
Each time reading pointer points to a queue i
if this queue is empty then
  give a delay σ_i, σ_i is the value calculated by (1);
  read the queue;
  if this queue is still empty then
    skip to the queue (i + 1);
  else
    read a packet from queue i;
    go to the queue (i + 1);
  end if
else
  read a packet from queue i;
  go to the queue (i + 1);
end if
```

5. Performance Evaluation

In this section, we compare our proposed method with PCRQ [1] and FIFO performed by NS-2 version 2.35 [6]. NS-2 simulation parameters are set as the table 1. In the simulation, the raw bandwidth of the channel is set to 2 Mbps, causing the maximum throughput about 1.4 Mbps due to

Table 1 NS-2 parameters configuration

Channel data rate	2 Mbps
Antenna type	Omni direction
Radio Propagation	Two-ray ground
Transmission range	250 [m]
MAC protocol	IEEE 802.11b
	(RTS/CTS mechanism is enable)
Routing protocol	DSDV
Connection type	UDP/CBR
Queue type	FIFO, PCRQ, proposal
Maximum Queue length	100 [packets]
Packet 's size	1024 Bytes
Simulation time	75s

the overhead in IEEE 802.11 [8]. For each flow, the packet interval is set to 200 packets per second, leading the offered load about 1.6 Mbps, so large that a single flow can occupy all the channel and the unfairness occurs when the network is in a saturation state. The initial values σ and η are respectively set to 0.02 and 0.01 after extensive simulations.

The metrics using to evaluate are throughput and fairness index. In term of throughput, we consider aggregate throughput, short-term throughput and average throughput of each flow. In the aspect of fairness, the short-term fairness [4] is examined. Actually, in the context of communication networks, there are two kinds of fairness, namely, *short term fairness and long term fairness.* Long term fairness implies fairness over long time scales while short term fairness focuses on the fairness evaluation of shared channel over short time scales [4]. Short-term fairness has a great impact on the performance of applications especially real-time applications [3]. Additionally, short-term fairness may suggest long-term fairness but not vice versa. Short-term fairness can be calculated by sliding window method (SWM), as given in [4], over an arrival packets stream as the illustration figure SWM begins from a packet trace and slide a

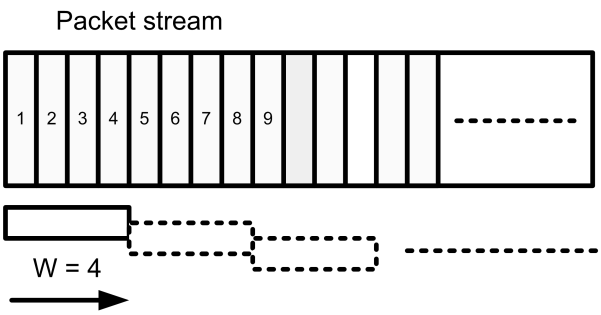

Fig. 2: Sliding window method for calculating short-term fairness with the window size equals to 4

given window size w across it. Initially, the left edge of the window is matched with the left edge of the packet arrival stream, i.e. the first packet. Then, we determine the packet ratios of packet arrivals from each flow i within this window boundary. Next, based on these ratios, we will compute Jain fairness index [3] for this window by this formula

$$\mathrm{FJ}(w) = \frac{(\sum_{i=1}^{n} x_i(w))^2}{n\sum_{i=1}^{n}(x_i(w))^2} \quad (3)$$

where n is the number of flows, $x_i(w)$ is the ratio of the number of packets from a flow i per window w. The fairness index ranges from $1/n$ for the worst case to 1 for the best case. We then step by step slide the window to the right side and calculate the appropriate fairness index to each window. Finally, the short-term fairness index for this given window size is computed by the average of all acquired fairness indices. Additionally, if the window size is large $w = \infty$, the short-term fairness will focus to the long-term fairness.

5.1 Scenario 1

Scenario 1 is a typical 3-node chain topology as Fig. 1. Two stations 1 and 2 generate UDP packets to the destination R. The fairness index and throughput comparison are shown in Fig. 3, Fig. 4a, Fig. 4b, Fig. 4c, and Fig. 5.

It can be clearly seen from the Fig. 3 that with our proposed

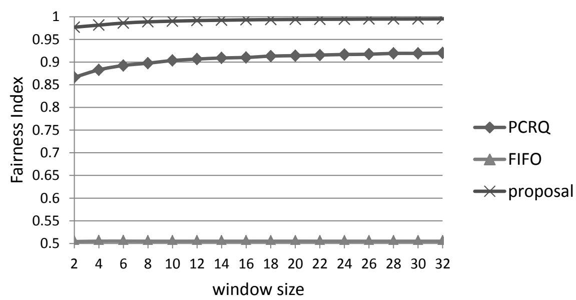

Fig. 3: Fairness Index

method, the fairness index is greatly improved compared to the case of FIFO. It is also considerably higher than PCRQ especially with a small window size. Our fairness result is also confirmed by the short-term throughput shown in the Fig. 4a, Fig. 4b and Fig. 4c. In our proposed method, the end-to-end throughputs are fairer and more stable than others, resulting in a better fairness index. The effect of our method can be explained as follows. With FIFO queue, as analysis in the Section 1, the direct flow tends to occupy almost all the buffer, this leads to packets from other flows are easily dropped due to buffer flow. Also, the position of the station 1 enables much more advantage in channel contention. Compared to FIFO, firstly, our proposed method provides a fair share between flows by round robin mechanism to solve the unfairness at the link layer. Secondly, by controlling packet input and delay using packet interval time information of each queue, our method can deal with the unfairness at the MAC layer. Compared to the PCRQ, this idea is based on the packet rate instead of queue length. The packet rate is more closely related to the lower layer. It is the direct measument that MAC layer notifies to the link layer while the queue length is indirect measurement and may be affected by other factors at the link layer. Therefore, this metric reflects the network state more exactly and can give quickly response with the contemporary network state.

The per-flow throughputs are shown in the Fig. 5 with a notice that Th1 and Th2 indicate end to end throughputs from station S1 and S2 to station R, respectively. From this chart, the aggregate throughput in the case of our proposed method is slightly degraded compared to that in the case PCRQ and FIFO. Actually, there is a trade-off between fairness and aggregate throughput in this type of topology. To explain about it, assume that the bandwidth of the channel is B. Then, if we could achieve the perfect fairness, 1 in particular, the per-flow throughput will equal to B/3. This is because the packet from the station S2 to station R need to be forwarded by the station S1. As a result, the aggregate of per-flow throughput will be 2B/3. In contrast, when the fairness is worst, 0.25 in particular, the throughput Th1 can be up to B because of its advantageous position while throughput Th2 is zero. Clearly, in this case, the aggregate can go up to B.

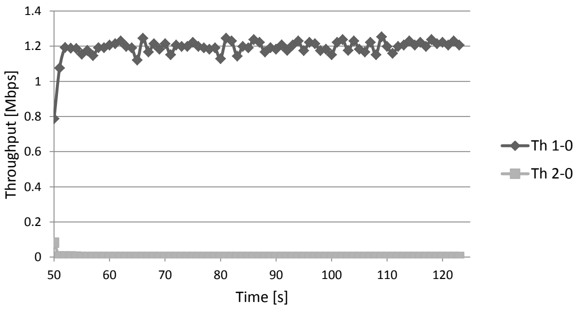

Fig. 4a: FIFO - short-term throughput

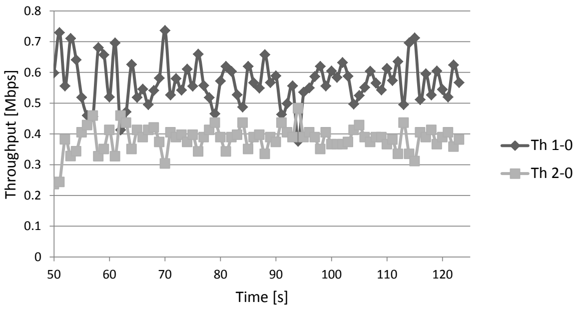

Fig. 4b: PCRQ - short-term throughput

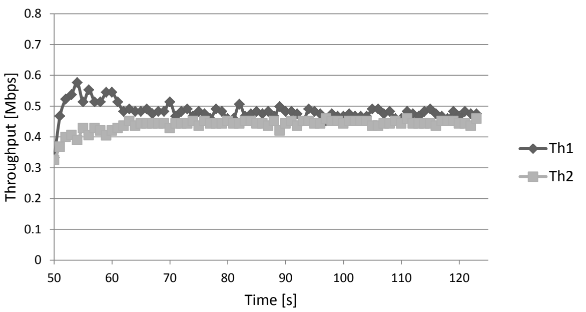

Fig. 4c: proposal - short-term throughput

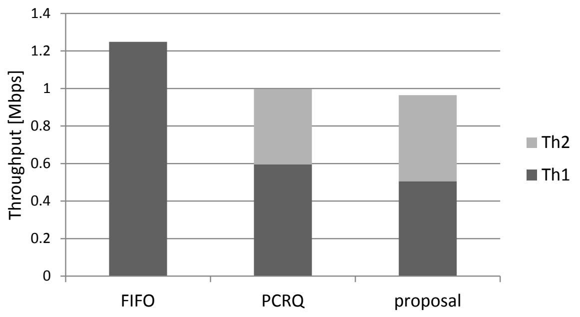

Fig. 5: Throughput scenario 1

5.2 Scenario 2

Scenario 2 is a typical 5-node chain topology as Fig. 6. The fairness index and throughput comparison are shown in Fig. 7, Fig. 8, Fig. 9a, Fig. 9b, and Fig. 9c.

From the figures, although in this case, both channel contention and link layer contention is much harder compared to the scenario 1, the fairness index is considerably improved especially in comparison to FIFO. Again, the reason is that our proposed method can face with both contention at both MAC and link layer. Also, using the packet interval time will enable the node to response more quickly with the change of network state. The short-term throughputs comparison exhibits in the Fig. 9a, Fig. 9b, Fig. 9c also prove that our proposed method can lead to a small fluctuation and a minimal difference between flows. The Fig. 8 shows the aggregate throughput comparison between FIFO, PCRQ and our proposed method. Because of the similar reason in the scenario 1, the aggregate throughput in our method is decreased a little. In this scenario, we give one more metric to compare the performance, that is the packet delay time shown in the table 2. In this table, it can be seen that our proposed method gives a small delay time, just slightly higher than PCRQ. In the case of FIFO, the delay time of packets from the flow 4 is small. This is because in FIFO case, the throughput of flow 4 is almost zero, i.e. there are few packets that are transmitted.

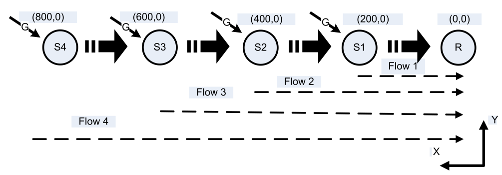

Fig. 6: Network topology for scenario 2

	Flow 1	Flow 2	Flow 3	Flow 4
FIFO	1.13	0.86	0.45	0.45
PCRQ	0.12	0.23	0.29	4.79
proposal	0.25	0.58	0.67	5.06

Table 2: Delay time [s] Scenario 2

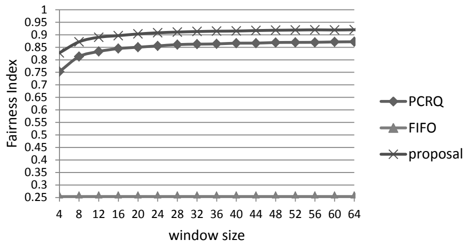

Fig. 7: Fairness Index

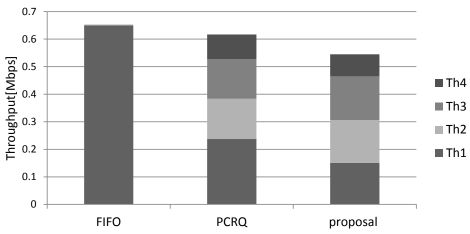

Fig. 8: Throughput scenario 2

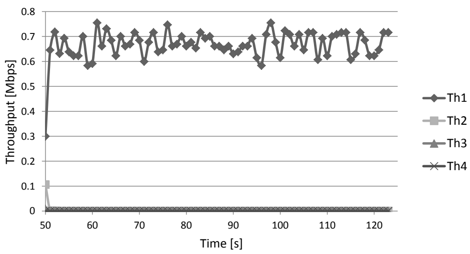

Fig. 9a: FIFO - short-term throughput

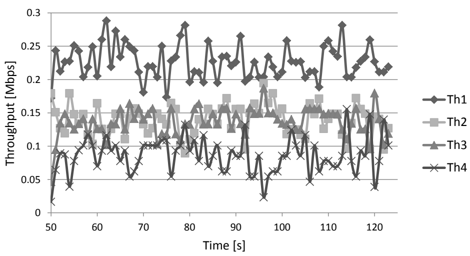

Fig. 9b: PCRQ - short-term throughput

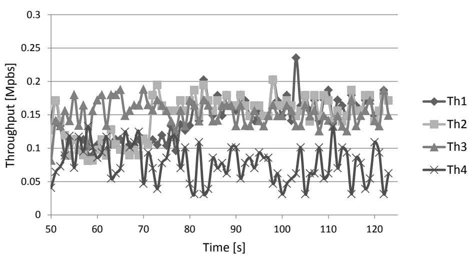

Fig. 9c: proposal - short-term throughput

6. Conclusion

In this paper, we focus on the unfairness problem in multi hops wireless network. The unfairness not only comes from the MAC layer but also from the link layer. Then, we propose a scheduling method based on round robin mechanism and packet interval time to deal with this problem. Our method-solve the unfairness problem at both MAC and link layers. The simulation results show that our proposed method is able to improve the fairness considerably. As a result, the short-term throughputs between flows are kept stable and balance. Thus, it facilitates a better performance especially for real-time applications. Additionally, this method requires only link layer modifications without changing the hardware.

References

[1] P.T. Giang and K. Nakagawa, "Achieving Fairness over 802.11 Multihop Wireless Ad Hoc Networks," IEICE Trans. on Commun., vol.E92-B, no.8, pp.2628-2637, 2009.

[2] R.Jain, D.-M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", Technical Report TR-301, DEC Research Report, 1984

[3] C. Koksal, H.Kassab and H. Balakrishnan, "Ananalysis of Short-Term Fairness in Wireless Media Access Protocols," Proc. of ACM SIGMETRICS, Santa Clara, CA, June 2000.

[4] C.L. Barrett, M.V. Marathe, D.C. Engelhart, A. Sivasubramaniam, "Analyzing the Short-term Fairness of IEEE 802.11 in Wireless Multihop Radio Networks," Proc. of MASCOTS '02, October 2002, Fort Worth, TX, USA, 2002

[5] Nandiraju N, Nandiraju D., "A Novel Queue Management Mechanism for Improving Performance of Multi-hop Flows in IEEE802.11s Based Mesh Networks," Proc. of the 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006). pp.168-173, 2006.

[6] The Network Simulator: NS-2, http://www.isi.edu/nsnam/ns/

[7] Teerawat Issariyakul and Ekram Hossain, "Introduction to Network Simulator NS2," Springer, 2012

[8] Z. Li, S. Nandi, and A. K. Gupta, "Ecs: An Enhanced Carrier Sensing Mechanism for Wireless Ad-hoc Networks," Computer Communication, vol. 28, no. 17, pp. 1970-1984, 2005.

[9] IEEE Standards Department, "IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE Standard 802.11, 2012 Edition, 2012.

[10] T.Nandagopal,T.Kim,X.Goa and V.Brarghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," Proc. of the 6th annual international conference on Mobile computing and networking, p.87-98, August 06-11, 2000, Boston, Massachusetts, USA.

[11] S. Xu and T. Saadawi, "Does the IEEE 802.11 Mac Protocol Work Well in Multihop Wireless Ad Hoc Networks?," Communications Magazine, IEEE, vol. 39, no. 6, pp. 130 137, 2001.

[12] Nandiraju N, Nandiraju D., "Service Differentiation in IEEE802.11s Mesh Networks: A Dual Queue Strategy," Proc. of Milcom (Military Communications Conference). IEEE, Washington, pp. 67-72.

[13] O. Shagdar, K. Nakagawa, and B. Zhang, "Achieving Per-Flow Fairness in Wireless Ad Hoc Networks," Elec. Comm. in Japan, Part 1, Vol. 89, No. 8, pp.37-49, 2006.

[14] J. Jun, M.L. Sichitiu, "Fairness and QoS in Multihop Wireless Networks," Proc. of IEEE VTC, pp.2936 -2940, 2003

[15] A. Jayasuriya, S. Perreau, A. Dadej, S. Gordon, "Hidden vs. Exposed Terminal Problem in Ad Hoc Networks," Proc. of the Australian Telecommunication Networks and Applications Conference, Sydney, Australia, 2004.