Channel Occupancy Time Based TCP Rate Control for IEEE 802.11 DCF

Kenichi Kashibuchi Tarik Taleb Yoshiaki Nemoto Nei Kato
Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi, Japan, 980-8579
Email: buchiken@it.ecei.tohoku.ac.jp

Abstract—**In multi-rate IEEE 802.11 wireless LAN, when multiple stations transmit frames at different data rates, the basic service set (BSS) suffers from "performance anomaly" problem. In this paper, to solve this issue, we propose a TCP rate control scheme for IEEE 802.11 DCF mode. In the proposed scheme, a TCP sender controls its maximum window size based on the throughput estimated at the TCP layer. The throughput estimation is based on the number of active stations and the channel occupancy time used by each station in the BSS, which are monitored at the MAC layer. The proposed scheme forms a cross layer approach involving both MAC and TCP layers. The performance of the proposed scheme over adaptive multi-rate IEEE 802.11 wireless LAN is evaluated and compared with that of standard TCP through several simulations. The simulation results show that the proposed scheme improves the aggregate throughput in the BSS and reduces packet drops at the TCP layer. The proposed scheme also exhibits fairness in terms of channel occupancy time among the competing stations.**

I. Introduction

Nowadays, IEEE 802.11 wireless local area networks (wireless LANs) are gaining popularity and are available at various places. IEEE 802.11 physical (PHY) layer provides multiple data rates. Associated rate adaptation algorithms at the Media Access Control (MAC) layer adapt data transmission rate and enable us to transmit frames at a rate appropriate to current wireless link conditions. However, when stations with a different data rate exist in the basic service set (BSS), the aggregate throughput in the BSS is dramatically degraded. This problem is called performance anomaly. This research work especially focuses on the impact of this issue in the multi-rate wireless LANs on the performance of the Transmission Control Protocol (TCP), which is the most dominant protocol in today's Internet traffic.

In this paper, we propose a scheme that improves the transmission efficiency of TCP in the adaptive multi-rate wireless LAN based on IEEE 802.11 with Distributed Coordination Function (DCF) mode. DCF is a mandatory access mode in the IEEE 802.11 MAC protocol. It is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. With DCF, stations monitor the medium and transmit a data frame if no other station is transmitting. Therefore, at the MAC layer, all stations in the BSS can know the channel occupancy time of each station and the number of active stations. In the proposed scheme, based on these observations, each station periodically estimates how long it can use the medium and calculates the maximum throughput at the TCP layer, while taking into account fairness among active stations. In addition, each station estimates the TCP throughput when the data rate at the MAC layer switches according to the rate adaptation algorithm. A TCP sender then calculates the appropriate window size and accordingly adjusts its maximum window size.

This paper is organized as follows. In Section II, we briefly introduce a rate adaptation algorithm and survey related work on performance anomaly with multi-rate wireless LAN. We then introduce our proposed TCP rate control mechanism for IEEE 802.11 DCF in Section III. In Section IV, we discuss the performance of the proposed scheme in an adaptive multi-rate wireless LAN. Finally, Section V concludes the paper by summarizing the achievements of the proposed algorithm and addressing scopes of some future work.

II. Related Work

The IEEE 802.11 PHY layer supports multiple data rates by employing different modulations and channel coding schemes. At the MAC layer, the underlying rate adaptation algorithm selects one of the data transmission rates according to the varying channel conditions. The higher the data rate is, the higher bit error rate a station suffers from.

The Automatic Rate Fallback (ARF) algorithm [1] is the most primitive and most widely used rate adaptation algorithm. In the ARF algorithm, each sender switches the transmission rate to a particular station after consecutive successes or failures in transmissions to that station. If the sender does not consecutively receive θ_{down} acknowledgement (ACK) frames, it decreases the transmission rate to the next lower data rate and sets a timer (T_{up}). When the sender successfully receives θ_{up} ACK frames consecutively or the timer expires, the sender increases the transmission rate to the next higher data rate and resets the timer. After increasing the rate, if the first transmission at the new rate fails, the sender immediately decreases the rate to the former rate and restarts the timer. In this way, stations can transmit frames at a rate suitable to current wireless channel conditions.

When multiple stations transmit frames at different data rates in a multi-rate wireless LAN, the BSS suffers a problem called performance anomaly [2]. Since the DCF gives approximately equal transmission opportunities to each competing station and the stations transmitting at the lower data rate occupy the medium for a long time, the throughputs of all stations transmitting at the higher data rate decrease. As a result, the aggregate throughput in the BSS is also degraded.

As a remedy to this issue, several approaches have been proposed. In [3], Tan *et al.* discuss the advantage of time-based fairness and propose the time-based regulator (TBR) algorithm based on the leaky bucket scheme. TBR takes into account the channel occupancy time of traffic at the access point (AP). In recent literatures, several schemes achieve airtime fairness by controlling MAC parameters, such as contention window size or frame length [4]–[7]. On the other hand, some solutions for the unfairness issue between up-link and down-link TCP flows seem to be also effective against the performance anomaly problem. In [8], a solution that enables APs to manipulate the receiver's advertised window field in the TCP header is proposed. A major concern with the work is that it cannot be used in a network where the IP security (IPSec) protocol is in use. The Distributed Access Time Control (DATC) [9] scheme regulates the rate of TCP flow by appropriately dropping packets. However, these research works do not consider scenarios where the ARF algorithm is used.

The work in [10] examines the performance of TCP over wireless LAN with ARF. In [11], Choi *et al.* present a protocol interaction between DCF, ARF, and TCP in multi-rate IEEE 802.11 wireless LANs based on Markov chain analysis. In [12], the authors proposed a TCP window size adjustment scheme for the ARF algorithm. However, we found that the proposed scheme often cannot work well with many stations in a BSS. In this paper, we propose a modified algorithm and show that the proposed algorithm can be used as an effective solution to the performance anomaly issue.

III. Proposed TCP rate control for IEEE 802.11 DCF

The proposed scheme defines a cross-layer design that involves the MAC and transport layers. Adequate information is exchanged between the two layers for the sake of TCP window size adjustment.

A. Information used in the proposed scheme

Each station periodically determines an observation slot of length T [s] and monitors the wireless channel to obtain the following information, as illustrated in Fig. 1. T is usually the time required to receive n beacon frames. In case of detecting a change in the number of active stations, T becomes shorter. It should be emphasized that all stations can compose their own observation slots in synchronization with one another. In the remainder of this section, we describe the procedures performed by a station STA$_x$. The following notations are used:

- N: Number of active stations (*i.e.,* stations actually transmitting or receiving frames, not only associating with the AP) in the BSS.
- t_i: Channel occupancy time of station STA$_i$ ($i \in \{1, ..., x, ..., N\}$) [s].
- t_{beacon}: Channel occupancy time used by beacon frames from the AP (*i.e.,* the duration used for receiving beacon frames) [s].
- C_x: Data transmission / reception rate at STA$_x$ [bps].
- M_x: Number of TCP connections connected to/via STA$_x$.
- $\tau_{x,j}$: Channel occupancy time used by the j-th connection of STA$_x$ [s].
- $\overline{\lambda_{x,j}}$: Average length of TCP data segments, transmitted or received by the j-th connection of STA$_x$ [bytes].
- $RTT_{x,j}$: Round trip time (RTT) of the j-th connection traversing STA$_x$ [s].
- b: Number of TCP segments acknowledged by a single TCP ACK (*i.e.*, when the delayed ACK option [13] is used, b is set to two. Otherwise, b is set to one).

The channel occupancy time can be measured by utilizing a duration field, which is generally used for setting a network allocation vector (NAV), in the 802.11 MAC header. More details on how to measure t_i are available in [12]. In the measurements of t_i, we use the average value of the backoff interval ($\overline{t_{\text{backoff}}}$) instead of the actual value.

B. Estimation of the available channel occupancy time

Upon receiving n beacon frames (Event (A) in Fig. 1) or detecting a change in the number of active stations (Events (B) and (C) in Fig. 1) since the start of an observation slot, all the stations calculate the available duration for their own connections based on Eqs. (1)–(9).

The available duration in the BSS is represented as:

$$t_{\text{BSS}}^{\text{available}} = T - t_{\text{beacon}}. \tag{1}$$

The proposed algorithm attempts to evenly allocate this available duration among the N active stations. Therefore, the available duration for STA$_i$ is given by the following equation.

$$\bar{t}_i^{\text{available}} = t_{\text{BSS}}^{\text{available}}/N. \tag{2}$$

On the other hand, BSS may have an unoccupied duration equal to:

$$t_{\text{BSS}}^{\text{remain}} = t_{\text{BSS}}^{\text{available}} - \sum_{i=1}^{N} t_i. \tag{3}$$

The proposed algorithm allocates this remaining duration evenly among the unsatisfied stations. Here, an "unsatisfied station" is a the station that meets the following criterion:

$$t_i \geq r \cdot \bar{t}_i^{\text{available}}, \quad (0 < r < 1), \tag{4}$$

where r is a threshold that defines the satisfaction level of a station. All stations count the number of unsatisfied stations $N_{\text{unsatisfied}}$ based on Eq. (4). The available duration for STA$_x$ is eventually calculated as:

$$t_x^{\text{available}} = \begin{cases} \bar{t}_x^{\text{available}} & \left\{x \,\middle|\, t_x < r \cdot \bar{t}_x^{\text{available}}\right\}. \\ \bar{t}_x^{\text{available}} + \dfrac{t_{\text{BSS}}^{\text{remain}}}{N_{\text{unsatisfied}}} & \left\{x \,\middle|\, t_x \geq r \cdot \bar{t}_x^{\text{available}}\right\}. \end{cases} \tag{5}$$

Each station allocates this available duration among its connections. Assuming that STA$_x$ evenly allocates $t_x^{\text{available}}$ among its M_x connections, the available duration for the j-th connection of STA$_x$ is given by the following equation.

$$\bar{\tau}_{x,j}^{\text{available}} = t_x^{\text{available}}/M_x. \tag{6}$$

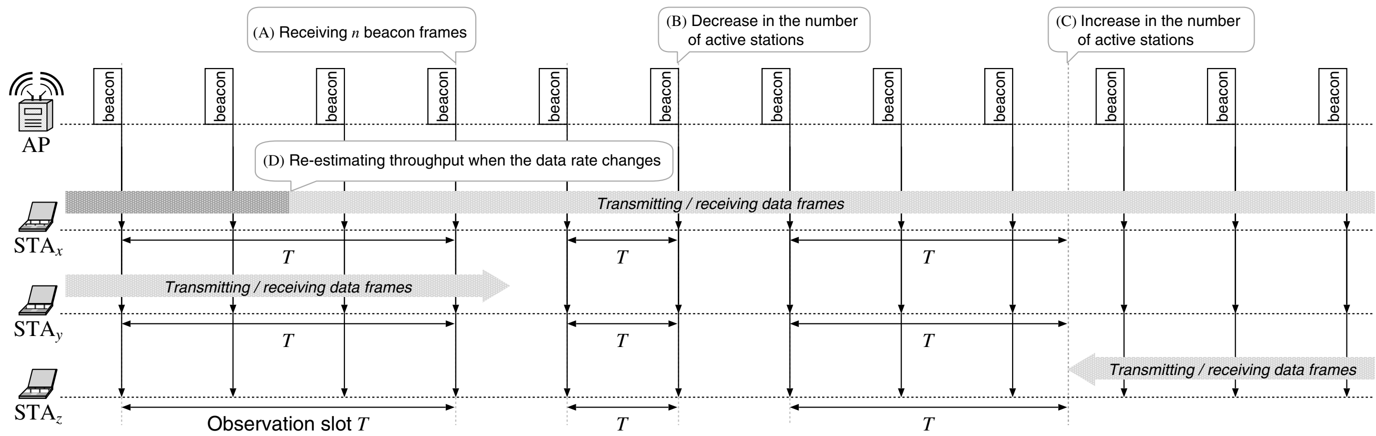


Fig. 1. An illustrative example of how observation slots are determined.

The unused duration in STA$_x$ is given by:

$$t_x^{\text{remain}} = t_x^{\text{available}} - \sum_{j=1}^{M_x} \tau_{x,j}. \tag{7}$$

STA$_x$ allocates this unused duration among the unsatisfied connections, which meet the following criterion.

$$\tau_{x,j} \geq r \cdot \bar{\tau}_{x,j}^{\text{available}}, \quad (0 < r < 1). \tag{8}$$

We suppose that STA$_x$ has $M_{\text{unsatisfied}\,x}$ unsatisfied connections. Finally, the available duration for the j-th connection of STA$_x$ is calculated as follows:

$$\tau_{x,j}^{\text{available}} = \begin{cases} \bar{\tau}_{x,j}^{\text{available}} & \left\{ j \,\middle|\, t_{x,j} < r \cdot \bar{\tau}_{x,j}^{\text{available}} \right\}. \\ \bar{\tau}_{x,j}^{\text{available}} + \dfrac{t_x^{\text{remain}}}{M_{\text{unsatisfied}\,x}} & \left\{ j \,\middle|\, t_{x,j} \geq r \cdot \bar{\tau}_{x,j}^{\text{available}} \right\}. \end{cases} \tag{9}$$

The stations then estimate the maximum throughput at the TCP layer for each connection, as will be described in the next section.

C. Estimation of the maximum throughput

Upon recalculating the available duration $\tau_{x,j}^{\text{available}}$ (Events (A) – (C) in Fig. 1) or changing one's data transmission or reception rate C_x at the MAC layer (Event (D) in Fig. 1) according to the rate adaptation algorithm (*e.g.,* ARF algorithm), a station STA$_x$ estimates the maximum throughput of each connection. It should be noted that, if the station adopts the ARF as a rate adaptation algorithm, the proposed scheme does not perform the following procedures when the station cannot receive the first DATA or ACK frame after increasing the data rate. This is because the ARF algorithm tends to fail in transmitting the first DATA frame after increasing the data rate. It should be also noted that, when switching the data rate, STA$_x$ calculates the maximum throughput using the available duration $\tau_{x,j}^{\text{available}}$ computed in the previous observation slot, as a new observation slot is not determined yet.

The maximum TCP throughput that can be achieved by the j-th connection of STA$_x$ is estimated as:

$$\theta_{x,j} = \frac{\tau_{x,j}^{\text{available}}}{T_{\text{TCP}}\left(C_x, \overline{\lambda_{x,j}}, b\right)} \cdot \overline{\lambda_{x,j}} \cdot b \cdot \frac{1}{T}, \tag{10}$$

where $T_{\text{TCP}}\left(C, \lambda, b\right)$ denotes the time required for transmitting or receiving b number of TCP data segments, each with a size of λ, and one TCP ACK segment at a data rate C. $T_{\text{TCP}}\left(C, \lambda, b\right)$ is given as follows:

$$T_{\text{TCP}}\left(C, \lambda, b\right) = T_{\text{data}}\left(C, \lambda\right) \cdot b + T_{\text{data}}\left(C, 0\right), \tag{11}$$

where $T_{\text{data}}\left(C, \lambda\right)$ is the time required for transmitting or receiving a TCP segment with a size of λ at a data rate C. With the request-to-send / clear-to-send (RTS / CTS) option, $T_{\text{data}}\left(C, \lambda\right)$ is expressed as follows:

$$\begin{aligned} T_{\text{data}}\left(C, \lambda\right) = t_{\text{DIFS}} + \overline{t_{\text{backoff}}} + t_{\text{RTS}} + t_{\text{SIFS}} + t_{\text{CTS}} \\ + t_{\text{SIFS}} + t_{\text{DATA}}(C, \lambda) + t_{\text{SIFS}} + t_{\text{ACK}}(C). \end{aligned} \tag{12}$$

where t_{DIFS} and t_{SIFS} denote the length of DCF inter-frame space (DIFS) and that of short inter-frame space (SIFS), respectively. t_{RTS}, t_{CTS}, $t_{\text{DATA}}(C, \lambda)$, and $t_{\text{ACK}}(C)$ are the duration for transmitting or receiving a RTS frame, a CTS frame, a DATA frame, and an ACK frame, respectively. They all depend on the PHY and MAC layers. More details on this computation are available in [12].

D. Adjustment of the TCP Maximum Window Size

The proposed scheme controls the transmission rate by bounding the TCP maximum window size, wnd_{max}. Since the theoretical maximum TCP throughput is given by (wnd_{max}/RTT), the maximum value of the available window size in the j-th connection of STA$_x$ is calculated as:

$$wnd_{\text{max}_{x,j}} = RTT_{x,j} \cdot \theta_{x,j}. \tag{13}$$

Here, we estimate the round trip time more accurately than in the standard TCP and use its minimum value as $RTT_{x,j}$ for each connection, as in TCP Vegas [14].

1) When the station is the sender: STA_x limits the congestion window size ($cwnd$) in the j-th connection to the calculated $wnd_{\max_{x,j}}$. Obviously, $wnd_{\max_{x,j}}$ should not exceed the sender's buffer size.

2) When the station is the receiver: If the station can obtain RTT of each connection, the STA_x transmits the calculated $wnd_{\max_{x,j}}$ to its corresponding node. In order to do so, the proposed scheme utilizes the receiver's advertised window ($rwnd$) field in the TCP header of ACK segments. Since the TCP sender sets its send window size to the minimum of $cwnd$ and $rwnd$, the maximum send window size is bounded to $wnd_{\max_{x,j}}$. However, in general, it is difficult for TCP receivers to accurately measure RTT. Actually, the station can measure RTT during a three-way handshake, but this may cause unfairness among connections. In this paper, the station transmits the estimated throughput $\theta_{x,j}$ to its corresponding node using $rwnd$ field of ACK segments. $\theta_{x,j}$ [bytes / sec] is scaled (left bit shifted) into 16 bit $rwnd$ field, as in TCP windows scaling option [15]. The corresponding TCP sender then calculates the appropriate window size based on Eq. (13) and limits the congestion window size.

IV. Performance Evaluation

We carried out several simulations using QualNet 4.0.1 [16] and compared the performance of the proposed scheme against that of standard TCP.

Fig. 2 depicts the configuration of the considered network. In our simulations, we model an IEEE 802.11a (5.2 GHz) BSS as an access point serving N stations. N is varied from one to 20. We use the two-ray propagation model and the constant shadowing model without fading. The coverage radius of the cell is about 360 meters when the data rate is 6 Mbps; and is about 36 meters when the data rate is 54 Mbps. The RTS/CTS option is always enabled (*i.e.*, RTS threshold is fixed to zero). All stations roam within the BSS area according to the random way point model with a maximum velocity of 10 m/s and a pause time of 0 sec.

In our scenarios, each station has only a single File Transfer Protocol (FTP) connection. All stations download a file from different FTP servers. End-to-end delays from servers to the AP are the same, in order to avoid the unfairness issue due to variance in RTT. Since all the wired links have more bandwidth than the wireless ones, congestion does not occur in the wired links. Unless otherwise specified, the maximum segment size (MSS) is set to 1460 bytes. The TCP window scaling option and the timestamps option [15] are enabled. The delayed ACK option is disabled. As standard TCP, we use TCP NewReno for comparison. We thus have integrated the proposed scheme into TCP NewReno and IEEE 802.11 MAC. In the proposed scheme, n and r are empirically set to three and 0.8, respectively. The total simulation time is set to 60 seconds. Table I shows an overall list of the simulation parameters. The other parameters are set to the default values in QualNet. All results are an average of multiple simulation runs.

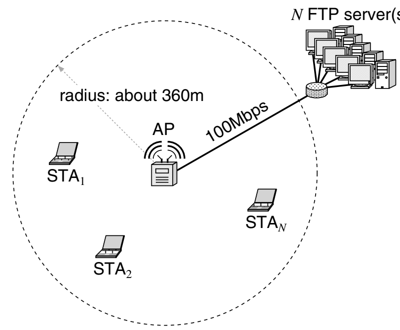


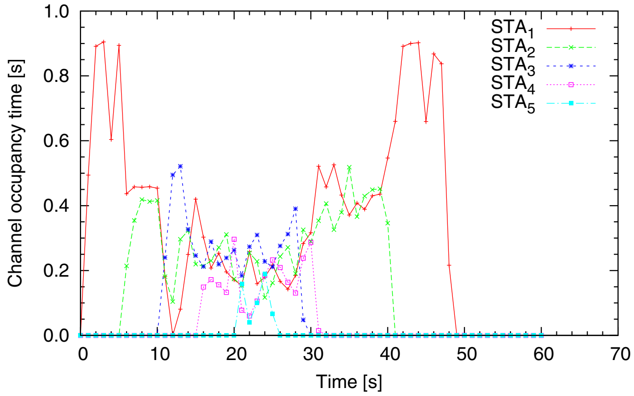
Fig. 2. Simulated network topology.

TABLE I
SIMULATION PARAMETER

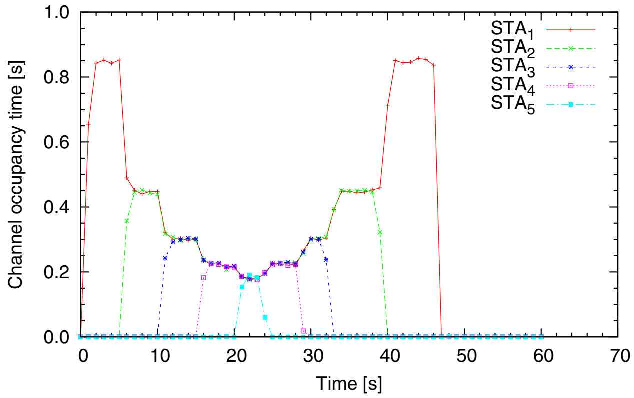
PHY	IEEE 802.11a (5.2 GHz)
Data rate (switched by ARF)	54 / 48 / 36 / 24 / 18 / 12 / 9 / 6 Mbps
ARF parameters (θ_{up} / θ_{down} / T_{up})	10 / 2 / 60 ms
Short / long retry limit	7 / 4
Cell radius (@6 M / @54 M)	*ca.* 360 m / *ca.* 36 m
Wired link bandwidth	100 Mbps
End-to-end RTT	*ca.* 40 ms
Maximum segment size (MSS)	1460 / 1024 / 512 bytes
Buffer size (sender & receiver)	131070 bytes
Velocity of STAs	0 – 10 m/s
Distances from AP to STAs	0 – 360 m
Number of STAs	1 – 20

As a first step, to demonstrate the effectiveness of the proposed scheme, we show an example of the transition of channel occupancy time of each station in Fig. 3. The channel occupancy time is calculated every second. Here, since the channel occupancy time does not include backoff time, its summation is below one second. In this scenario, five stations exist in the BSS ($N = 5$). STA_1, STA_2, STA_3, STA_4, and STA_5 start downloading files of 40, 20, 10, 5, and 1 Mbyte(s) at 0, 5, 10, 15, and 20-th second, respectively. All stations are settled and the distance from the AP to each station is fixed to 10 m. In standard TCP, the used channel occupancy time fluctuates, especially upon increasing the number of stations. On the other hand, the proposed scheme keeps fairness in terms of channel occupancy among stations, and gives relatively smooth transition. Although we omit results due to space limitations, stations achieve fairness in terms of throughput in this case, as they are equidistant from the AP. Moreover, as the proposed scheme exhibits a higher throughput in total, STA_1 downloads the file faster in contrast with the standard TCP. The result indicates that the proposed scheme accurately estimates the throughput at the TCP layer and sets the window size to more appropriate values than standard TCP does.

We now direct our focus to evaluating the performance of the proposed algorithm by using more complex scenarios. Figs. 4 and 5 illustrate the simulation results for varied numbers of stations. Figs. 4(a) and 5(a) plot the fairness in terms of channel occupancy during the entire simulation time

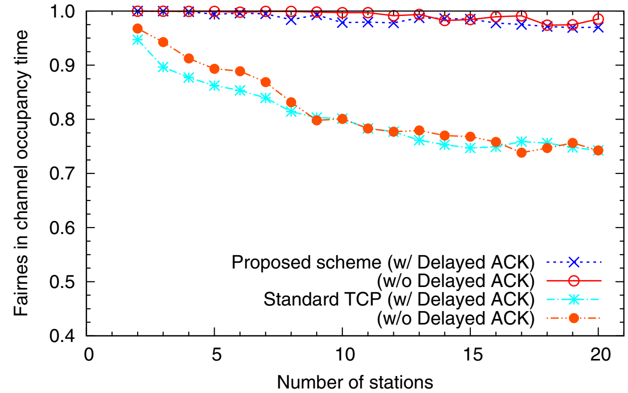


(a) Standard TCP.

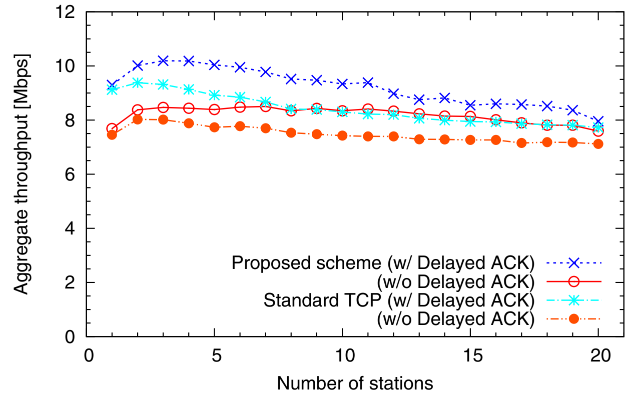


(b) Proposed scheme.

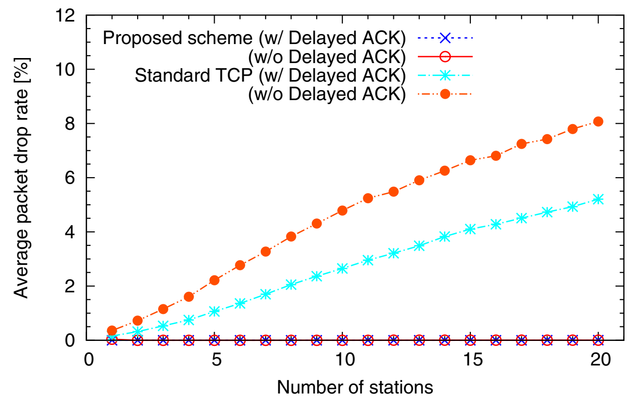
Fig. 3. The transitions of channel occupancy time.



(a) Fairness in channel occupancy time.



(b) Total throughput.



(c) Average packet drop rate.

Fig. 4. Effect of delayed ACK option.

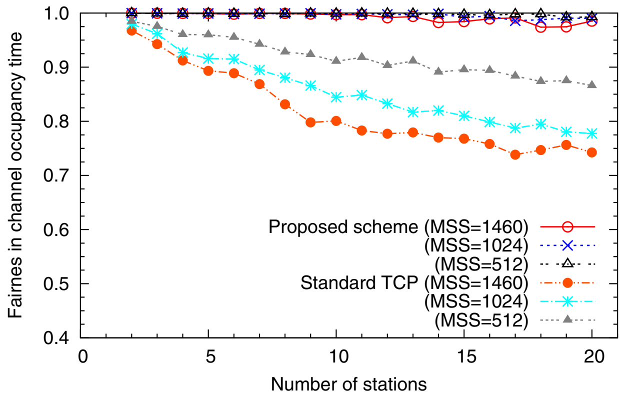
(for 60 seconds). Based on Jain's Fairness index, we use the following metric as the fairness index in channel occupancy time:

$$f_c = \frac{\left(\sum_{i=1}^{N} t_i\right)^2}{N \sum_{i=1}^{N} t_i^2}, \tag{14}$$

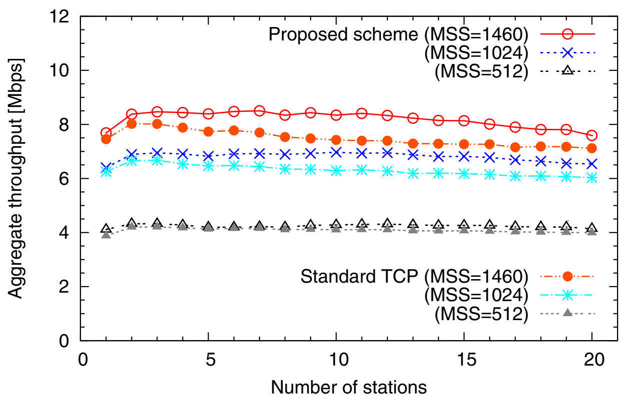
where t_i denotes the channel occupancy time used by STA$_i$. The closer the index is to one, the more fairness the system exhibits in terms of channel occupancy. Figs. 4(b) and 5(b) show the aggregated throughput of all stations in the BSS. Figs. 4(c) and 5(c) represent the average packet drop rate per station. Each packet drop rate is calculated as the percent ratio of dropped TCP data segments to transmitted TCP data segments.

As one can see from Fig. 4, the proposed scheme exhibits better performance than standard TCP does in all simulated scenarios, regardless of the use of delayed ACK option. Given the fact that each station is randomly distributed and randomly roams in the area of BSS, they often use different data rates. As mentioned in Section II, with standard DCF and TCP, this causes performance anomaly. On the other hand, the proposed scheme remedies the issue and exhibits good fairness in terms of channel occupancy time among stations. The better performance of the proposed scheme is due to the fact that a TCP terminal controls its rate based on the number of competing stations and on its own data rate at MAC layer. It accordingly achieves higher total throughput than standard TCP does. While the average packet drop rates in standard TCP increase with the increment of stations, the proposed scheme achieves low drop rates.

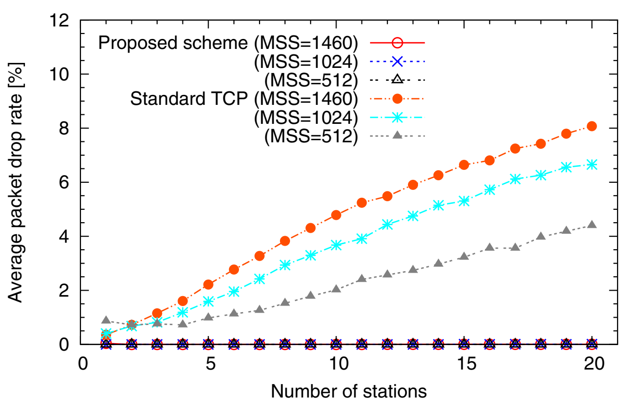
Lastly, in order to examine the impact of frame length on the performance of the proposed scheme, we varied MSS from 1460 bytes to 512 or 1024 bytes. As shown in Fig. 5(b), stations can obtain the highest throughput when MSS is set to 1460 bytes, in both schemes, as the frame length at MAC layer



(a) Fairness in channel occupancy time.



(b) Total throughput.



(c) Average packet drop rate.

Fig. 5. Effect of MSS.

is the longest. For all MSSs, although there are differences in degrees, the proposed scheme represents a better performance in terms of airtime fairness, throughput, and packet drop rate.

V. Conclusion and Future Work

In this paper, we proposed a new TCP window size adjustment algorithm for IEEE 802.11 DCF. The proposed scheme utilizes information on the radio usage monitored at the MAC layer. In the proposed scheme, each station periodically computes the channel occupancy time available for each connection. Finally, the proposed scheme adjusts the maximum window size to the calculated values.

The performance of the proposed scheme was evaluated through several simulations using a simple IEEE 802.11a BSS and was compared with that of the standard TCP. The obtained results demonstrated that the proposed scheme can remedy the performance anomaly problem. The proposed scheme improved the transmission efficiency and the fairness in channel occupancy time among the competing stations. It is also shown that the proposed scheme works well also with TCP delayed ACK option or variant frame lengths.

As a part of our future work, more simulations with connections having different RTTs will be conducted. Comparison with other schemes is also envisioned.

Acknowledgment

This work was sponsored by the Grant-in-Aid for Japan Society for the Promotion of Science (JSPS) Fellows.

References

[1] A. Kamerman and L. Monteban, “WaveLAN®-II: A High-Performance Wireless LAN for the Unlicensed Band,” *Bell Labs Tech. J.*, vol. 2, no. 3, pp. 118–133, summer 1997.

[2] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance Anomaly of 802.11b,” in *Proc. IEEE INFOCOM*, vol. 2, San Francisco, CA, USA, Mar./Apr. 2003, pp. 836–843.

[3] G. Tan and J. Guttag, “Time-based Fairness Improves Performance in Multi-rate WLANs,” in *Proc. USENIX Annual Tech. Conf.*, Berkeley, CA, USA, Jun. 2004, pp. 269–282.

[4] D. Yang, T. Lee, K. Jang, J. Chang, and S. Choi, “Performance Enhancement of Multirate IEEE 802.11 WLANs with Geographically Scattered Stations,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 7, pp. 906–919, Jul. 2006.

[5] A. Banchs, P. Serrano, and H. Oliver, “Proportional fair throughput allocation in multirate IEEE 802.11e wireless LANs,” *Wireless Netw.*, vol. 13, no. 5, pp. 649–662, Oct. 2007.

[6] A. Babu and L. Jacob, “Fairness Analysis of IEEE 802.11 Multirate Wireless LANs,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3073–3088, Sep. 2007.

[7] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, “Airtime Fairness for IEEE 802.11 Multirate Networks,” *IEEE Trans. Mobile Comput.*, vol. 7, no. 4, pp. 513–527, Apr. 2008.

[8] S. Pilosof, S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, “Understanding TCP Fairness over Wireless LAN,” in *Proc. IEEE INFOCOM*, vol. 2, San Francisco, CA, USA, Mar./Apr. 2003, pp. 863–872.

[9] D. Kim, E. Park, and C. Choi, “Distributed Access Time Control for Per-Station Fairness in Infrastructure WLANs,” *IEICE Trans. Commun.*, vol. 89, no. 9, pp. 2572–2579, Sep. 2006.

[10] N. Kim, S. Choi, and H. Yoon, “IEEE 802.11b WLAN Performance with Variable Transmission Rates: In View of High Level Throughput,” in *Proc. ICN*, Reunion Island, France, Apr. 2005, pp. 1080–1087.

[11] J. Choi, K. Park, and C. Kim, “Cross-Layer Analysis of Rate Adaptation, DCF and TCP in Multi-Rate WLANs,” in *Proc. IEEE INFOCOM*, Anchorage, AK, USA, May 2007, pp. 1055–1063.

[12] K. Kashibuchi, T. Taleb, A. Jamalipour, Y. Nemoto, and N. Kato, “Performance Enhancement of TCP over Adaptive Multi-Rate IEEE 802.11 Wireless LANs,” in *Proc. IEEE RWS*, Orlando, FL, USA, Jan. 2008, pp. 579–582.

[13] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” RFC 2581, Apr. 1999.

[14] L. Brakmo and L. Peterson, “TCP Vegas: End to End Congestion Avoidance on a Global Internet,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[15] V. Jacobson, R. Braden, and D. Borman, “TCP Extensions for High Performance,” RFC 1323, May 1992.

[16] Scalable Network Technologies, Inc. Qualnet. [Online]. Available: http://www.scalable-networks.com/