Performance Enhancement of TCP over Adaptive Multi-Rate IEEE 802.11 Wireless LANs

Kenichi Kashibuchi †, Tarik Taleb †, Abbas Jamalipour ‡, Yoshiaki Nemoto †, and Nei Kato †

†Graduate School of Information Sciences, Tohoku University, Sendai, Miyagi, 980-8579, Japan
‡School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia
Email: buchiken@it.ecei.tohoku.ac.jp

***Abstract*— This paper presents a new TCP window size adjustment scheme for the automatic rate fallback (ARF) algorithm. The proposed scheme utilizes the information about radio usage monitored on the IEEE 802.11 MAC layer. Each station calculates the maximum available window size on each long-lived connection based on these information when the data transmission / reception rate is switched, and limits the maximum sending window size.**

The performance of the proposed scheme is investigated and compared with that of standard TCP through several simulations. The simulation results show that the proposed scheme enhances the transmission efficiency of TCP.

***Index Terms*— Adaptive modulation and coding, cross-layer design, transmission control protocol, wireless LAN.**

I. Introduction

In recent years, wireless local area networks (Wireless LANs) have become important network infrastructures and are available at home as well as in public areas. The IEEE 802.11 wireless LAN physical (PHY) layer supports multiple data rates by using various modulation techniques and channel coding rates. For example, the IEEE 802.11a PHY provides eight data rates from 6 Mbps to 54 Mbps. The link adaptation algorithms select one of the multiple available transmission data rates according to varying conditions.

In this paper, we propose a scheme that enhances the transmission efficiency of Transmission Control Protocol (TCP) for the automatic rate fallback (ARF) [1], the most commonly used link adaptation algorithm. The proposed scheme adjusts the TCP window size based on the information on radio usage monitored by the IEEE 802.11 media access control (MAC) layer. In the proposed scheme, each station calculates the maximum available window size on each long-lived connection based on these information when the transmission / receiving data rate is switched, and eventually limits the maximum sending window size.

The remainder of this paper is organized as follows. Section II gives a brief summary of the ARF algorithm. We then introduce a new TCP window size adjustment mechanism for ARF in Section III. In Section IV, we discuss the performance of the proposed scheme. Finally, Section V concludes the paper.

II. Background

ARF is the earliest rate adaptation algorithm and is the most widely used one. In the ARF algorithm, each sender switches the transmission rate to a particular station after consecutive successes or failures in transmissions to that station. With the IEEE 802.11 MAC protocol, when a station receives a unicasted frame, it responds with an acknowledgement (ACK) frame. Therefore, the sender is aware of the success or failure in the transmission. If the sender fails in receiving the consecutive θ_{down} ACK frames[1], it decreases the transmission rate to the next lower data rate and sets a timer (T_{up}). When the sender successfully receives the consecutive θ_{up} ACK frames 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.

To improve the transmission efficiency on the MAC layer, several approaches [2] have been proposed. However, there have been few papers investigating the impact of ARF on the transmission efficiency of TCP.

III. TCP Window Size Adjustment for ARF

The proposed scheme aims at improving the transmission efficiency of TCP by exchanging information among layers (*i.e.*, a cross-layer approach). Specifically, the proposed scheme utilizes the following information to adjust the window size:

- The channel occupancy time used by the station itself
- The channel occupancy times used by the other stations
- The number of active users
- Transmission or receiving data rate

In the remainder of this paper, we describe in detail the proposed scheme only when the RTS/CTS option is used. Other cases are omitted due to the paper length limitations. It should be noted that the proposed scheme can be also employed without the RTS/CTS option.

[1]When the request-to-send/clear-to-send (RTS/CTS) option is used, θ_{down} or θ_{up} includes the number of successes or failures in receiving CTS frames, respectively.

A. Measurement of Channel Occupancy Times on the MAC layer

In this section, we show how to measure the channel occupancy time. Here, $d_{\text{me}}^{\text{DL}}$, $d_{\text{others}}^{\text{DL}}$, $d_{\text{me}}^{\text{UL}}$, and $d_{\text{others}}^{\text{UL}}$ denote various channel occupancy times, respectively, as follows:

- $d_{\text{me}}^{\text{DL}}$: Duration for which a station downloads a data frame from the access point (AP)
- $d_{\text{others}}^{\text{DL}}$: Duration for which another station downloads a data frame from the AP
- $d_{\text{me}}^{\text{UL}}$: Duration for which the station in question uploads a data frame to the AP
- $d_{\text{others}}^{\text{UL}}$: Duration for which another station uploads a data frame to the AP

Fig. 1 shows the access mechanism of the IEEE 802.11 distributed coordination function (DCF), which is the fundamental one. t_{DIFS} and t_{SIFS} denote the length of DCF inter-frame space and that of short inter-frame space, respectively. t_{DIFS} is given by $(t_{\text{SIFS}} + 2 \cdot t_{\text{slot}})$, where t_{slot} represents a slot time. t_{backoff} is a backoff interval. Assuming that no collision occurs, its average is given by:

$$\overline{t_{\text{backoff}}} = \frac{CW_{\min}}{N+1} \cdot t_{\text{slot}} \tag{1}$$

where $CW_{\min}$ is the minimum value of the contention window, and N is the number of active users (*i.e.*, the number of STAs actually communicating, not only associating, with the AP). N can be estimated by counting how many stations transmit RTS, CTS, DATA, or ACK.

If there are hidden terminals, some station cannot sense a data frame from some other stations. With RTS/CTS, all stations can receive either RTS frames or CTS frames. Therefore, each station can estimate the channel occupancy times used by any station as follows:

$$\begin{aligned} d_{\text{me}}^{\text{DL}} &= d_{\text{others}}^{\text{DL}} \\ &= t_{\text{DIFS}} + \overline{t_{\text{backoff}}} + t_{\text{RTS}}^{\text{recv}} + \delta_{\text{RTS}} && (2) \\ d_{\text{me}}^{\text{UL}} &= t_{\text{DIFS}} + \overline{t_{\text{backoff}}} + t_{\text{RTS}}^{\text{send}} + \delta_{\text{RTS}} && (3) \\ d_{\text{others}}^{\text{UL}} &= t_{\text{DIFS}} + \overline{t_{\text{backoff}}} \\ &\quad + \hat{t}_{\text{RTS}}^{\text{send}} + t_{\text{SIFS}} + t_{\text{CTS}}^{\text{recv}} + \delta_{\text{CTS}} && (4) \end{aligned}$$

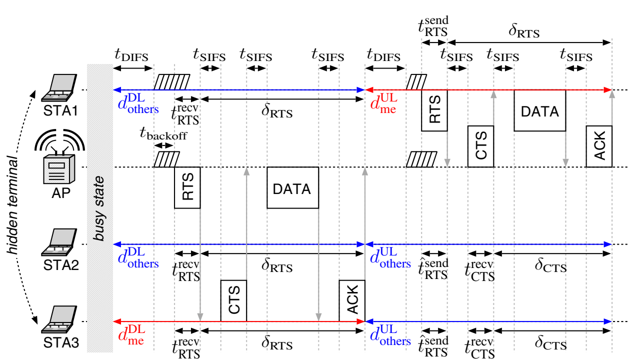


Fig. 1. Measurements of channel occupancy times.

where $t_{\text{RTS}}^{\text{send}}$ and $t_{\text{RTS}}^{\text{recv}}$ are the durations for sending and receiving a RTS frame, respectively. $\hat{t}_{\text{RTS}}^{\text{send}}$ is the estimated duration for sending a RTS frame. It can be calculated because RTS frames are of fixed length and are always transmitted at the minimum data rate so that all stations can receive them. $t_{\text{CTS}}^{\text{recv}}$ is the duration for receiving a CTS frame. δ_{RTS} and δ_{CTS} are indicated in the duration field in the MAC header of RTS and CTS frames, respectively. δ_{RTS} is given by adding $(3 \cdot t_{\text{SIFS}})$ to the duration for transmitting a CTS, a DATA, and an ACK frames. δ_{CTS} is given by adding $(2 \cdot t_{\text{SIFS}})$ to the duration for transmitting a DATA frame and an ACK frame.

B. Calculation of Maximum Available Window Size on the TCP layer

The proposed scheme controls the maximum value of the available window size, $wnd_{\max}$. Each station STA$_i$ recalculates $wnd_{\max}$ when increasing or decreasing its transmission / receiving data rate. $wnd_{\max}$ is calculated based on the following values:

- T: The length of an observation slot [s]
- t_{beacon}: The channel occupancy time used by beacon frames from the AP (*i.e.*, the duration for receiving beacon frames) in T [s]
- C_i: The data transmission / reception rate STA$_i$ used in T [bps],
- t_i: The channel occupancy time used by STA$_i$ in T [s]
- S_i: The total size of the sections to which the adaptive modulation was applied (see Fig. 2), which was transmitted or received by STA$_i$ in T [bit]
- b: The number of TCP segments acknowledged by a single TCP ACK (*i.e.*, when the delayed ACK option [3] is used, b is set to two, otherwise b is set to one)
- M_i: The number of relatively long-lived TCP connections that STA$_i$ had in T (*e.g.*, the number of file transfer protocol (FTP) connections)
- $\tau_{i,j}$: The channel occupancy time used by the j-th connection of STA$_i$ in T [s]
- $\overline{\lambda_{i,j}}$: The average length of TCP data segments, which were transmitted or received by the j-th connection of STA$_i$ in T [s]

It should be noted that every station $\{\text{STA}_i, i=1,\ldots,N\}$ needs to compute the channel occupancy times (t_i) used not only by the station itself but also by the other stations, which can be done based on the description in Section III-A. Meanwhile, each station needs to compute the other values only related to itself.

1) When the data transmission / reception rate increases: As the ARF algorithm tends to fail in transmitting the first DATA frame after increasing the data rate, the proposed scheme performs the following procedures only when a station can receive the first DATA frame or the first ACK frame after that.

When STA$_i$ increases the data rate from C_i to C_i^{new}, the extra duration will occur in the basic service set (BSS):

$$\Delta^{+} = \frac{S_i}{C_i} - \frac{S_i}{C_i^{\text{new}}} \tag{5}$$

In addition, the BSS has an unoccupied duration in T equal to:

$$t_{\text{remain}} = T - \sum_{n=1}^{N} t_n - t_{\text{beacon}} \tag{6}$$

The newly available duration in the BSS is thus estimated as $(\Delta t = \Delta^{+} + t_{\text{remain}})$.

The proposed algorithm attempts to evenly allocate this increment of duration among the N active STAs. The newly available duration for STA$_i$ is given by:

$$\Delta t_i = \Delta t / N \tag{7}$$

Similarly, STA$_i$ allocates Δt_i evenly among M_i connections. The newly available duration for the j-th connection is $(\Delta\tau_{i,j} = \Delta t_i / M_i)$.

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

$$\Theta_{i,j} = \frac{\tau_{i,j} + \Delta\tau_{i,j}}{T_{\text{TCP}}\left(C_i^{\text{new}}, \overline{\lambda_{i,j}}, b\right)} \cdot \overline{\lambda_{i,j}} \cdot b \cdot \frac{1}{T} \tag{8}$$

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

$$T_{\text{TCP}}\left(C, \lambda, b\right) = T_{\text{data}}\left(C, \lambda\right) \cdot b + T_{\text{data}}\left(C, 0\right) \tag{9}$$

where $T_{\text{data}}\left(C, \lambda\right)$ is the period required for transmitting or receiving a TCP segment with a size λ at a data rate C:

$$\begin{aligned} T_{\text{data}}\left(C, \lambda\right) = t_{\text{DIFS}} + \overline{t_{\text{backoff}}} + t_{\text{RTS}} + t_{\text{SIFS}} + t_{\text{CTS}} \\ + t_{\text{SIFS}} + t_{\text{DATA}}(C, \lambda) + t_{\text{SIFS}} + t_{\text{ACK}}(C) \end{aligned} \tag{10}$$

where t_{RTS}, t_{CTS}, $t_{\text{DATA}}(C, \lambda)$, and $t_{\text{ACK}}(C)$ that depend on the specifications of the PHY and MAC layers, are the duration for sending or receiving a RTS frame, a CTS frame, a DATA frame, and an ACK frame, respectively (Fig. 2).

Since the maximum TCP throughput is ideally given by (wnd_{max}/RTT), the maximum value of the available window size in the j-th connection of STA$_i$ is calculated as:

$$wnd_{\text{max}_{i,j}} = RTT_{i,j} \cdot \Theta_{i,j} \tag{11}$$

where $RTT_{i,j}$ is the round trip time of the connection.

2) When the data transmission/reception rate decreases: The proposed scheme does not perform the following procedures when a station cannot receive the first DATA frame or the first ACK frame after increasing the data rate, because in that case the data rate is merely reverted to the former one.

When STA$_i$ decreases the data rate from C_i to C_i^{new}, more time will be required in the BSS:

$$\Delta^{-} = \frac{S_i}{C_i^{\text{new}}} - \frac{S_i}{C_i} \tag{12}$$

STA$_i$ should decrease the channel occupancy time by Δ^{-}. On the other hand, if the BSS has some unoccupied time (t_{remain}) in T, the proposed algorithm attempts to distribute this duration equally among the N active STAs. Therefore, the newly available duration for STA$_i$ is given by:

$$\Delta t_i = \frac{\Delta t_{\text{remain}}}{N} - \Delta^{-} \tag{13}$$

which may be a negative value.

Hereinafter, it is the same as when STA$_i$ increases the data rate. The maximum TCP throughput the j-th connection of STA$_i$ can achieve is estimated as in Eq. (8) and the maximum value of the available window size in that connection is calculated according to Eq. (11).

C. Adjustment of the Sending Window Size

The proposed scheme controls the transmission rate by bounding the maximum window size.

1) When the station is the sender: The STA$_i$ limits the congestion window size ($cwnd$) of each connection to the calculated $wnd_{\text{max}_{i,j}}$. Obviously, $wnd_{\text{max}_{i,j}}$ should not exceed the sender buffer size.

2) When the station is the receiver: The STA$_i$ transmits the calculated $wnd_{\text{max}_{i,j}}$ 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. If $wnd_{\text{max}_{i,j}}$ does not exceed the station's receiver buffer size, $wnd_{\text{max}_{i,j}}$ is written in the $rwnd$ field. By doing so, since the TCP sender sets its sending window size to the minimum of $cwnd$ and $rwnd$, the maximum sending window size is bounded to $wnd_{\text{max}_{i,j}}$.

IV. Performance Evaluation

We carried out several simulations using QualNet 4.0.1 [4] and compared the performance of the proposed scheme against that of standard TCP.

Fig. 3 depicts the configuration of the considered network. In our simulations, we model an IEEE 802.11a (5.2 GHz) BSS consisting of an AP and a STA ($N = 1$). There is no other active station. The coverage radius of

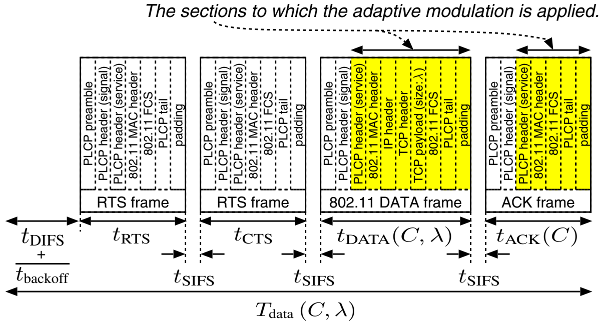


Fig. 2. Calculation of $T_{\text{data}}\left(C, \lambda\right)$ (in case of IEEE 802.11a)

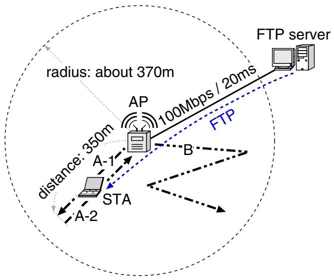
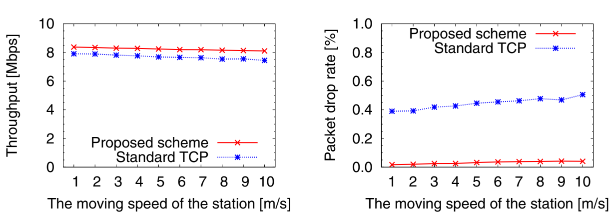


Fig. 3. Simulated network topology.

the cell is about 370 meters when the data rate is 6 Mbps; and is about 37 meters when the data rate is 54 Mbps. The RTS/CTS option is disabled (*i.e.*, RTS threshold is fixed to 0). The parameters in the ARF algorithm, θ_{up}, θ_{down}, and T_{up}, are 10, 2, and 60 msec, which are the default values in QualNet, respectively.

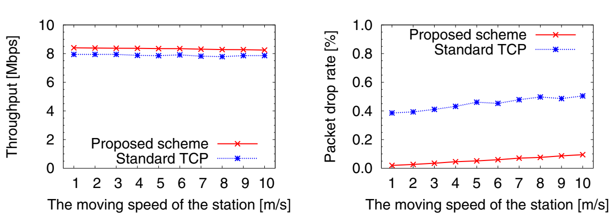
In our scenarios, STA has only one FTP connection and downloads a file with infinite length from a FTP server. The maximum segment size (MSS) is set to 1460 bytes and IP packet size, including TCP/IP headers, is equal to 1500 bytes. TCP window scaling option and timestamps option [5] are enabled. The delayed ACK option is disabled ($b = 1$). Both the sender and receiver buffer sizes are set to 131072 bytes. As a standard TCP, we used TCP NewReno for comparison. We thus have integrated the proposed scheme into TCP NewReno and IEEE 802.11 MAC. The results are an average of 20 simulation runs.

We first investigate the performance when the STA goes straightly. The moving speed of the STA is varied from 1 m/s to 10 m/s. Figs. 4 and 5 show the results when the STA moves in a direction away from the AP ("A-1" in Fig. 3) and toward the AP ("A-2"), respectively. In both scenarios, the proposed scheme achieves a higher throughput and a lower packet drop rate regardless of the movement speed of STA. The reason behind this



(a) Throughput. (b) Packet drop rate.

Fig. 4. STA moves away from the AP.



(a) Throughput. (b) Packet drop rate.

Fig. 5. STA moves toward the AP.

performance consists in the fact that the proposed scheme sets the window size to more appropriate values than standard TCP does when the data rate is changed on the MAC layer. Furthermore, because the STA changes the data rate many times in a short interval due to the ARF algorithm, the proposed scheme becomes more effective. By the same token, as shown in Table I, when the STA moves according to the random way point model with a maximum velocity 10 m/s, a minimum velocity 0 m/s and a pause time of 0 sec ("B" in Fig. 3), the proposed scheme exhibits similar performance.

TABLE I

STA MOVES RANDOMLY.

	Throughput [Mbps]	Packet drop rate [%]
Proposed scheme	8.85	0.017
Standard TCP	8.44	0.351

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new TCP window size adjustment algorithm for ARF. The proposed scheme utilizes information on the radio usage monitored by the IEEE 802.11 MAC layer. In the proposed scheme, each station computes the maximum window size available at each long-lived connection based on these information when the data rate is changed. Finally, the proposed scheme adjusts the maximum sending window size to the computed values.

The performance of the proposed scheme was evaluated and compared with that of the standard TCP through several simulations using a simple IEEE 802.11a BSS. The obtained results demonstrated that the proposed scheme effectively improves the transmission efficiency, *i.e.*, the throughput and the packet loss ratio.

As a part of our future work, more extensive simulations using different number of STAs, connections, and background traffic will be conducted. The comparison with other TCP variants (*e.g.*, TCP Westwood [6]) will be also performed.

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 Technical Journal*, vol. 2, no. 3, pp. 118–133, summer 1997.

[2] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proc. ACM MSWiM*, Venice, Italy, Oct. 2004.

[3] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, Apr. 1999.

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

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

[6] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," in *Proc. ACM MOBICOM*, Jul. 2001.