Master's Thesis

Title

Congestion Control Mechanisms for Alleviating TCP Unfairness in Wireless LAN Environment

Supervisor

Professor Hirotaka Nakano

Author

Masafumi Hashimoto

February 15th, 2010

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Congestion Control Mechanisms for Alleviating TCP Unfairness
in Wireless LAN Environment

Masafumi Hashimoto

Abstract

Per-flow unfairness among TCP flows in IEEE 802.11 wireless LAN environment has been reported in past literature. Many researchers have proposed various solutions for alleviating the unfairness. These solutions diminish TCP throughput unfairness by mostly modifying MAC protocol or queue management mechanisms at access points based on detailed information in wireless LAN such as the number of coexisting stations, flow types, and flow rates. However, MAC protocols of wireless access points are generally implemented in hardware, so it takes large cost to change them. Moreover, collecting the detailed information of the network increase the network complexity.

In this thesis, the author proposes a simple modification to TCP congestion control mechanisms to alleviate the unfairness, which activates the congestion control when detecting TCP ACK packet losses. As an evaluation metric of fairness among users in above unfairness problems, Jain's fairness index has been generally utilized. Some solutions which addressed bandwidth sharing among coexisting flows (users) would alleviate the unfairness while degrading the total bandwidth utilization. Since Jain's fairness index depends only on the variation of allocated values to users, the index cannot evaluate such trade-off relationships between fairness and throughput accurately. Therefore, in this thesis, the author proposes a new metric, which can evaluate the trade-off relationships between per-flow fairness and bandwidth utilization at the network bottleneck.

Through experimental evaluations using real wireless LAN environments, the author presents that the proposed method is significantly effective not only for TCP fairness among upstream flows but also for fairness between upstream and downstream flows, and that it can take quite better trade-off between fairness and bandwidth utilization regardless of vendor implementations of access points and wireless interface cards.

Keywords

Transmission Control Protocol (TCP)

Wireless LAN

Fairness

Performance metric

Congestion control against TCP ACK packet losses

Contents

List of Figures

List of Tables

1 Introduction

With recent development on wireless networking technologies, accessing the Internet through wireless LAN (WLAN) is becoming common situation and WLAN-based Internet access environment is often served in public sites such as railway stations and airports. In such situations, it is important to consider fairness among coexisting users.

IEEE 802.11 families [1-4] are standardized by IEEE as the wireless LAN environment. IEEE 802.11 standard defines two coordination functions as medium access control protocol: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). In DCF, medium access mechanism is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In CSMA/CA, when the channel is sensed as busy before a transmission, the transmission is deferred for a random backoff interval for reducing the probability of frame collisions. The traffic in a WLAN consists of the traffic flowing from client stations to wired networks via an access point (called upstream traffic) and vice versa (called downstream traffic). CSMA/CA enables stations, including an access point, to access wireless channel fairly. Therefore, fairness among stations in the WLAN is ensured at the MAC layer. However, downstream traffic is transmitted only from an access point whereas upstream traffic is generated from multiple client stations. Therefore, upstream traffic obtains more access opportunities to wireless channel than downstream traffic. This means that the fairness among stations realized at MAC layer does not mean the fairness among flows at the upper layers. In fact, unfairness among flows at the upper layer has been reported [5]. Furthermore, it has been pointed out that TCP flows experience severe unfairness in WLANs [6-9]. When TCP is utilized as a transport-layer protocol, the unfairness among upstream flows caused by its congestion control mechanisms: TCP activates the congestion control against data packet losses, but not against ACK packet losses.

On the other hand, PCF is a contention-free mechanism. The implementation of PCF is optional according to the standard. Utilizing PCF is one of possible solutions to solve the above unfairness caused by utilizing DCF. However, PCF cannot realize enough Quality of Service (QoS) due to unpredictable beacon delays and unknown transmission durations of the polled stations [10]. In addition, PCF is not included in the part of Wi-Fi Alliance's interoperability standard [11]. Therefore, PCF is implemented in few wireless devices.

For the above reason, IEEE 802.11e [12] was standardized for supporting QoS requirements in

WLANs. It realizes different QoS requirements among flows that are generated from the stations by utilizing per-flow queueing at the station's network interface. However, since it is not targeted to per-station fairness, it does not solve the unfairness in WLANs described above.

There are various solutions proposed to alleviate the unfairness problems [5-7, 13-30]. These solutions diminish TCP throughput unfairness by mainly modifying MAC protocol parameters or queue management mechanisms in access points. However, MAC protocols of wireless access points are generally implemented in hardware, so it takes large cost to change them. Furthermore, Some solutions can alleviate the unfairness, but they may also cause the other unfairness at transport layer: Priority-based solution at access point [14] can significantly improve TCP fairness, whereas it may cause UDP unfairness by giving priorities to the access point too much. Furthermore, some methods need to estimate the number of flows and the throughput of each flow at the access point.

Therefore, in this thesis, the author proposes a transport-layer solution for alleviating the unfairness in WLANs. The reasons of employing a transport-layer solutions are as follows: First, MAC-layer solutions may cause the other unfairness at transport layer described above. Second, the unfairness is caused by the behavior of the transport-layer protocols. The last reason is that transport-layer protocols are generally implemented in software, so it is easier to modify transport-layer protocols than to modify 802.11 MAC protocols. The proposed method alleviates the unfairness by detecting TCP ACK packet losses as an indication of the congestion at an access point. It requires small modification on TCP congestion control mechanisms only on stations in the WLAN.

Generally, as an evaluation metric of fairness among users, flows and so on, Jain's fairness index [31] has been utilized. Since the index depends only on the variation of allocated values, the index values intuitively represents whether the allocated values are fair or not. Here, assume that we have two solutions where both have identically effective for alleviating unfairness but one of them degrades the total amount of allocated values while the other does not decrease the total value. People think that the latter is superior to the former, but both solutions are identical from the viewpoint of Jain's fairness index. That is, Jain's fairness index cannot evaluate such situations accurately.

In this thesis, a novel performance metric is proposed, considering the trade-off relationships between per-flow fairness and bandwidth utilization at network bottleneck. The proposed metric can evaluate both of fairness and bandwidth utilization concurrently with a single metric value, and can control the balance of fairness and utilization in calculating the metric value.

The author conducts the extensive experiments using real WLAN environments with the products from several vendors. In the experiments, ten laptops, which share a single access point, and a wired computer transmits bulk data for both directions, varying the ratio of upstream and downstream TCP flows. The effectiveness of the proposed method is evaluated from the viewpoint of fairness and bandwidth utilization by using throughput of each flow, Jain's fairness index, and the proposed index as evaluation metrics. The author also confirms the performance sensitivity against the vendor implementations of access points and wireless client terminals since the evaluation is conducted on actual wireless environments.

The rest of the thesis is organized as follows. In Section 2, unfairness problems among TCP flows in IEEE 802.11 WLAN environment are discussed. In Section 3, a novel performance metric considering both of per-flow fairness and bandwidth utilization is introduced. Section 4 describes my solution for alleviating unfairness problems in WLANs. In Section 5, the author presents experimental results using real WLAN environment after experimental settings and methodologies are described. The author finally concludes this thesis and discusses future work in Section 6.

2 Fairness among TCP Flows in WLAN Environments

In the typical WLAN environment, all stations including an access point use the same parameters of CSMA/CA. Thus, the contending stations obtain the fair access opportunities to wireless channel in the long term. It also means that upstream flows, transmitted from multiple client stations, and downstream flows, transmitted from an access point, share the same wireless channel. Therefore, when n client stations and a single access point exist in the WLAN network, the access opportunities to wireless channel of upstream traffic is $n/(n+1)$, whereas that of downstream traffic is $1/(n+1)$. Hence, an access point is likely to become a congested point and the fairness at MAC layer dose not contribute to the fairness at the upper layer such as UDP and TCP. Such difference of access opportunities to wireless channel between upstream and downstream traffic causes serious problems since many applications used in WLANs generate not only downstream traffic but also upstream traffic (e.g. P2P file sharing and VoIP applications).

When TCP is used as a transport-layer protocol, the serious per-flow unfairness occurs not only between upstream and downstream flows but also among upstream flows. In what follows, the two types of unfairness are explained in Subsections 2.1 and 2.2, respectively. In what follows in this section, the author assumes that multiple stations share a singe access point and the wireless network bandwidth is fully utilized, resulting that the access point becomes a congested point.

2.1 Fairness among Upstream Flows

Suppose that each client station has an upstream TCP flow, as depicted in Figure 1. Since the wireless interface of the access point is a congested point, TCP ACK packets of the upstream flows are discarded at an access point buffer. Since TCP does not activate the congestion control against ACK packet losses, the congestion window size of the upstream flows keeps growing up until Retransmission Time Out (RTO) occurs when *all* ACK packets in a window are lost. When all ACK packets of a certain upstream flow are discarded, its congestion window size is set to one packet. The problem is that the flow cannot increase the congestion window in a short time since the buffer at the access point is still fully utilized and ACK packet losses cannot be avoided. That is, once a certain flow experienced RTO, the flow cannot increase its congestion window size for a long time (such as hundreds and thousands seconds as shown in Subsection 5.3). This causes the throughput unfairness among upstream TCP flows.

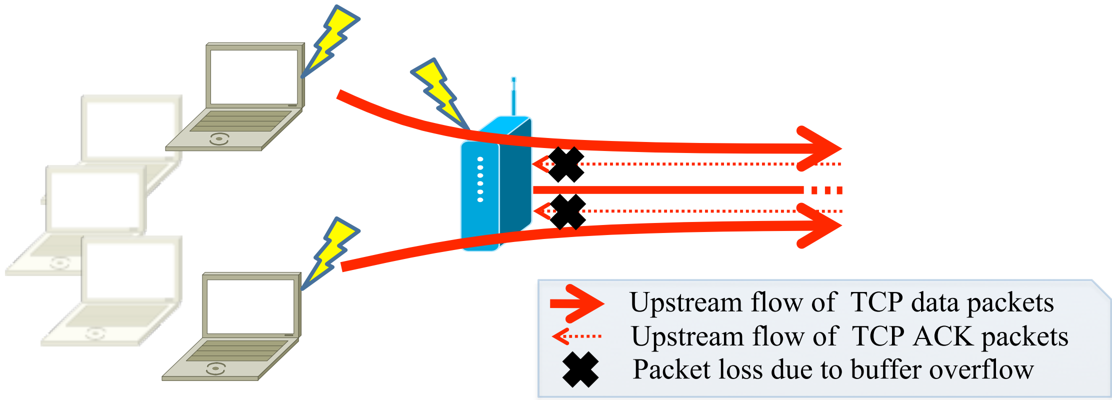


Figure 1: Access point congestion caused by multiple upstream TCP flows

2.2 Fairness between Upstream and Downstream Flows

On the other hand, when upstream and downstream TCP flows coexist as shown in Figure 2, ACK packets of upstream flows and data packets of downstream flows are discarded at access point buffer. In this situation, upstream TCP flows keep growing its congestion window, whereas downstream TCP flows decrease the congestion window, since TCP activates the congestion control mechanism only against data packet loss(es). That is, the downstream TCP flows keep low transmission rate, whereas the upstream TCP flows keep high transmission rate. Furthermore, upstream TCP flows obtain more access opportunities to wireless channel than downstream TCP flows for the same reason as explained in the beginning of Section 2. Consequently, the serious throughput unfairness occurs between upstream and downstream TCP flows. Note that such unfairness also occurs when UDP is used as a transport-layer protocol, but the degree of the unfairness among TCP flows more serious than that among UDP flows due to TCP congestion control mechanisms [5].

2.3 Existing Solutions for Alleviating Unfairness

In wired networks, the unfairness problems described in Subsections 2.1 and 2.2 does not occur because the network resources in wired networks is generally operated in full-duplex mode, whereas network resources of WLAN is operated in half-duplex mode. That is, when WLAN is operated in full-duplex mode, the unfairness problem can be avoided.

However, it is not suitable to operate WLAN in full-duplex mode. In wired networks, it is

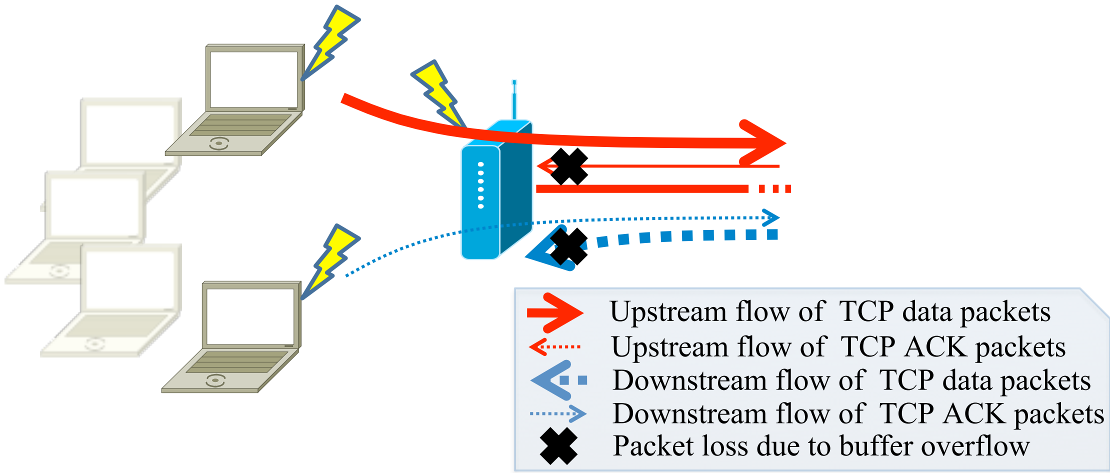


Figure 2: Access point congestion caused by upstream and downstream TCP flows

easily to add no-interference links by adding new transmission cables. To do the same thing in WLAN, the stations and the access point require at least two no-interference wireless interfaces for upstream and downstream transmissions. However, such separated channel utilization would increase the complexity of the network, since it increases the hardware cost and since it requires the different treatment of access point and client station. Also, it may waste the wireless resources due to the unpredictable traffic demands for both directions. Moreover, since it is common situations that multiple WLAN services operated in different policies coexist in WLAN environments, such wasting channel usage should be avoided. Thus, WLANs are operated in full-duplex mode is unbefitting to solve the unfairness in WLAN.

Instead, various solutions have been proposed for diminishing above TCP unfairness problems [5-7, 13-30] without utilizing additional wireless channel. A solution proposed in [6] improves fairness among upstream and downstream flows by rewriting the advertised receiver window size at the access point. This solution has a drawback in terms of the dependence on each flow's Round Trip Time (RTT). Furthermore, it has been pointed out that this method degrades the wireless channel utilization in some situations [22]. A solution proposed in [9] divides the buffer in an access point into for data packets and for ACK packets. The packets buffered in each buffer are discarded with certain probabilities. In [7], TCP unfairness among uplink flows is diminished by filtering ACK packets in an access point and decreasing the number of buffered ACK packets at the access point. The authors in [14] proposed to shorten the carrier sense duration of the

WLAN access points. A solution proposed in [22] improves TCP unfairness among upstream and downstream flows by controlling the rate of upstream flows such that total throughput should be divided equally between upstream and downstream flows.

However, the methods in [6, 7, 9] need to parse and rewrite the TCP header at the access point. Moreover, the method in [7] requires maintaining per-flow information at the access point. Using the method in [14], since transmission of the access point gets preference over that of the other stations, it can significantly improve TCP fairness, however, it may cause UDP unfairness due to the priority of excessive downstream UDP flows. The method in [22] also requires monitoring the throughput of upstream and downstream flows. In addition, these methods take large cost to change existing hardware devices due to requiring modification to MAC protocols or queue management mechanisms.

On the other hand, in this thesis, the author employs transport-layer solutions to alleviate unfairness problem among TCP flows in WLAN environment. The reasons of this are as follows:

- MAC-layer solutions such as priority-based solution at access points [14] may cause the other unfairness at transport layer.
- The unfairness described in Subsections 2.1 and 2.2 is caused by TCP behaviors.
- Transport-layer protocols are generally implemented in software, so it is easily to modify them.

The proposed method requires only small modifications to the sender-side TCP behavior in wireless stations.

3 A Metric for Trade-off Evaluation between Fairness and Utilization

The definition of fairness is important when we discuss fairness among flows, since the improvement of fairness is sometimes achieved at the expense of total bandwidth utilization. In the past researches [6, 7, 9, 14, 22], the fairness is defined as that all flows contending wireless channel in a WLAN achieve the same throughput.

To evaluate the fairness, Jain's fairness index has been utilized, defined as follows:

$$F_j(X) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} \tag{1}$$

where n is the number of contending users, $X = \{x_1, x_2, \cdots, x_n\}$ is a set of allocations for n users such that x_i is an allocation for user i. The smaller the variation of allocations the index value is close to one, and the larger the variation of allocations, the index value approaches $1/n$. Note that Jain's fairness index is independent of the scale of allocations. For example, consider fairness when allocating 30, 40 and 10 dollars respectively to three persons and allocating 300, 400 and 100 dollars respectively to three persons. Both cases are equivalent from the viewpoint of Jain's fairness index (0.82).

However, the total amounts of allocated values are different. That is, Jain's fairness index is not suitable to compare fairness as well as considering the total amount of allocations. The total amount of allocated values corresponds to the network bandwidth utilization in the context of network bandwidth sharing. Therefore, when we have a solution for alleviating unfairness while slightly degrading total throughput, Jain's fairness index cannot evaluate such performance trade-off accurately.

For above reasons, a novel evaluation index is necessary for evaluating fairness improvement methods in WLAN environments. In what follows, the novel evaluation index defines in Subsection 3.1 and simple comparisons between Jain's fairness index and the proposed index is shown in Subsection 3.2.

3.1 Definition

Given a throughput set $X = \{x_1, x_2, \cdots, x_n\}$, where x_i is the throughput of i th flow, and network bandwidth at bottleneck, C, where $\sum_{i=1}^{n} x_i \le C$. The author now defines *fair and fully-utilized* *throughput* $x_f = \frac{C}{n}$ where all flows achieve the same throughput and the network bandwidth is fully utilized. Here, using the relationship between Jain's fairness index $F_j(X)$ in Equation (1) and total throughput $\sum_{i=1}^{n} x_i$, the author defines the desired properties for proposed fairness index $F(X, C)$ as follows:

(i) If $F_j(X) = F_j(Y)$ and $\sum_{i=1}^{n} x_i < \sum_{i=1}^{n} y_i \le C$, then $F(X, C) < F(Y, C)$.

(ii) If $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \le C$ and $F_j(X) < F_j(Y)$, then $F(X, C) < F(Y, C)$.

where $Y = \{y_1, y_2, \cdots, y_n\}$.

The author starts from the index $f(X, C)$ which represents how far is the throughput of each user from the fair and fully-utilized throughput (x_f):

$$f(X, C) = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_f)^2. \tag{2}$$

The author then normalizes $f(X, C)$ by x_f and obtains $g(X, C)$ as follows:

$$g(X, C) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_f)^2}}{x_f}. \tag{3}$$

According to [31], Jain's fairness index can be transformed into

$$F_j(X) = \frac{1}{1 + COV^2} \tag{4}$$

$$COV = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\bar{x}} \tag{5}$$

where COV is a coefficient of variance of allocations to the users and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is an average of allocations. By comparing Equations (3) and (5), an index $h(X, C)$ is composed as the similar way in Equation (4) as follows:

$$\begin{aligned} h(X, C) &= \frac{1}{1 + g(X, C)^2} \\ &= \frac{C^2}{n\sum_{i=1}^{n} x_i^2 - 2C\sum_{i=1}^{n} x_i + 2C^2}. \end{aligned} \tag{6}$$

The index $h(X, C)$ considers the total utilization at a certain degree, but it gives undesired results in some situations. For example, consider the network bandwidth at bottleneck link is 10 Mbps and the number of users is five ($x_f = 2$ Mbps). Comparing the two throughput distributions $X = \{3\ \text{Mbps}, 1\ \text{Mbps}, 1\ \text{Mbps}, 1\ \text{Mbps}, 1\ \text{Mbps}\}$ and $Y = \{3\ \text{Mbps}, 3\ \text{Mbps}, 1\ \text{Mbps}, 1\ \text{Mbps},$

1 Mbps$\}$, $h(X, 10 \text{ Mbps})$ is equal to $h(Y, 10 \text{ Mbps})$ $(= 0.8)$ despite total throughput of Y is larger than that of X. For this reason, an additional term is necessary to the index to take the utilization into more consideration. Finally, the following definition is obtained as a novel index for evaluating the trade-off relationships between fairness and utilization:

$$F(X, C) = \frac{C^2}{n\sum_{i=1}^{n} x_i^2 - 2C\sum_{i=1}^{n} x_i + 2C^2}\left(\frac{\sum_{i=1}^{n} x_i}{C}\right)^{\alpha} \tag{7}$$

where α is a parameter for adjusting the contribution of bandwidth utilization at network bottleneck to the index.

Note that the index satisfies the above-mentioned properties (i) and (ii). The index become close to one when the bandwidth utilization of network bottleneck is close to 100% and the throughput variation of each flow is small. Oppositely, it approaches zero when the bandwidth utilization is low and the throughput variation is large. The index is equivalent to Jain's fairness index when total throughput is equal to bandwidth utilization at network bottleneck, i.e., when $\sum_{i=1}^{n} x_i = C$.

3.2 Comparison with Jain's Fairness Index

In Table 1, the comparison between Jain's fairness index and the proposed index is shown by using simple examples where the network capacity bandwidth is 30 Mbps. In Cases 1, 2 and 3 in Table 1, the throughput distributions have the same variation, but the total throughput is different. These situations are related to the property (i) in Subsection 3.1. In these situations, the proposed index can differentiate them whereas Jain's index cannot distinguish among them. Similarly, in Cases 4, 5 and 6 where only one user occupies the network bandwidth, the proposed index can differentiate them while Jain's index has same values. In Cases 2, 7 and 8 present situations where the total throughputs are identical, but the throughput distributions have different variations. These situations corresponded to the property (ii) in Subsection 3.1. In these situations, the order of the three cases in Jain's index and that in the proposed index are identical. Moreover, when the total throughput is equal to the network bandwidth capacity such as Cases 4, 9 and 10, Jain's index and the proposed index show the identical index values.

Table 1: Comparison between Jain's fairness index and the proposed index ($C = 30$ Mbps)

Case	Throughput distribution [Mbps]	Total [Mbps]	Jain's index	The proposed index
1	{ 3, 3, 3, 3, 3, 3, 3, 3, 3, 3}	30	1.00	1.00
2	{ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2}	20	1.00	0.60
3	{ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}	10	1.00	0.23
4	{30, 0, 0, 0, 0, 0, 0, 0, 0, 0}	30	0.10	0.10
5	{15, 0, 0, 0, 0, 0, 0, 0, 0, 0}	15	0.10	0.14
6	{ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}	1	0.10	0.02
7	{ 0, 0, 1, 1, 2, 2, 2, 4, 4, 4}	20	0.65	0.49
8	{ 3, 3, 2, 2, 2, 2, 2, 2, 1, 1}	20	0.91	0.58
9	{ 1, 1, 1, 2, 2, 2, 3, 3, 3, 12}	30	0.48	0.48
10	{ 1, 1, 1, 3, 3, 3, 3, 4, 5, 6}	30	0.78	0.78

4 TCP Congestion Control for ACK Packet Losses

The main reason of the unfairness among TCP flows described in Subsections 2.1 and 2.2 is that a TCP keeps growing the congestion window size even when the access point is highly congested and numerous ACK packets are discarded. Therefore, the author proposes an end-to-end basis modification to TCP congestion control mechanisms to alleviate the unfairness problems. The proposal is based on a simple idea: TCP should activate congestion control when detecting ACK packet losses, whereas the traditional TCP activates it only against data packet losses.

In detail, the proposed method activates congestion control when the number of ACK packet losses in a window exceeds the pre-determined threshold (*thresh_ack_losses*). The TCP sender detects ACK packet losses by monitoring the sequence number of received ACK packets. When the TCP sender observes abnormal jumps of the ACK sequence numbers, it is regarded as ACK packet losses in the network, calculated as follows:

$$ack\_loss = \max\left(\left\lfloor \frac{ack - prev\_ack}{MSS} \right\rfloor - 1,\ 0\right) \tag{8}$$

where ack_loss is the number of ACK packet losses, ack is the sequence number of ACK packet received currently (bytes), $prev\_ack$ is the sequence number of the last ACK packet received previously (bytes) and MSS is maximum segment size (bytes).

When the number of ACK packet losses in a RTT exceeds *thresh_ack_losses*, a TCP sender halves the congestion window ($cwnd$) and slow-start threshold $(ssthresh)$ is set to the halved congestion window. Note that before halving the window size, the TCP sender waits for one RTT to avoid false detection caused by the disorder of data packet reception at the TCP receiver. Additionally, when detecting data packet losses, a TCP sender stops checking the ACK packet losses for one RTT.

When using Equation (8) for detecting ACK packet losses, the effect of delayed ACK option [32] should be considered. Note that the delayed ACK option has been implemented in both Windows and Linux [33, 34]. Let b be the number of data packets acknowledged by a received ACK packet. Many TCP receiver implementations send one ACK packet for two consecutive packets received, so b is typically two. When delayed ACK option is enable, Equation (8) cannot estimate the number of ACK packet losses correctly. Therefore, TCP senders need to know b.

There are two possible methods to obtain the value of b: One is that a TCP receiver informs a TCP sender about b explicitly. The other is that a TCP sender estimates b without any explicit information from a TCP receiver. In the former method, a TCP sender obtains b accurately, but it requires modifications at TCP receiver. For this reason, the author employs the latter method without TCP receiver-side modifications.

In the proposed method, b is estimated as follows:

$$b_{est} = \left\lfloor sb_i + \frac{1}{2} \right\rfloor \tag{9}$$

$$sb_i = (1-\beta)\, sb_{i-1} + \beta \left(\frac{ack - prev\_ack}{MSS} \right) \tag{10}$$

where b_{est} is the estimated value of b and sb_i is the i th smoothed value for b with smoothing factor β. Note that sb_i in Equation (10) is a continuous value, but b_{est} in Equation (9) should be a discrete value since b should be a discrete value. Thus, b_{est} is calculated by half-adjust rounding. By using Equation (9), Equation (8) is transformed into:

$$ack\_loss = \left\lfloor \frac{ack - prev\_ack}{b_{est} \times MSS} \right\rfloor - 1. \tag{11}$$

Algorithm 1 shows the pseudo-code of the proposed method. Note that the proposed method can be utilized with arbitrary TCP modifications such as Compound TCP [35] and CUBIC [36], since the proposed method can be applied to such TCP variants without any ill-effect.

Algorithm 1 TCP congestion control for ACK packet losses

Initialization:

$prev\_ack \leftarrow$ the smallest sequence number of the unacknowledged packets (snd_una)

$sb \leftarrow MSS \times 2,\ b \leftarrow MSS \times 2$

$cnt\_ack\_loss \leftarrow 0$

$data\_loss \leftarrow 0,\ wait\_state \leftarrow 0$

On each ACK:

$sb \leftarrow (1-\beta)\, sb + \beta \left(\frac{ack - prev\_ack}{MSS} \right)$

$b_{est} \leftarrow \left\lfloor sb + \frac{1}{2} \right\rfloor$

$cnt\_ack\_loss \leftarrow \max \left(\left\lfloor \frac{ack - prev\_ack}{b_{est} \times MSS} \right\rfloor - 1 + cnt\_ack\_loss,\ 0 \right)$

$prev\_ack \leftarrow ack$

On each RTT:

if $wait\_state$ **then**

    if $cnt\_ack\_loss \geq thresh\_ack\_losses$ and not $data\_loss$ **then**

        $cwnd \leftarrow \max \left(\frac{cwnd}{2},\ 1 \right)$

        $ssthresh \leftarrow cwnd$

    end if

    $data\_loss \leftarrow 0$

    $cnt\_ack\_loss \leftarrow 0$

    $wait\_state \leftarrow 0$

else

    if $cnt\_ack\_loss \geq thresh\_ack\_losses$ or $data\_loss$ **then**

        $wait\_state \leftarrow 1$

    end if

end if

Packet loss:

$data\_loss \leftarrow 1$

Packet disorder:

$cnt\_ack\_loss \leftarrow \max(cnt\_ack\_loss - 1,\ 0)$

5 Performance Evaluation

5.1 Experimental Settings and Methodologies

Two experimental environments are shown in Figures 3 and 4, respectively. In both environments, 10 client stations share one access point. All client stations are located within 50 cm from the access point to avoid packet losses due to wireless link error. In Figure 3, a wired node is directly connected to the access point through a wired link. On the other hand, the experimental environment in Figure 4 introduces a PC router between the access point and the wired node for the purpose of evaluating in long delay environments. DELL Latitude E5500 laptops [37] and a DELL Precision 390 desktop [38] are used as the client stations and the wired node, respectively. All nodes, including the wired node, use Ubuntu 8.10 [39] (Linux kernel 2.6.28) as OS. Another DELL Precision 390 desktop is used as the PC router with netem [40] in Figure 4. Netem is a network emulator module applied to Linux kernel, which can emulate delay, packet loss, packet duplication and re-ordering at the PC router between two network interfaces. In the experiment, netem is utilized only for generating 50 ms delay to the wired link between the access point and the wired node. Unless otherwise noted, the experiments are conducted in the experimental environment of Figure 3.

Web100 [41] patch is utilized for collecting TCP connection information from the Linux kernel. Web100 is a Linux kernel patch, which adds a TCP instrument set to record TCP information by per connection basis, and it can provide TCP connection information such as congestion window size, RTT, the number of RTO occurred and so on, with a reasonable load of an observed computer. The author utilized TCP Reno version and implemented the proposed method described

Table 2: Wireless devices

(a) Wireless Interface Cards

Vendor	Product name
Buffalo	WLI-CB-AGHP
NEC	Aterm WL54AG

(b) Access Points

Vendor	Product name
Buffalo	WAPS-HP-AM54G54
NEC	Aterm WR8500N
Corega	CG-WLR300NNH

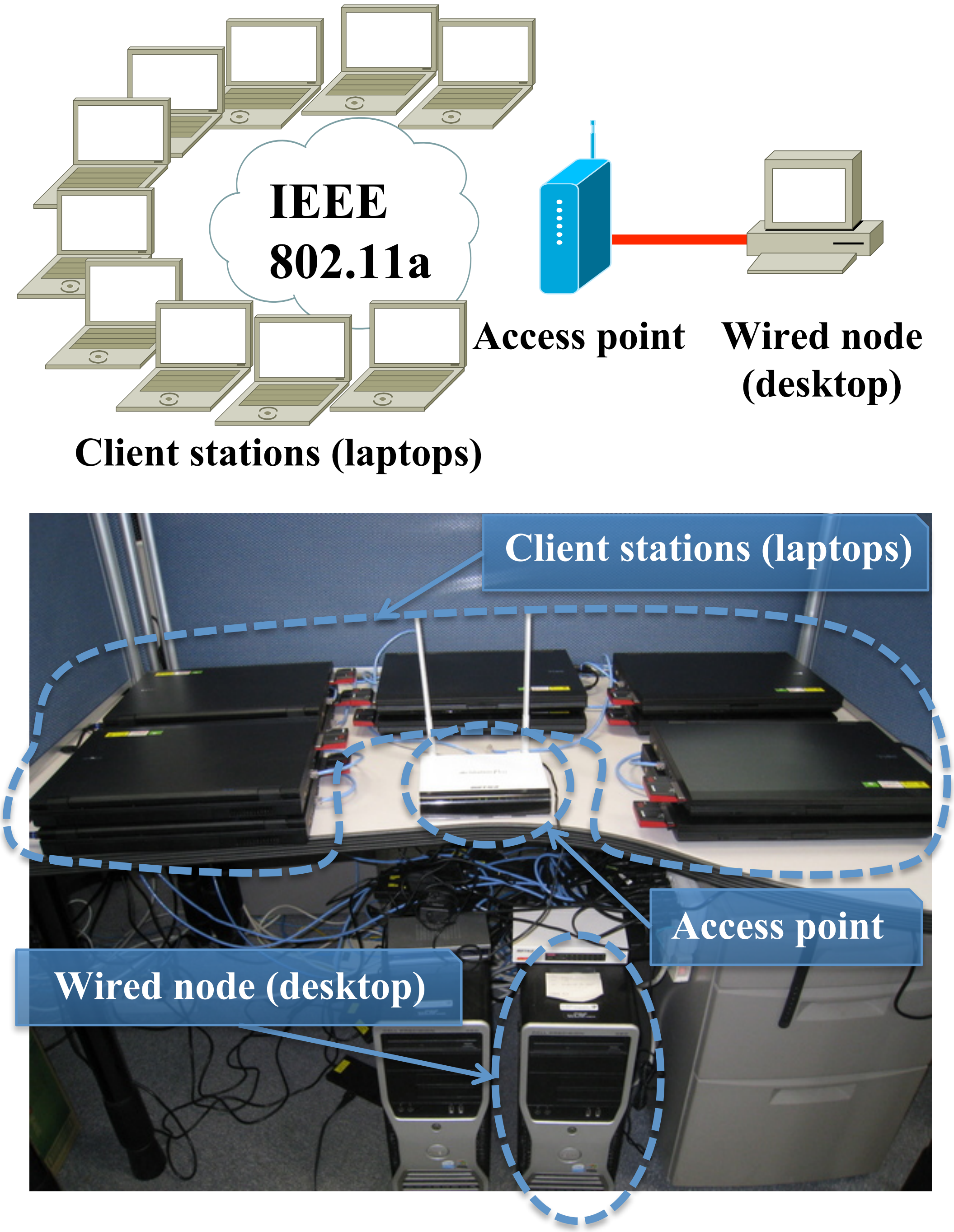


Figure 3: Experimental environment

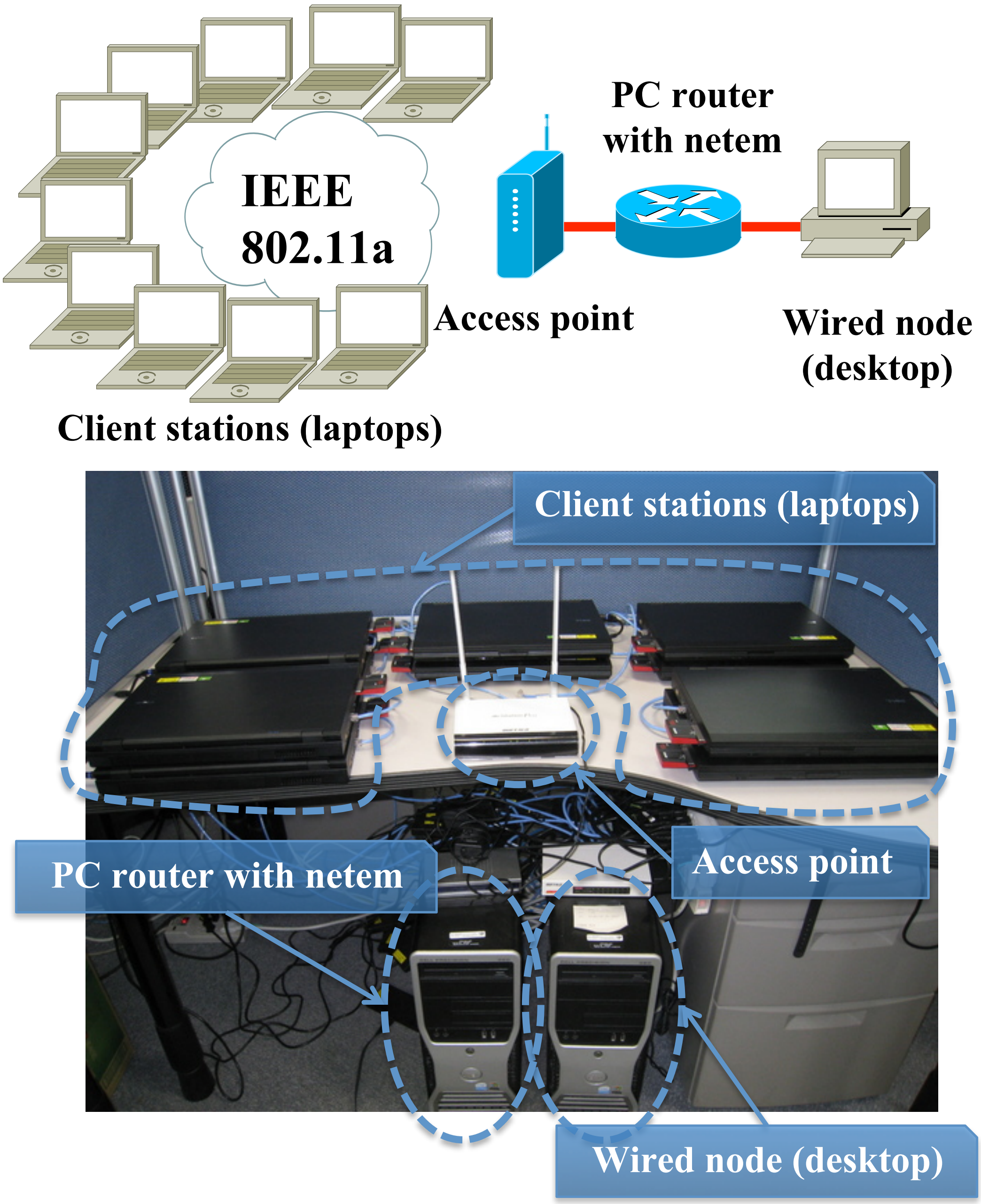


Figure 4: Experimental environment with a PC router

Table 3: Estimated buffer size of access points

Vendor	$minRTT$ [ms]	T [Mbps]	$cwnd_{overflow}$ [bytes]	B_{est} [packets]
Buffalo	0.975	21.8	118003	78.9
NEC	0.563	19.6	76719	51.5
Corega	0.674	20.2	366826	250.0

in Section 4 on the Linux code of TCP Reno.

Wireless devices in Table 2 are utilized as wireless interface card for client stations and access point. Note that all client stations utilize the same kind of wireless interface cards in Table 2(a) in each experiment. In what follows, access points and wireless interface cards are abbreviated as *[vendor name]-AP* and *[vendor name]-NIC*, respectively, e.g., Buffalo-AP and NEC-NIC.

The buffer size of access point impacts on flow's throughput and end-to-end delay [42]. However, access point vendors do not publish the buffer size in detail. For the reason, the author estimates the buffer size through the simple experiments with a single TCP connection, with an estimation equation as follows:

$$B_{est} = \frac{\frac{cwnd_{overflow}}{8} - \frac{T}{minRTT}}{MSS} \tag{12}$$

where B_{est} is the estimated buffer size of access point (packets), $cwnd_{overflow}$ is the congestion window when buffer overflow occurs at the access point and only one TCP connection exists in the network (bytes), T is throughput just before buffer overflow is occurred (bps), and $minRTT$ is a minimum RTT (seconds). Table 3 presents the results of estimated buffer size of each access point. Note that the experiments are conducted ten times for averaging the results.

The experiments using environments in Figures 3 and 4 are conducted as follows. Only one TCP flow is generated for each client station using by Iperf [43], assuming bulk data transfer. The author keeps the number of concurrent TCP flows to be ten, and changes the ratio of upstream and downstream TCP flows from (0, 10) to (10, 0). TCP connections use either the proposed method on TCP Reno or conventional TCP Reno for comparison purposes. The parameter *thresh_ack_losses* in the proposed method is set to one. The experiment time is set to 180 seconds and each TCP connection is generated simultaneously when the experiment starts. The author disabled vendor-specific functions implemented at access points. Unless otherwise noted, TCP delayed ACK option was disabled at TCP receivers. For each experimental setting, the experiments are conducted ten times for averaging the results.

5.2 Evaluation Metric

The effectiveness of the proposed method is evaluated from the viewpoint of fairness and bandwidth utilization by assessing the throughput of each flow, Jain's fairness index and the index proposed in Section 3.

To evaluate fairness and trade-off relationships between fairness and bandwidth utilization, Sliding Window Method (SWM) function [44] is applied to the indexes. The SWM can give a quantitative measure of fairness over a wide range of time scales and it has an advantage of measuring *short-term fairness* and *long-term fairness* at the same time. Intuitively, short-term fairness of a data transmission flow refers to its ability to provide equitable access to resources to all the contending flows over short time scales. In contrast, long-term fairness measures the average amount of resources assigned over a longer time.

The SWM function which applied to Jain's fairness index (Equation (1)) is as follows:

$$SWM_j(w) = \frac{(\sum_{i=1}^{n} x_i(w))^2}{n \sum_{i=1}^{n} x_i(w)^2} \tag{13}$$

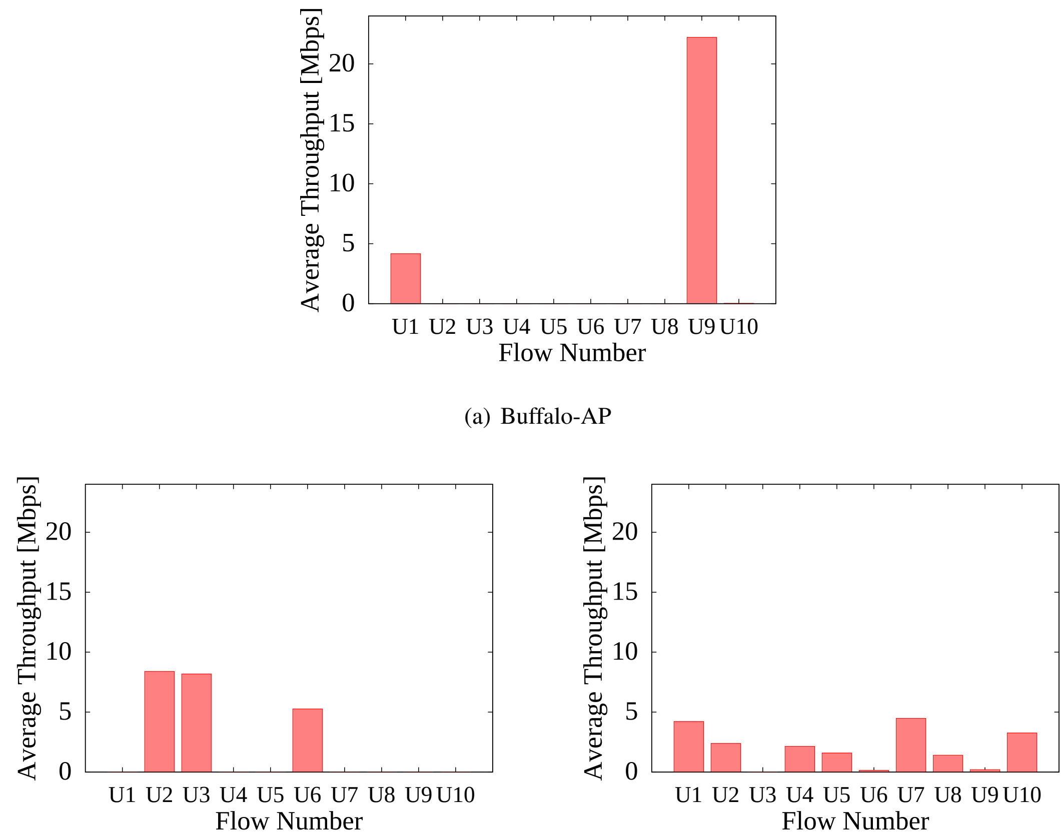
where w is a time-window size for evaluating the fairness and $x_i(w)$ is an average throughput of flow i in a time-window w. The SWM function which applied to Equation (7) is as follows:

$$SWM(w) = \frac{C^2}{n \sum_{i=1}^{n} x_i(w)^2 - 2C \sum_{i=1}^{n} x_i(w) + 2C^2} \left(\frac{\sum_{i=1}^{n} x_i(w)}{C} \right)^{\alpha}. \tag{14}$$

5.3 Experimental Results and Discussions

5.3.1 Evaluation of Fairness and Bandwidth Utilization with Each Flow's Throughput

The author first presents the snapshot results in Figures 5 and 6 where average throughput of 10 upstream flows using TCP Reno and the proposed method, respectively, where the three kinds of access point are utilized. In both figures, Buffalo-NICs are utilized for all stations. The average throughput is calculated using the amount of data transmitted in 50-180 seconds in the experiment. Figure 5 shows that serious throughput unfairness occurs regardless of vendors of the access points. In detail, some upstream flows occupy the network bandwidth and other flows are completely starved. From the viewpoint of fairness among flows, Corega-AP gives better results

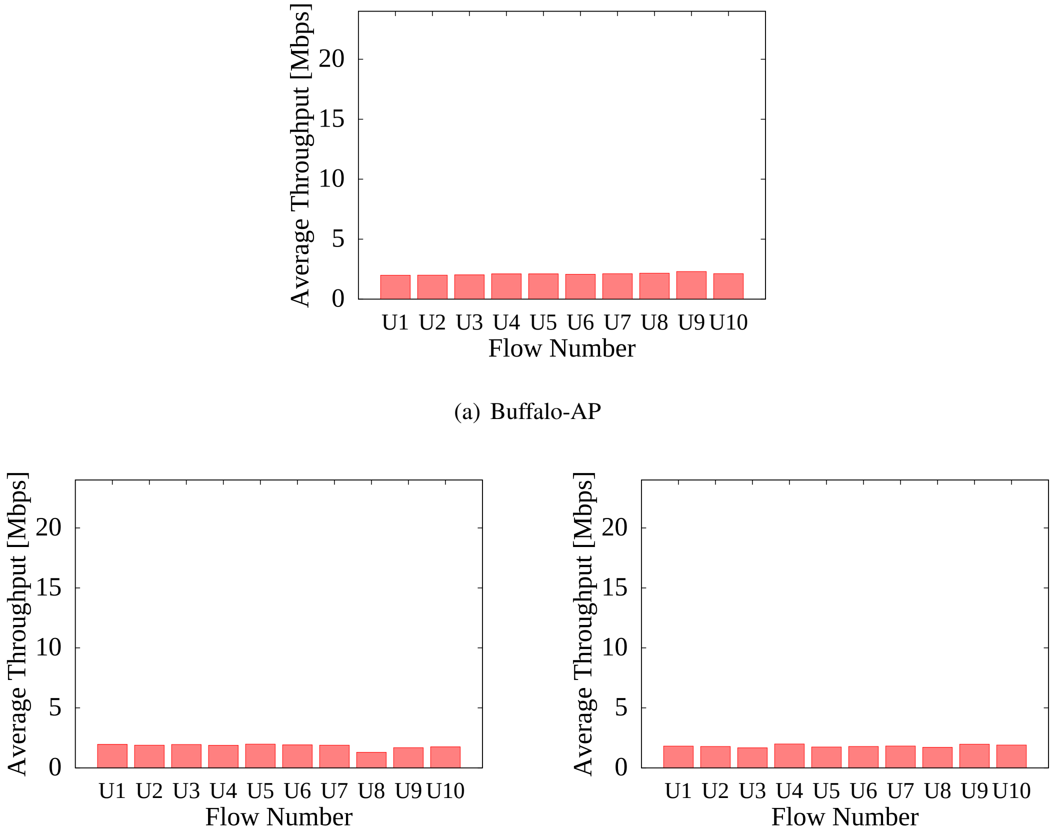


(a) Buffalo-AP

(b) NEC-AP

(c) Corega-AP

Figure 5: Average throughput of 10 upstream flows using TCP Reno when using Buffalo-NIC



(a) Buffalo-AP

(b) NEC-AP

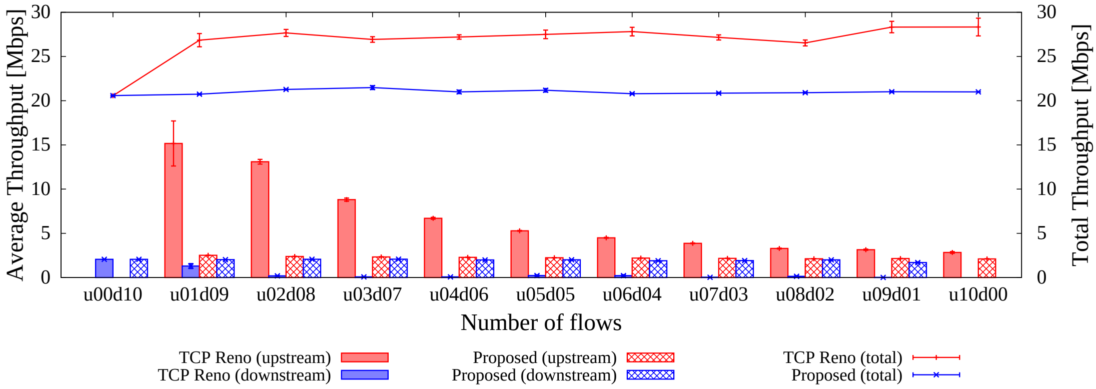
(c) Corega-AP

Figure 6: Average throughput of 10 upstream flow using the proposed method when using Buffalo-NIC

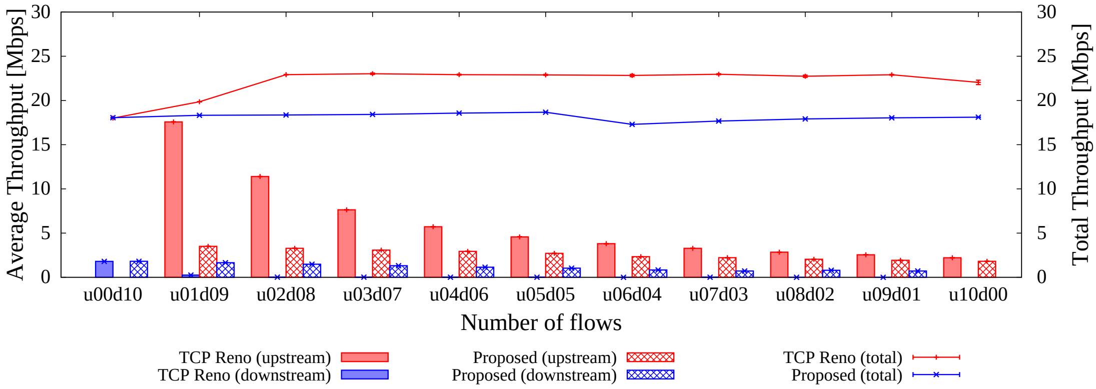
than the other access points. In other words, the number of starved flows when using Corega-AP is smaller than that when using the other access points. This is because of the difference in the buffer size at the access points: The buffer size of Corega-AP is larger than that of the other access points as shown in Table 3. On the other hand, Figure 6 shows that the proposed method successfully alleviates throughput unfairness among upstream TCP flows, regardless of the access point products.

Figure 7 presents the average throughput of upstream and downstream flows and the total throughput when changing the ratio of upstream and downstream flows and using Buffalo-NICs and three types of access points. In the figure, uxdy means that the number of upstream and downstream TCP flows are x and y, respectively. Figure 7 shows that, using TCP Reno, when at least one upstream TCP flow exists in the network, the upstream flows occupy the almost all network bandwidth and downstream flows are starved. On the other hand, the proposed method can significantly improve the throughput fairness between upstream and downstream flows and there is no starved flow. In Figure 7(b), we can observe that the degree of fairness improvement is small when using NEC-AP. The reason of this is as follows. In this case, the flows experience RTO even when the proposed method is utilized and using NEC-AP, whereas the proposed method brings no RTO when using the other access points. The reason of this is that the buffer size of the NEC-AP is smaller than the other access points as shown in Table 3. However, even in this case the proposed method can avoid the starvation of flows.

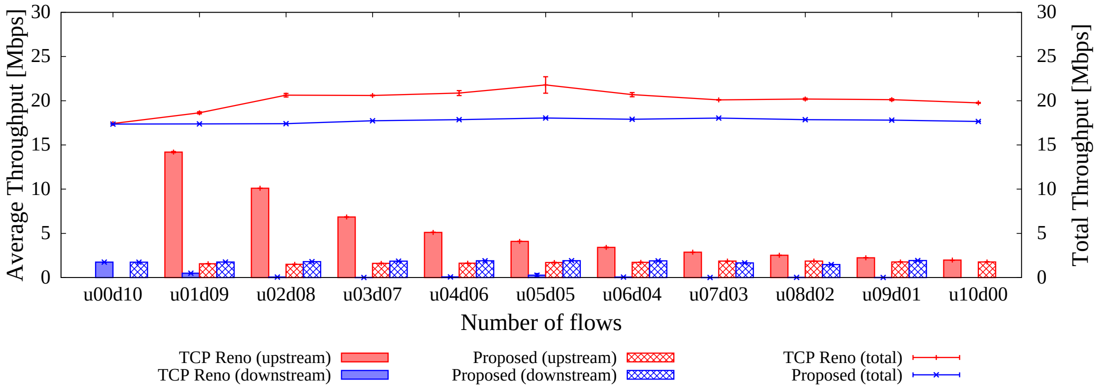
In terms of total throughput, we can observe that total throughput of 10 downstream flows with TCP Reno and the proposed method are equivalent regardless of kinds of access point. This is because that ACK packets are not discarded at the access points and the behaviors of TCP with and without the proposed method are identical. However, when one or more upstream flows exist in the network, the proposed method degrades the total throughput while the fairness improves significantly. The reason is explained as follows. When the proposed method is not utilized, many ACK packets of upstream TCP flows are discarded at the access points. This means that the number of data packets and ACK packets in the WLAN is not balanced, and the number of data packets which are transmitted in the WLAN increases. However, when using the proposed method, the number of data packets and ACK packets would be balanced since the proposed method activates congestion control against ACK packet losses. The author should note that the total throughput would increase when deactivating the proposed method, but the increased throughput is distributed



(a) Buffalo-AP



(b) NEC-AP



(c) Corega-AP

Figure 7: Effect of the number of upstream and downstream flows when using Buffalo-NIC

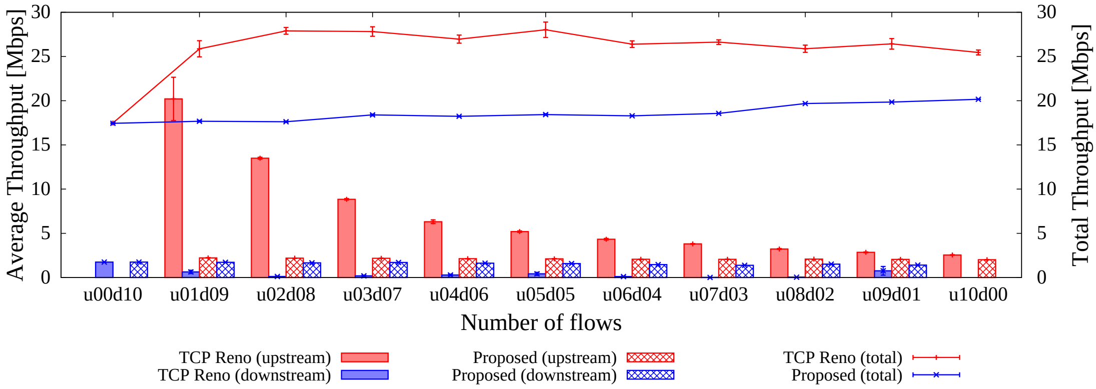
only to non-starved flows and the fairness among flows further degrades. In other words, there is a trade-off relationship between fairness and bandwidth utilization in the proposed method.

Figure 8 presents the results when using NEC-NICs instead of Buffalo-NICs. When comparing Figures 7 and 8, the proposed method is effective regardless of the kind of wireless interface cards in terms of fairness between upstream and downstream flows. Also, the degree of unfairness among TCP flow depends on properties of the access point such as buffer size rather than that of the wireless interface cards.

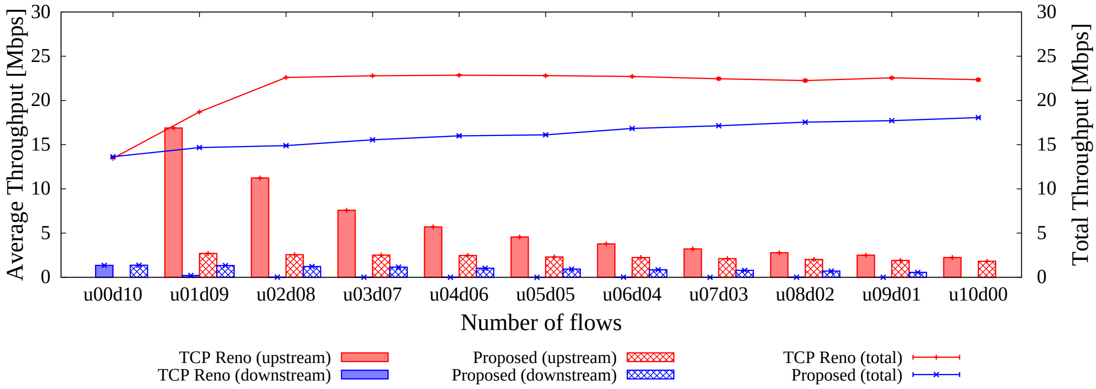
5.3.2 Fairness Evaluation with Jain's Index

Figures 9, 10 and 11 present the evaluation results with Jain's fairness index with SWM when using Buffalo-AP, NEC-AP, and Corega-AP, respectively, and Buffalo-NICs. The index value are identical in the cases with or without proposed method when there is no upstream flow as shown in Figures 9(a), 10(a) and 11(a). It implies that the proposed method is not activated since no ACK packet loss occurs. On the other hand, comparing the proposed method with original TCP Reno when upstream flows exist, original TCP Reno significantly degrade fairness not only among upstream TCP flows but also between upstream and downstream TCP flows, whereas the proposed method can achieve the same amount of fairness as that when no upstream flow exists. However, when using NEC-AP, the degree of fairness improved by the proposed method is small due to the little of the access point buffer as described in Subsection 5.3.1. When using Corega-AP, Figure 11(c) indicates that the degree of fairness among upstream TCP flows with original TCP Reno is better than when using the other access points. This is due to a large buffer size at the access point as shown in Table 3.

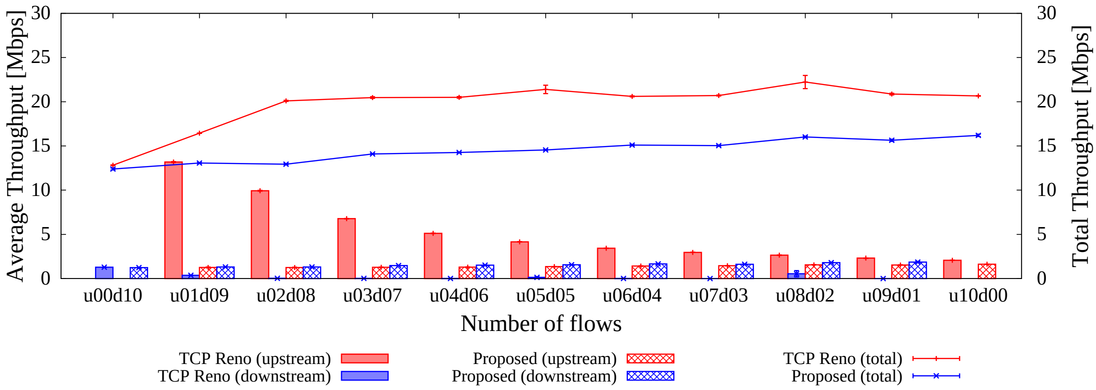
Figures 12, 13, and 14 show the corresponding results to Figures 9, 10, and 11 when using NEC-NICs. Figures 9, 10 and 11, and Figures 12, 13 and 14 are quite similar from the viewpoint of Jain's fairness index. It implies that vendors of wireless interface cards have little impact on the fairness among TCP flows.



(a) Buffalo-AP

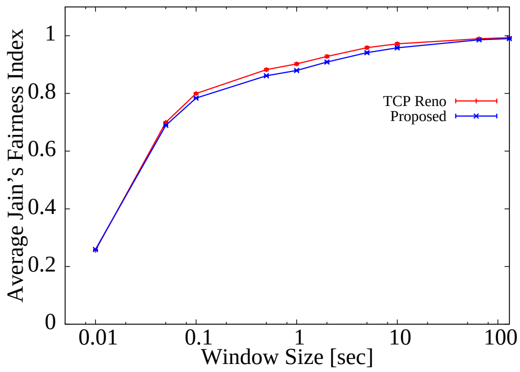


(b) NEC-AP

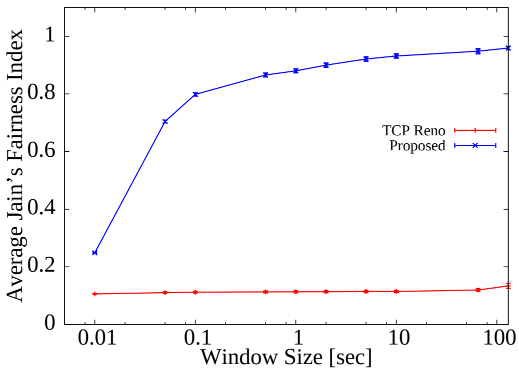


(c) Corega-AP

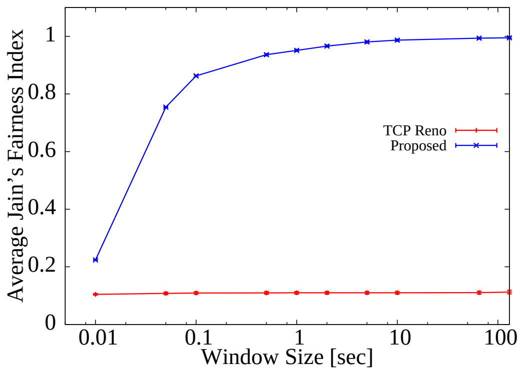
Figure 8: Effect of the number of upstream and downstream flows when using NEC-NIC



(a) 10 downstream flows

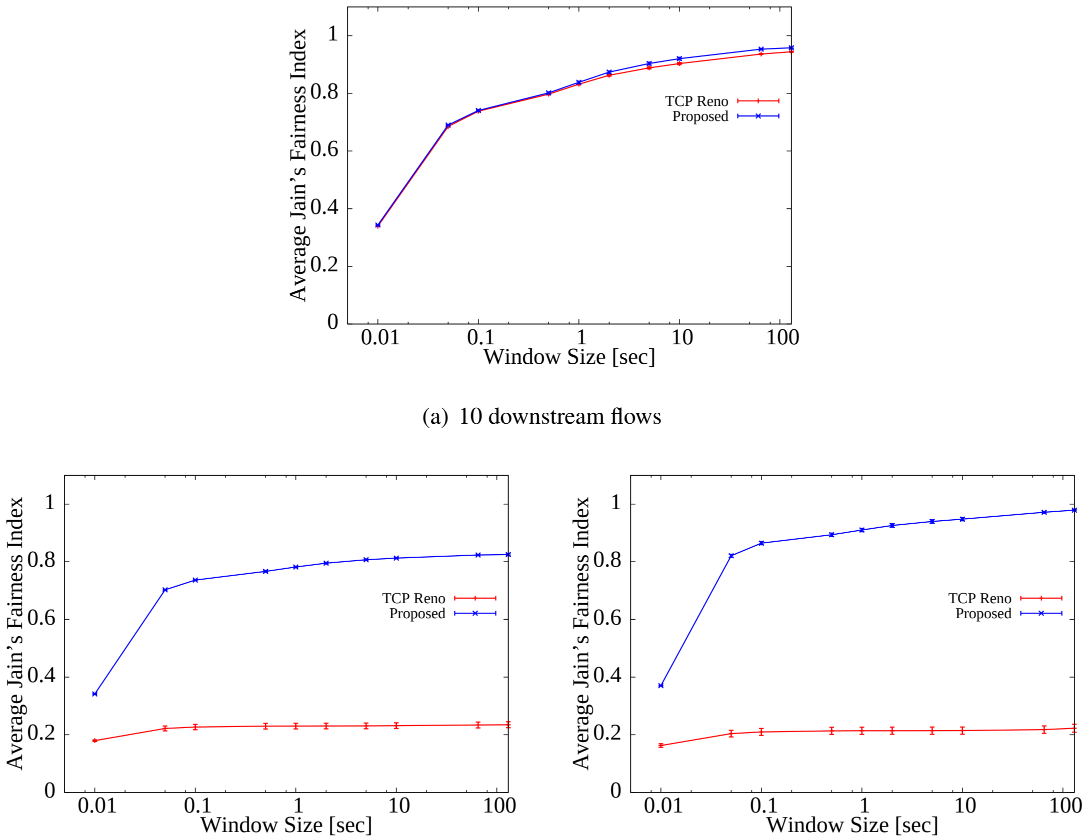


(b) 5 upstream flows and 5 downstream flows



(c) 10 upstream flows

Figure 9: Jain's fairness index with SWM when using Buffalo-AP and Buffalo-NIC

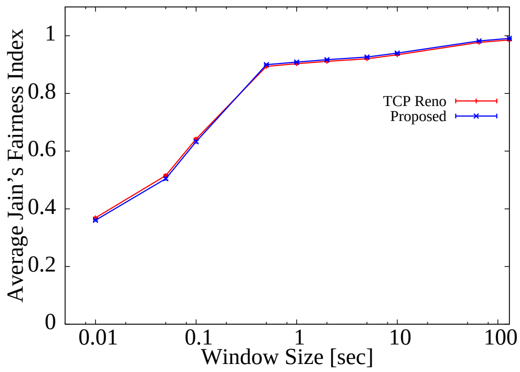


(a) 10 downstream flows

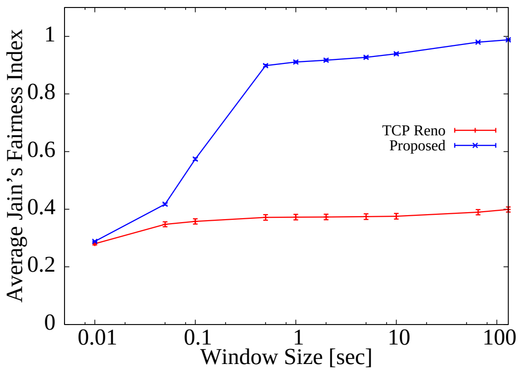
(b) 5 upstream flows and 5 downstream flows

(c) 10 upstream flows

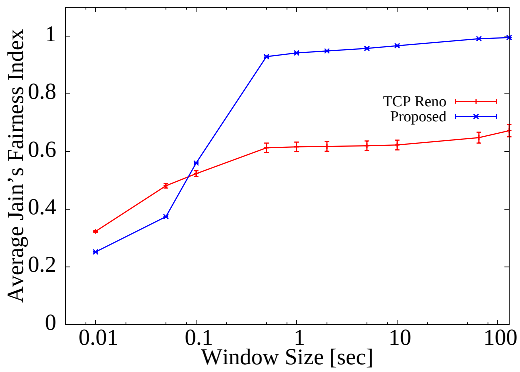
Figure 10: Jain's fairness index with SWM when using NEC-AP and Buffalo-NIC



(a) 10 downstream flows

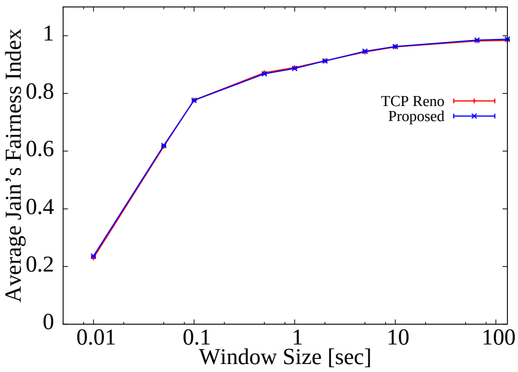


(b) 5 upstream flows and 5 downstream flows

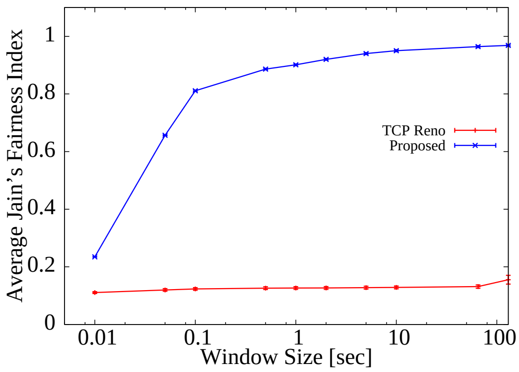


(c) 10 upstream flows

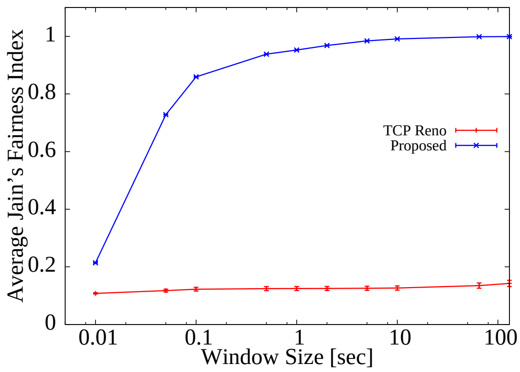
Figure 11: Jain's fairness index with SWM when using Corega-AP and Buffalo-NIC



(a) 10 downstream flows

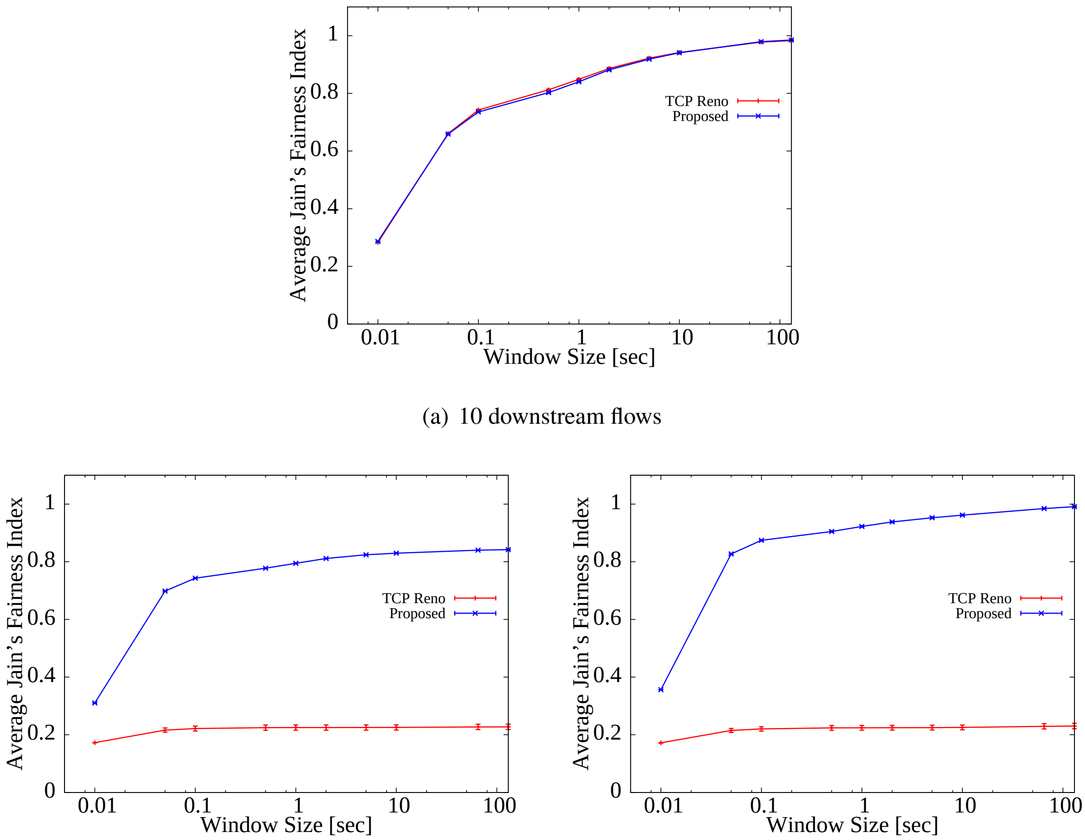


(b) 5 upstream flows and 5 downstream flows



(c) 10 upstream flows

Figure 12: Jain's fairness index with SWM when using Buffalo-AP and NEC-NIC

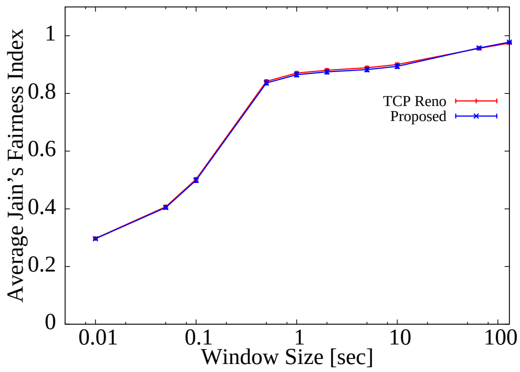


(a) 10 downstream flows

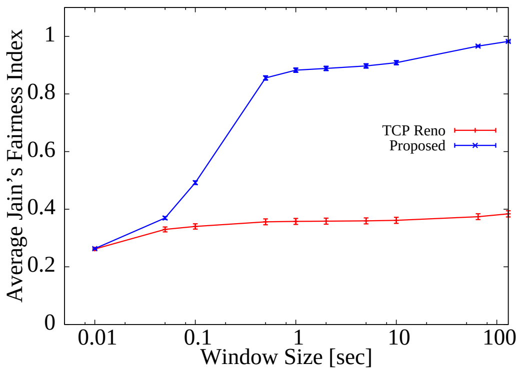
(b) 5 upstream flows and 5 downstream flows

(c) 10 upstream flows

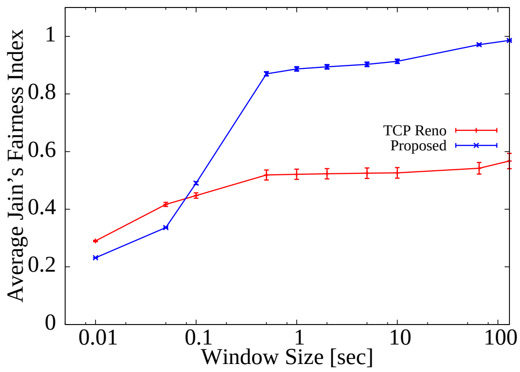
Figure 13: Jain's fairness index with SWM when using NEC-AP and NEC-NIC



(a) 10 downstream flows



(b) 5 upstream flows and 5 downstream flows



(c) 10 upstream flows

Figure 14: Jain's fairness index with SWM when using Corega-AP and NEC-NIC

5.3.3 Trade-Off Evaluation between Fairness and Bandwidth Utilization with Proposed Index

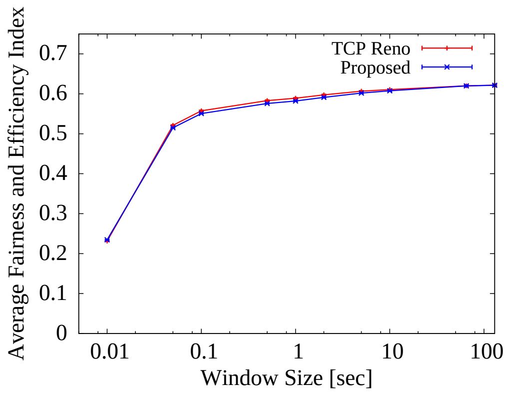
Figures 15, 16 and 17 present the evaluation results with the proposed index with SWM when using Buffalo-AP, NEC-AP and Corega-AP, respectively, and using Buffalo-NICs. The parameter α is set to one and C in Equation (14) is set to 29.60 Mbps according to the theoretical maximum throughput of IEEE 802.11a WLAN with 1460 bytes MTU [45].

These figures exhibit that the index values are identical in the cases with or without the proposed method when there is no upstream flow (Figures 15(a), 16(a) and 17(a)). The reason of this is the same as in Figures 9(a), 10(a), and 11(a), meaning that the proposed method is not activated since there is no ACK packet loss. On the other hand, when using Buffalo-AP or NEC-AP, the index values of the proposed method are quite better than that of original TCP Reno in terms of not only long-term fairness but also short-term fairness when one or more upstream flows exist in the network. However, when using Corega-AP, the index values of the proposed method are almost identical to that of the normal TCP Reno. The reason of this is as follows. The degree of unfairness when using Corega-AP is better than that when using the other access points described in Subsection 5.3.1 (Figure 5(c)). In terms of total throughput, the proposed method is smaller than original TCP Reno (Figures 7 and 8) when one or more upstream flows exist. Therefore, in terms of trade-off relationship between fairness and bandwidth utilization, the proposed method and original TCP Reno are almost identical.

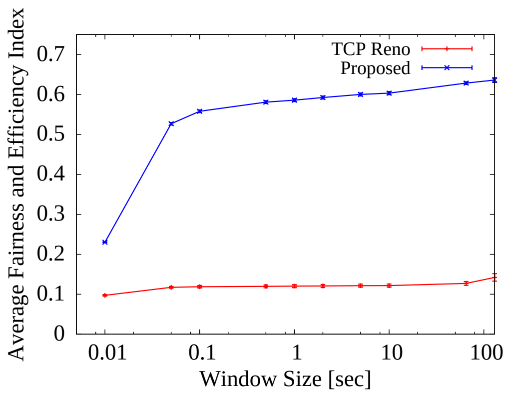
Figures 18, 19 and 20 show the corresponding results to Figures 15, 16, and 17 using NEC-NICs. Note that when comparing the results using Buffalo-NIC with those using NEC-NIC, the index values of the latter case is smaller than that of the former case, whereas Jain's fairness index of the latter case and that of the former case are identical as shown in Subsection 5.3.2. This is because of the difference in the total bandwidth utilization and the proposed index can consider the effect of bandwidth utilization as well as fairness among flows.

5.3.4 Effect of estimating delayed ACK parameter

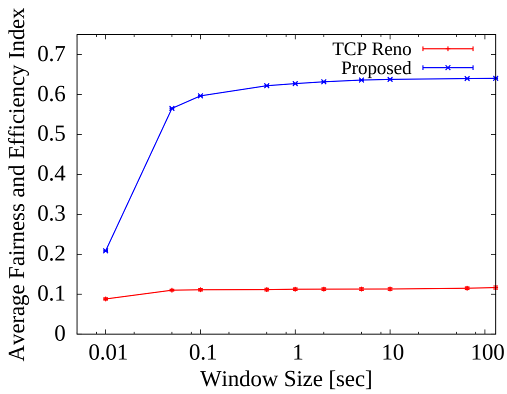
To investigate the effect of Equation (9) for estimating the parameter in delayed ACK option, the author conducted the experiment with or without delayed ACK option. Figure 21 shows the average throughput of upstream and downstream flows and the total throughput when changing the



(a) 10 downstream flows

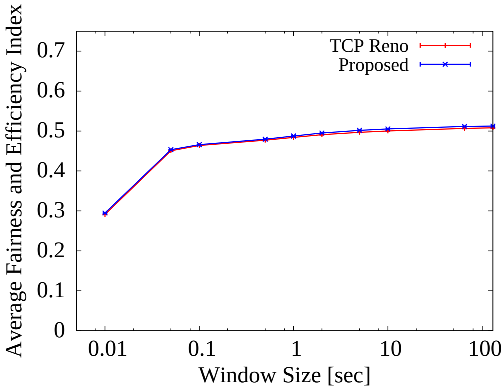


(b) 5 upstream flows and 5 downstream flows

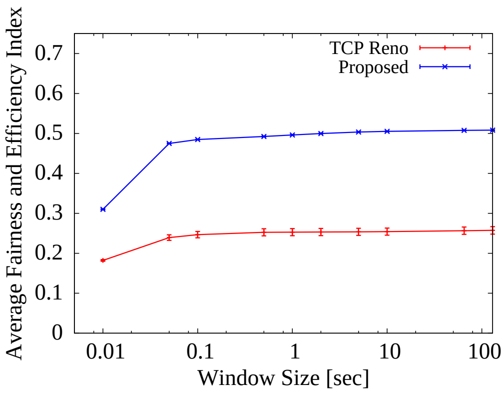


(c) 10 upstream flows

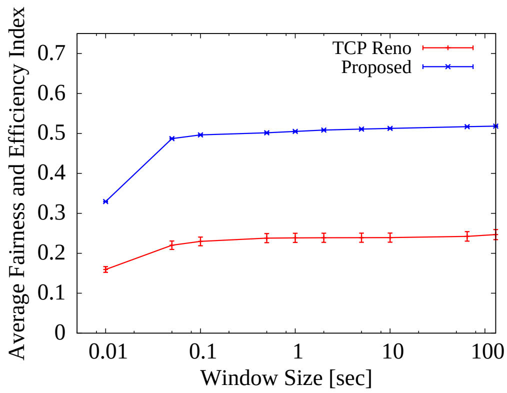
Figure 15: Proposed index with SWM when using Buffalo-AP and Buffalo-NIC



(a) 10 downstream flows

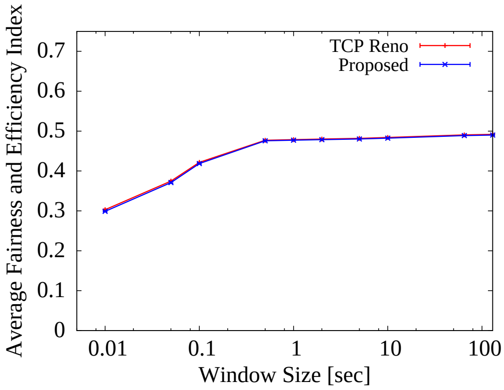


(b) 5 upstream flows and 5 downstream flows

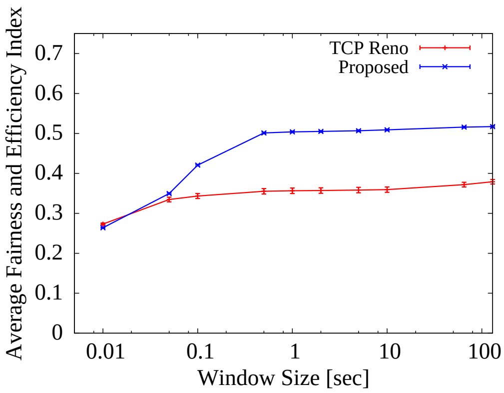


(c) 10 upstream flows

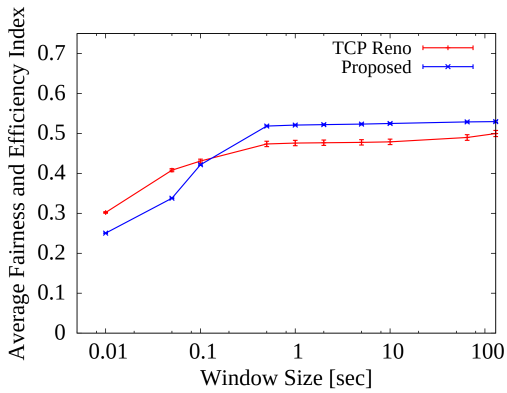
Figure 16: Proposed index with SWM when using NEC-AP and Buffalo-NIC



(a) 10 downstream flows

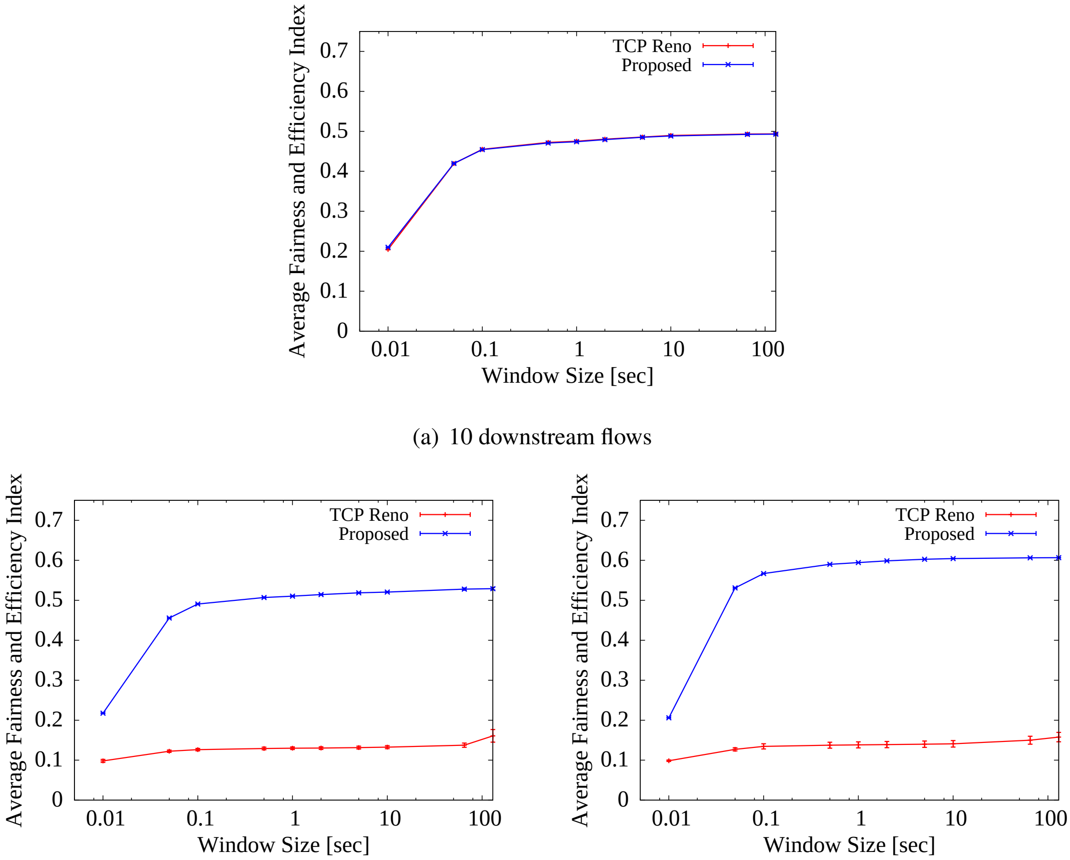


(b) 5 upstream flows and 5 downstream flows



(c) 10 upstream flows

Figure 17: Proposed index with SWM when using Corega-AP and Buffalo-NIC

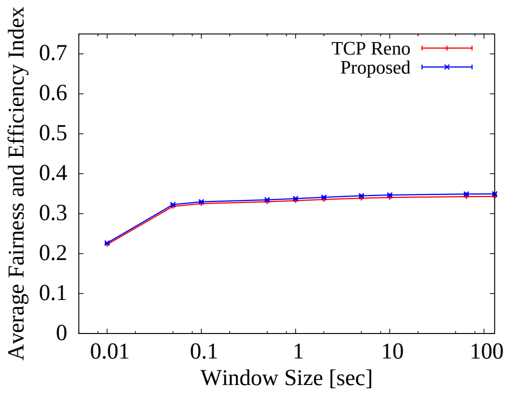


(a) 10 downstream flows

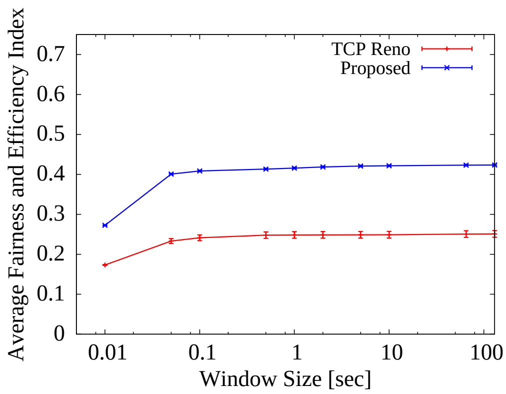
(b) 5 upstream flows and 5 downstream flows

(c) 10 upstream flows

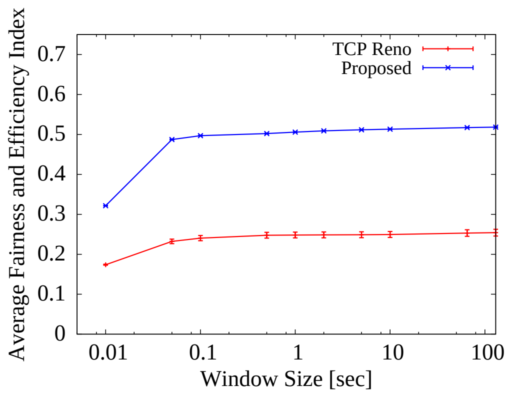
Figure 18: Proposed index with SWM when using Buffalo-AP and NEC-NIC



(a) 10 downstream flows

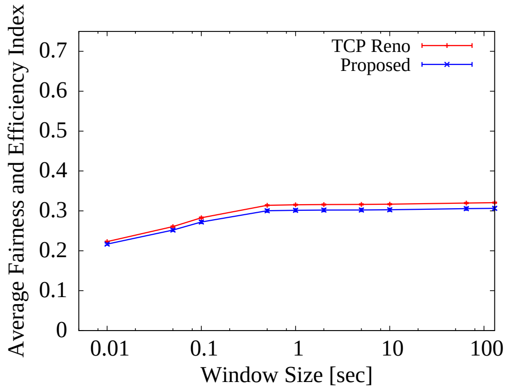


(b) 5 upstream flows and 5 downstream flows

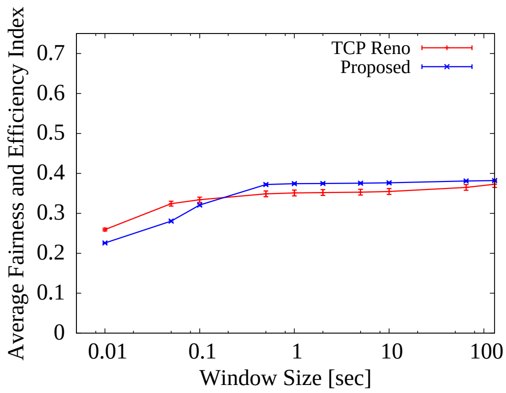


(c) 10 upstream flows

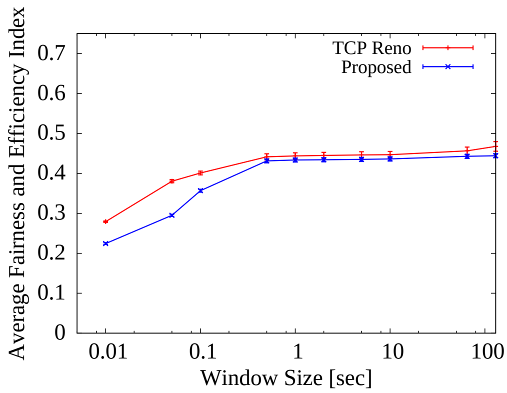
Figure 19: Proposed index with SWM when using NEC-AP and NEC-NIC



(a) 10 downstream flows



(b) 5 upstream flows and 5 downstream flows



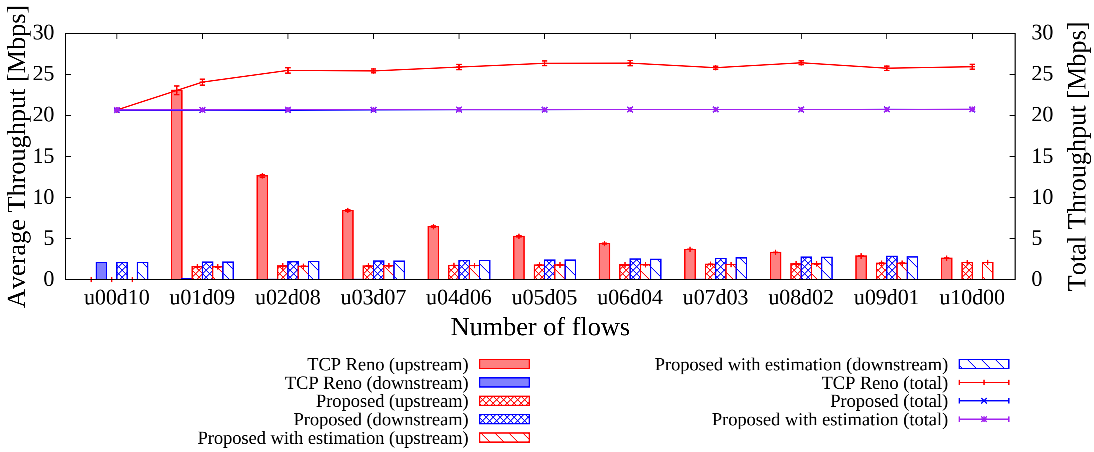
(c) 10 upstream flows

Figure 20: Proposed index with SWM when using Corega-AP and NEC-NIC

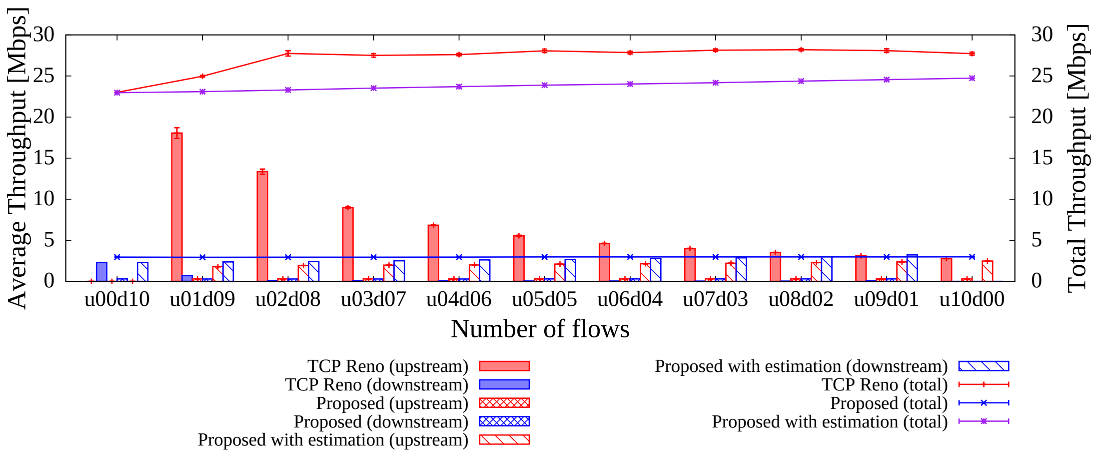
ratio of upstream and downstream flows in the experimental environment in Figure 4 with 50 ms delay, where Buffalo-AP and Buffalo-NIC are used. When the proposed method is not utilized the estimation with Equation (9), It utilizes Equation (8) to estimate the number of ACK packet losses, assuming that value of b is one. In Figure 21, the proposed method with and without the estimation are labeled as *Proposed with estimation* and simply *Proposed*, respectively.

In Figure 21(a), we can observe that the proposed method with or without the estimation show the same performance when delayed ACK is disabled. This means that the estimation by using Equation 9 can estimate value of b accurately, and the behaviors of the proposed method with or without the estimation of b are identical. On the other hand, when delayed ACK is enabled, Figure 21(b) shows that the proposed method without the estimation significantly degrades the throughputs. The reason of this is that the proposed method without the estimation regards the number of ACK packets decreased by delayed ACK as ACK packet losses due to the buffer overflow at the access point, and it keeps small congestion window. On the other hand, the proposed method with the estimation of b value can significantly improve the fairness between upstream and downstream TCP flows as when delayed ACK is disabled. Furthermore, when comparing the total throughput in Figures 21(a) and 21(b), we can observe that the total throughput increase when using delayed ACK option with the original TCP Reno or the proposed method with the estimation. This is because the delayed ACK option decreases the number of ACK packets in wireless channel and it consequently increases the number of data packets in wireless channel.

Figure 22 shows the results of the same situation as Figure 21 with or without delayed ACK option. When delayed ACK option is enabled, the index values with original TCP Reno are better than that when delayed ACK option is disabled. The reason is as follows. Because delayed ACK option decreases the number of ACK packets generated at TCP receivers, the access point can buffer more data packets. Although that improve the fairness, the improvement degree is limited. That is, delayed ACK option for alleviating the unfairness is not a silver bullet. On the other hand, the proposed methods with and without the estimation in Equation (9) when delayed ACK option is enabled are almost identical. This means that the estimation of delayed ACK parameter does not degrade fairness property of the proposed method. However, the proposed method without the estimation when delayed ACK option gives almost complete fairness, whereas the total throughput is significantly lower than that of TCP Reno, as mentioned above (Figure 21). That is, Jain's fairness index is not suitable for evaluating such situation.

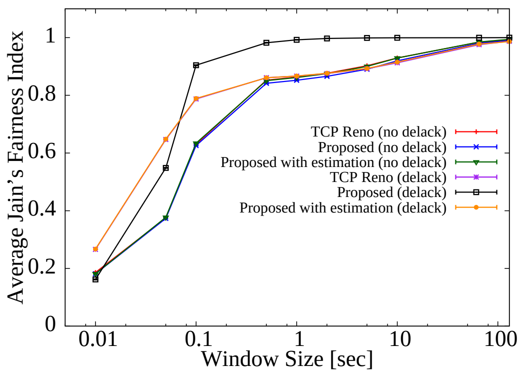


(a) Delayed ACK is disabled

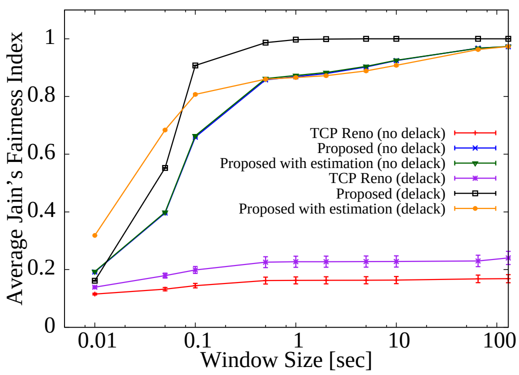


(b) Delayed ACK is enabled

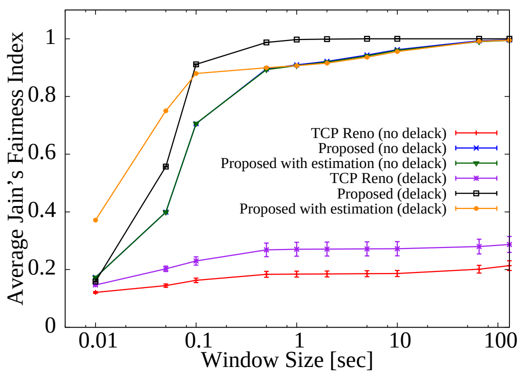
Figure 21: Effect of delayed ACK option when using Buffalo-NIC and Buffalo-AP with 50 ms one-way delay



(a) 10 downstream flows



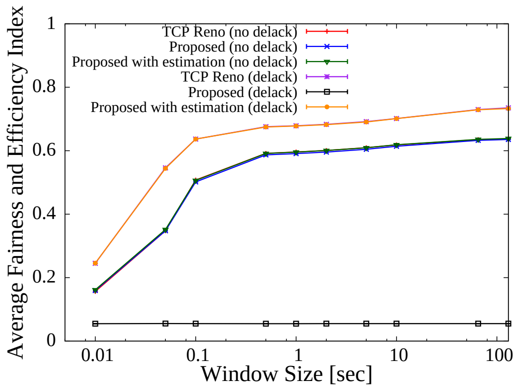
(b) 5 upstream flows and 5 downstream flows



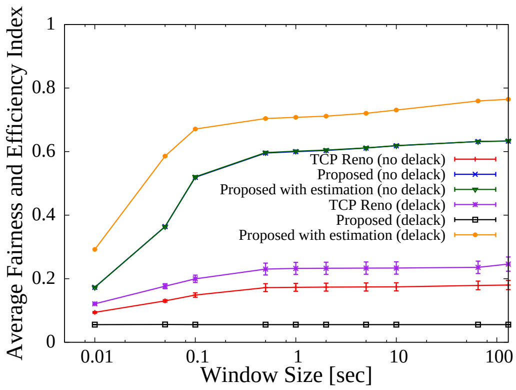
(c) 10 upstream flows

Figure 22: Jain's fairness index with SWM when using Buffalo-NIC and Buffalo-AP with 50 ms one-way delay

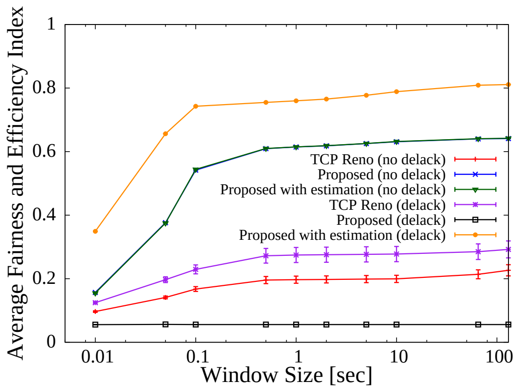
Figure 23 shows the evaluation results with the proposed index in the same situation as Figure 22. The proposed method and normal TCP Reno are identical when there is no upstream TCP flow with or without that delayed ACK option is enabled (Figure 23(a)). On the other hand, the proposed method without the estimation in Equation 9 has the worst value in Figure 23 because of the poor utilization described in Figure 21(b) despite almost perfect fairness. Paying attention to index value, the index values of the proposed method and TCP Reno when delayed ACK option is enabled are better than that when delayed ACK option is disabled. The reason of this is that the bandwidth utilization improves by enabling delayed ACK option, as described in Subsection 5.3.1. Furthermore, the proposed method with the estimation when delayed ACK option is enabled has the best values of the index. It means that the proposed method can enhance throughput by using delayed ACK option without degrading effectiveness of fairness improvements.



(a) 10 downstream flows



(b) 5 upstream flows and 5 downstream flows



(c) 10 upstream flows

Figure 23: Proposed index with SWM when using Buffalo-NIC and Buffalo-AP with 50 ms one-way delay

6 Conclusion

In this thesis, the author first proposed a novel performance index, considering the trade-off relationships between per-flow fairness and bandwidth utilization at network bottleneck. The proposed index is based on the variations in throughput of concurrent flows and the ideal throughput distribution where all flows achieve the same amount of throughput and network bandwidth is fully utilized.

The author next proposed the transport-layer solution for alleviating the unfairness between upstream and downstream TCP flows, and among upstream TCP flows in WLAN. The proposed method alleviates the unfairness by detecting TCP ACK packet losses as an indication of the congestion at access point. It required small modification on TCP congestion control mechanisms only on stations in WLAN.

Through the extensive experiments using real WLAN environments with the products from several vendors, the author revealed the proposed method is effective not only for TCP fairness among upstream flows but also for fairness between upstream and downstream flows regardless of the number of upstream and downstream flows, while slightly degrading the total throughput. The author also presented that the proposed method has a small effect on alleviating the unfairness is when the access point has the small buffer. However, the proposed method avoids starving any flows even in such cases. Furthermore, the author revealed that when utilizing delayed ACK option, the proposed method could enhance the total throughput without degrading the effectiveness of fairness improvements. By using the proposed metric, the author showed that the proposed method could take quite better trade-off between fairness and throughput regardless of vendor implementations of wireless access points and wireless interface cards.

For future work, the author plans to investigate the performance of the proposed method in environments where the unfairness in WLAN is not occurred such as wired network because the proposed method may activates congestion control by failed detection in such environment.

Acknowledgements

I would like to greatly appreciate to Professor Hirotaka Nakano of Osaka University, for his advices on my study from diversified standpoints.

I especially would like to express my deepest gratitude to Associate Professor Go Hasegawa of Osaka University. He has always given me a number of to-the-point advices and feedback. They helped me to consider carefully not only my work but also my future. Without his extensive support during the past three years, I would not have achieved this thesis.

My deepest appreciation goes to Professor Masayuki Murata of Osaka University for his invaluable comments, helpful guidance and continuous supports. They were vital to achieve my research.

I appreciate to Assistant Professor Yoshiaki Taniguchi of Osaka University, who gave me useful supports.

I owe a lot to my friends and colleagues in the Department of Information Networking of the Graduate School of Information Science and Technology, Osaka University. Our conversations and discussions greatly helped me to advance my study. In particular, Kazuhito Matsuda gave me useful advices for my research. I wish to say thank you to him again.

Finally, I express my thanks to my parents for their constant encouragement to date.

References

[1] IEEE 802.11b, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*. IEEE, Feb. 1999.

[2] IEEE 802.11a, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band*. IEEE, Feb. 1999.

[3] IEEE 802.11g, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band*. IEEE, June 2003.

[4] IEEE 802.11n, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput*. IEEE, Oct. 2009.

[5] X. Lin, X. Chang, and J. K. Muppala, "VQ-RED: An efficient virtual queue management approach to improve fairness in infrastructure WLAN," in *Proceedings of the 5th International IEEE Workshop on Wireless Local Networks (WLN 2005)*, pp. 632–638, Nov. 2005.

[6] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003)*, vol. 2, pp. 863–872, Mar. 2003.

[7] F. Keceli, I. Inan, and E. Ayanoglu, "TCP ACK congestion control and filtering for fairness provision in the uplink of IEEE 802.11 infrastructure basic service set," in *Proceedings of IEEE International Conference on Communications*, pp. 4512–4517, June 2007.

[8] F. Keceli, I. Inan, and E. Ayanoglu, "Fair and efficient TCP access in IEEE 802.11 WLANs," in *Proceedings of Wireless Communications and Networking Conference 2008 (WCNC 2008)*, pp. 1745–1750, Apr. 2008.

[9] J. Ha and C.-H. Choi, "TCP fairness for uplink and downlink flows in WLANs," in *Proceedings of IEEE Global Telecommunications Conference 2006 (GLOBECOM 2006)*, pp. 1–5, Nov. 2006.

[10] S. Mangold, S. Choi, G. R. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS support in wireless LANs," *IEEE Wireless Communications*, vol. 10, pp. 40–50, Dec. 2003.

[11] WiFi Alliacne. available at `http://www.wi-fi.org/`.

[12] IEEE 802.11e, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service Enhancements*. IEEE, Oct. 2005.

[13] C. Casetti and C. F. Chiasserini, "Improving fairness and throughput for voice traffic in 802.11e EDCA," in *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2004 (PIMRC 2004)*, pp. 525–530, Sept. 2004.

[14] Y. Fukuda and Y. Oie, "Unfair and inefficient share of wireless LAN resource among uplink and downlink data traffic and its solution," *IEICE Transactions on Communications*, vol. E88-B, pp. 1577–1585, Apr. 2005.

[15] S. W. Kim, B.-S. Kim, and Y. Fang, "Downlink and uplink resource allocation in IEEE 802.11 wireless LANs," *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 320–327, Jan. 2005.

[16] S. Gopal and D. Raychaudhuri, "Experimental evaluation of the TCP simultaneous-send problem in 802.11 wireless local area networks," in *Proceeding of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, pp. 23–28, Aug. 2005.

[17] Y. Wu, Z. Niu, and J. Zheng, "Study of the TCP upstream/downstream unfairness issue with per-flow queuing over infrastructure-mode WLANs," *Wireless Communications and Mobile Computing*, vol. 5, pp. 459–471, June 2005.

[18] M. Bottigliengo, C. Casetti, C. F. Chiasserini, and M. Meo, "Smart traffic scheduling in 802.11 WLANs with access point," in *Proceedings of IEEE 58th Vehicular Technology Conference 2003 (VTC 2003)*, vol. 4, pp. 2227–2231, Oct. 2003.

[19] S. Shin and H. Schulzrinne, “Balancing uplink and downlink delay of VoIP traffic in WLANs using adaptive priority control (APC),” in *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2006)*, vol. 191, Aug. 2006.

[20] H. Kim, H. Lee, S. Shin, and I. Kang, “On the cross-layer impact of TCP ACK thinning on IEEE 802.11 wireless MAC dynamics,” in *Proceedings of IEEE Vehicular Technology Conference (VTC 2006)*, pp. 1–6, Sept. 2006.

[21] J. Freitag, N. L. S. da Fonseca, and J. F. de Rezende, “Tuning of 802.11e network parameters,” *IEEE Communications Letters*, vol. 10, pp. 613–613, Aug. 2006.

[22] N. Blefari-Melazzi, A. Detti, I. Habib, A. Ordine, and S. Salsano, “TCP fairness issues in IEEE 802.11 networks: Problem analysis and solutions based on rate control,” *IEEE Transactions on Wireless Communications*, vol. 6, pp. 1346–1355, Apr. 2007.

[23] K. Cai, M. Blackstock, R. Lotun, M. J. Feeley, C. Krasic, and J. Wang, “Wireless unfairness: Alleviate MAC congestion first!,” in *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2007)*, pp. 43–50, Sept. 2007.

[24] F. Micó, J. M. Villalón, and P. Cuenca, “Unfairness uplink/downlink in IEEE 802.11 WLANs,” in *Proceedings of Networking and Electronic Commerce Research Conference 2008 (NAEC 2008)*, pp. 524–539, Sept. 2008.

[25] F. Keceli, I. Inan, and E. Ayanoglu, “Achieving fair TCP access in the IEEE 802.11 infrastructure basic service set,” in *Proceedings of IEEE International Conference on Communications (ICC 2008)*, pp. 2637–2643, May 2008.

[26] F. Keceli, I. Inan, and E. Ayanoglu, “Weighted fair uplink/downlink access provisioning in IEEE 802.11e WLANs,” in *Proceedings of IEEE International Conference on Communications (ICC 2008)*, pp. 2473–2479, May 2008.

[27] Q. Wu, M. Gong, and C. Williamson, “TCP fairness issues in IEEE 802.11 wireless LANs,” *Computer Communications*, vol. 31, pp. 2150–2161, June 2008.

[28] B. A. H. S. Abeysekera, T. Matsuda, and T. Takine, “Dynamic contention window control mechanism to achieve fairness between uplink and downlink flows in IEEE 802.11 wireless LANs,” *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3517–3525, Sept. 2008.

[29] J. Ha, E.-C. Park, K.-J. Park, and C.-H. Choi, “A cross-layer dual queue approach for improving TCP fairness in infrastructure WLANs,” *Wireless Personal Communications*, vol. 51, pp. 499–516, June 2009.

[30] E.-J. Lee, H.-T. Lim, S.-J. Seok, and C.-H. Kang, “A scheme for enhancing TCP fairness and throughput in IEEE 802.11 WLANs,” in *Proceedings of NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, vol. 4479, pp. 368–379, Nov. 2007.

[31] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, 1989.

[32] R. Braden, “Requirements for internet hosts – communication layers,” *Request for Comments 1122*, Oct. 1989.

[33] Microsoft Corporation, *Microsoft Windows Server 2003 TCP/IP Implementation Details*, June 2003.

[34] P. Sarolahti and A. Kuznetsov, “Congestion control in linux TCP,” in *Proceedings of 2002 USENIX Ammial Tecnical Conference*, pp. 49–62, June 2002.

[35] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks,” in *Proceedings of International Workshop on Protocols for Future, Large-Scale and Diverse Network Transport (PFLDnet 2006)*, Feb. 2006.

[36] S. Ha, I. Rhee, and L. Xu, “CUBIC: A new TCP-friendly high-speed TCP variant,” *ACM SIGOP Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[37] Dell Latitude E5500. available at `http://www.dell.com/downloads/jp/products/latit/latitude_e5400_e5500.pdf`.

[38] Dell Precision 390. available at http://www.dell.com/downloads/jp/products/precn/Precision390_0711.pdf.

[39] Ubuntu. available at http://www.ubuntu.com/.

[40] netem. available at http://www.linuxfoundation.org/en/Net:Netem.

[41] M. Mathis, J. Heffner, and R. Reddy, "Web100: Extended TCP instrumentation for research, education and diagnosis," *ACM Computer Communications Review*, vol. 33, pp. 69–79, July 2003.

[42] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki, "Measuring queue capacities of IEEE 802.11 wireless access points," in *Proceedings of the 4th International Conference on Broadband Communications, Networks, and Systems (BROADNETS 2007)*, pp. 846–853, Sept. 2007.

[43] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf-the TCP/UDP bandwidth measurement tool." available at http://dast.nlanr.net/Projects/Iperf/.

[44] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proceedings of ACM Sigmetrics 2000*, June 2000.

[45] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of IEEE 802.11 and its applications," in *Proceedings of the IEEE International Symposium on Network Computing and Applications (NCA 03)*, pp. 249–256, Apr. 2003.