# A Scheme for Enhancing TCP Fairness and Throughput in IEEE 802.11 WLANs

Eun-Jong Lee[1], Hyung-Taig Lim[1], Seung-Joon Seok[2], and Chul-Hee Kang[1]

[1]Department of Electronics Engineering, Korea University
5-ga, Anam-dong, Sungbuk-gu, Seoul 136-701 Korea
{lej,limht,chkang}@widecomm.korea.ac.kr
[2]Department of Computer Science and Engineering, Kyungnam University
449, wolyong-dong, Masan, 631-701 Korea
sjseok@kyungnam.ac.kr

**Abstract.** In this paper, we consider two fairness problems that occur in the infrastructure network (fairness between TCP uplink/downlink flows and fairness between competing TCP uplink flows). A large number of existing works have studied the TCP fairness issues and greatly solved the TCP unfairness problems. However, these solutions suffer from the drawback of TCP throughput degradation even though the TCP unfairness problem can be solved. In order to solve these problems effectively, we propose a scheme that modifies the receiver window size on the basis of maximum window size which is able to maximize link utilization. We have evaluated our scheme with ns-2 simulator and the results demonstrate that our scheme greatly improves both the TCP fairness and total throughput.

## 1 Introduction

Wireless local area networks (WLAN) based on the IEEE 802.11 standard are rapidly being deployed in hot spots such as hotels, libraries, and airports to provide the nomadic users with open access to the Internet. This WLAN is divided into two networks: infrastructure and Ad-hoc WLAN. The former is a network where all mobile stations only communicate through an access point (AP) in order to access to wired network and the latter is a network where mobile stations communicate directly with each other without access to the wired network.

The IEEE 802.11 standard provides two channel access functions: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). DCF is contention-based channel access method using carrier sense multiple access with collision avoidance (CSMA/CA). The CSMA/CA guarantees equal medium access opportunity to all stations. PCF is a contention free channel access method employing a polling technique by the Point Coordinator (PC) which determines the station that is going to transmit data. In this paper, our target network is infrastructure WLAN under DCF and we assume that the bottleneck between TCP end nodes is only on the wireless link.

Currently, most of WLANs support the DCF with CSMA/CA mechanism. However, in the infrastructure network in the presence of both mobile senders and receivers, this mechanism leads to significant unfairness problems between uplink and downlink flows. In this paper, two TCP unfairness problems that occur in infrastructure WLAN are considered.

Many existing works have studied the TCP fairness issues and greatly solved the TCP unfairness problems. However, these solutions suffer from the drawback of TCP throughput degradation even though the TCP unfairness problem can be solved. In order to solve these problems, we propose a scheme that modifies the advertised receiver window size on the basis of maximum window size which is able to maximize the channel utilization. Our scheme is only deployed on AP and we do not conduct any changes to the mobile station.

The rest of the paper is organized as follows. Section 2 describes two TCP unfairness problems in infrastructure WLANs in detail. Section 3 presents related works. Section 4 investigates the source of the TCP throughput degradation in an existing solution. In Section 5, we describe the proposed scheme. In Section 6, we evaluate the proposed scheme based on simulation results. Finally, in Section 7, we present our conclusions.

## 2   TCP Unfairness in Infrastructure WLANs

In this section, we illustrate two unfairness problems that arise between TCP flows in the infrastructure network.

### 2.1   Unfairness between TCP uplink/downlink flows

The scenario we consider is shown in Figure 1, where mobile stations in the WLAN communicate with a fixed server in a high-speed wired network through an AP. The mobile stations consist of one TCP sender and n TCP receivers. The TCP sender forwards TCP data to the AP through the uplink and the AP forwards TCP data to the mobile receivers through the downlink. All TCP data that will be transmitted to the mobile receivers are queued in the downlink buffer of AP and all TCP data that will transmit from the mobile sender are queued in a buffer of the each mobile station.
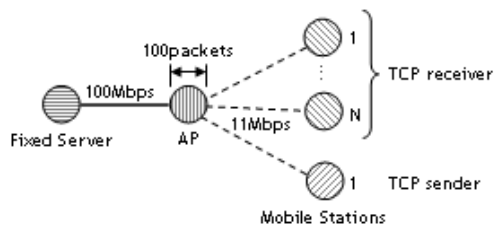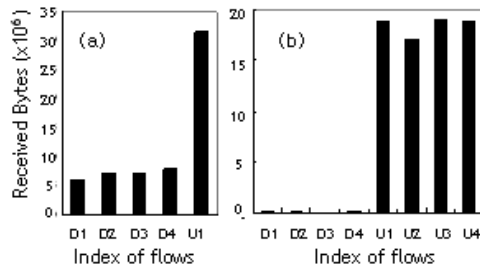


**Fig. 1.** Network topology

**Fig. 2.** TCP throughput for multiple flows;
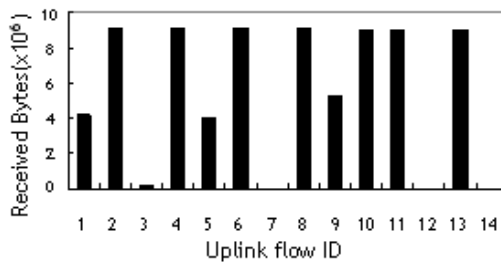(a) 4 downlink flows and 1 uplink flows, (b) 4 downlink flows and 4 uplink flows.



**Fig. 3.** TCP received bytes for 14 uplink flows in IEEE 802.11b WLAN

Since the WLAN under DCF provides equal access opportunity to the media for all mobile stations, the AP and TCP sender equally compete to obtain channel access opportunity. The AP and one mobile sender are provided with equal access opportunities to the media. Hence, the mobile sender gets half of the channel bandwidth and the remaining half is equally shared by all the mobile receivers through the AP. Therefore, the data to be sent from the TCP sender has n times more opportunities than the data to be sent to the TCP receiver. The more the number of mobile sender increases, the more serious becomes the unfairness between uplink and downlink flow. Since the wired network is assumed to be a fast Ethernet at 100Mbps, the downlink access to the IEEE 802.11b at 11Mbps is the bottleneck. Moreover, the AP buffer suffers from heavy traffic load since the TCP data and ACKs share the AP downlink buffer. Consequently, overflow occurs at the AP and TCP senders of downlink flow recognize the packet drop as network congestion and reduce their data sending rate accordingly. However, the uplink flow still arrives at the maximum window size while the downlink flow struggles with the small window size of packets [1]. In order to confirm this problem, a simulation is conducted using the ns-2 simulator. Without loss of wireless channel, multiple mobile stations are set separately as senders and receivers to communicate with a fixed server. The AP queue limit is set to 100 packets [1]. The propagation delay of the wired network is set at 50ms. The

TCP version is Reno and the size of TCP/IP packets is fixed at 1040bytes. The default receiver window size is set to 60 packets and we run the simulation for 190 seconds. The simulation results for received bytes for each receiver of multiple uplink and downlink flow are presented in Figure 2. As shown in Figure 2 (a), it is obvious that whenever there is an uplink flow passing through the AP, the throughput of all downlink TCP flows is decreased greatly. Moreover, as shown in Figure 2 (b), when the number of uplink flows increases, the unfairness between uplink/downlink flows becomes much more serious.

### 2.2   Unfairness between TCP uplink flows

The greedy closed loop control nature of TCP leads to the unfair sharing of the channel bandwidth among each TCP, even when all mobile senders are in the WLAN. Consider the scenario of WLANs where all mobile stations operate as the TCP sender. When TCP senders transmit TCP data, the wireless stations queue TCP data to be sent to the destination and the returning TCP ACK packets are queued at the downlink buffer of the AP to access the wireless channel. For large numbers of stations, the AP downlink buffer is overwhelmed with the TCP ACKs, and then the TCP ACK may be dropped by buffer overflow. The dropping of TCP ACKs can disturb congestion window growth and invoke repeated timeouts. TCP flows with a small number of packets in flight are liable to invoke timeouts since the loss of a small number of data or ACK packets is sufficient to induce a timeout. Hence, TCP ACK losses at the AP buffer can easily occur in a newly started TCP flow [2]. Therefore, these stations cannot escape from this adverse condition.

We confirmed the unfairness between competing uplink flows using ns-2 simulator. Figure 3 presents the received bytes for each TCP receiver. As shown in Figure 3, some stations experience starvation as a result of repeated timeouts.

## 3   Related Works

Many existing works have studied the TCP fairness issues in the cellular network, infrastructure WLANs and ad-hoc WLANs. In this section, we focus on the existing solutions related to infrastructure WLANs. These solutions are largely divided into two methods - a method for increasing the sending rate of the downlink flows and decreasing the sending rate of the uplink flows.

The former method is illustrated in [2]. The authors ensure fairness between competing TCP uploads and downloads by prioritizing to TCP downlink traffic using the 802.11e AIFS, TXOP and $CW_{min}$ parameters. The authors also indicated the unfairness between competing uplink flows and for this problem, proposed the ACK prioritization scheme. However, this solution suffers from the weak point that is possible to adapt only to the network that supports IEEE 802.11e MAC.

The latter method is illustrated in [1], [3] and [4]. The authors in [3] and [4] proposed a scheme that suitably drops the incoming data to AP uplink buffer.
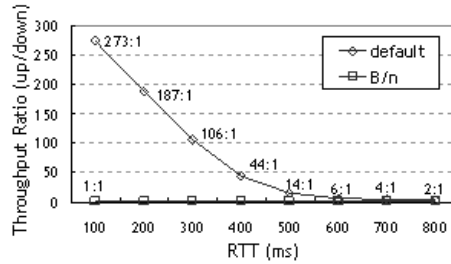
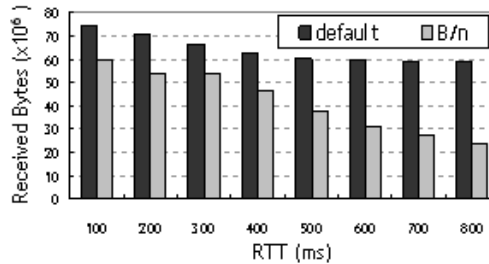**Fig. 4.** TCP throughput ratio (uplink/downlink) for RTT



**Fig. 5.** TCP received bytes for RTT

The solution in [3] is based on rate control mechanisms using the token bucket filter and the scheme proposed in [4] ensures the uplink/downlink fairness using a virtual queue management, named VQ-RED. However, these solutions suffer from the drawback that they waste wireless channel utilization. Finally, the solution proposed in [1] is a TCP receiver window size manipulation scheme. The authors indicated that TCP uplink/downlink unfairness is influenced by AP buffer size and then proposed a scheme that modifies the receiver window size in TCP ACK into min (rwnd, ⌊buffer_size / flow_number⌋) so that any loss of the packet does not occur in the downlink buffer. This solution is the most efficient among existing solutions because the TCP sender can reduce the data sending rate by itself and also dose not waste the wireless channel utilization owing to no loss of data at AP. Moreover, this method can be supported irrespective of IEEE 802.11 MAC type. These explain why our scheme is based on TCP receiver window size manipulation.

## 4   TCP Throughput Limits in an Existing Solution

Existing solutions suffer from the drawback of TCP throughput degradation while the TCP unfairness problem can be solved. We investigate the cause of the TCP throughput degradation based on TCP receiver window size manipulation that is most efficient among existing solutions.

We confirmed TCP throughput degradation by ns-2 simulation. Figure 4 shows the TCP uplink/downlink throughput ratio for Round Trip Time (RTT) in the case of the IEEE 802.11b standard applying the TCP receiver window size manipulation scheme (denoted in the following as "B/n", where B is the AP buffer size and n is the number of TCP flows) and without applying any scheme (denoted in the following as "default"). As shown in Figure 4, the B/n scheme ensures the TCP uplink/downlink fairness irrespective of RTT while the default case suffers from significant unfairness during a small RTT. However, Figure 5 displays that the B/n scheme greatly decreases TCP total throughput in the case of an increase in RTT. The B/n scheme never reaches the maximum window size due to the limitation on the TCP congestion window growth. The TCP sender chooses the minimum value between congestion window and receiver window, this solution restricts the receiver window size to B/n. In other words, the TCP throughput is as follows.

$$TCP\ throughput = \frac{Window\ size}{RTT} \tag{1}$$

Then, we recognize that the B/n scheme limits the window size only to AP buffer size regardless of the increase or decrease in RTT. Thereby, TCP throughput with constant window size is reduced according to the increase in RTT.

In order to solve this problem, we investigate the maximum window size that is able to be accommodated in the link without loss of the channel. The maximum window size considers a single bottleneck link with a capacity of $\mu$ packets per second and a FIFO buffer of size of B packets. For each connection, T denotes the propagation delay plus the service time in the bottleneck, namely RTT. The maximum window size is as follows.

$$W_{max} = B + \mu T \tag{2}$$

In this case, the bottleneck buffer is always full of packets and there are $\mu$T packets in flight [5]. The scheme that modifies receiver window size into min ( rwnd, $\lfloor$B/n$\rfloor$ ) limits the window size only to AP buffer size (B) regardless of the factor titled T. Based on these causes, if the sending rate of all senders is limited by the standard of the maximum window size, not only TCP fairness will be guaranteed, but also link utilization will be maximized.

## 5    Proposed Solution

In this section, we illustrate the proposed solution which improves both TCP fairness and total throughput through a simple correction based on TCP receiver window size manipulation scheme.

### 5.1    $W_{max}$-based TCP receiver window size manipulation scheme

We define proposed solution as $W_{max}$-based TCP receiver window manipulation scheme. This scheme is to use the receiver window size field in the acknowledg-

**Table 1.** IEEE 802.11b Parameter Values

| Parameter | Definition | Value |
|---|---|---|
| $T_{slot}$ | slot time | 20 $\mu$s |
| $\tau$ | Propagation delay | 1 $\mu$s |
| $T_p$ | Transmission time of the physical Preamble | 144 $\mu$s |
| $T_{PHY}$ | Transmission time of the PHY header | 48 $\mu$s |
| $T_{DIFS}$ | DIFS time | 50 $\mu$s |
| $T_{SIFS}$ | SIFS time | 10 $\mu$s |
| $CW_{min}$ | Minimum backoff window size | 31 $\mu$s |
| $CW_{max}$ | Maximum backoff window size | 1024 $\mu$s |
| $L_{MAC\_H}$ | MAC overhead | 68 bytes |
| $L_{MAC\_ACK}$ | ACK size | 38 bytes |
| $L_{RTS}$ | RTS size | 44 bytes |
| $L_{CTS}$ | CTS size | 38 bytes |
| $L_{TCP\_data}$ | TCP data size | 1040 bytes |
| $L_{TCP\_ACK}$ | TCP ACK size | 40 bytes |
| $R_{data}$ | Data rate | 11 Mbps |
| $R_{control}$ | Control rate | 2 Mbps |

ment packet. This field is originally used to match sending rate of TCP sender to processing rate of TCP receiver. Since the TCP sender chooses the minimum value between congestion window and receiver window, we can naturally reduce the sending rate of TCP using the receiver window field. Thus by manipulating the receiver window size at the AP, we can ensure that the TCP sending rate is limited to the value we calculate. A similar approach was used in [1] for TCP uplink/downlink fairness. However, this solution, as shown in the Section 4, suffers from TCP throughput degradation.

Our solution is to modify the receiver window size on the basis of the maximum window size which is able to maximize the link utilization instead of the AP buffer size. We modify the receiver window size in all TCP acknowledgment packets passing through AP into not min ( rwnd, $\lfloor B/n \rfloor$ ), but min ( rwnd, $\lfloor W_{max}/n \rfloor$ ). In our solution, the maximum window size is calculated by AP when TCP connection opens. This value is originally for a single connection, but we have to consider a network where there are multiple TCP connections. However, this is not a problem since our solution calculates the value for a single connection at the AP and then divides the maximum window size among all TCP connections.

### 5.2   Calculating Maximum Window Size

We consider a particular scenario of WLANs where a single mobile station communicates with a fixed server. In this case, we refer to the maximum window size shown in expression (2) that can be accommodated in steady state in the link. The bottleneck link corresponds to the downlink of AP in our scenario.

Therefore, B is equal to AP buffer size and $\mu$ is the same as AP service rate. Therefore, we can newly present the expression (2).

$$W_{max} = AP\ Buffer\ size + AP\ Service\ rate \cdot RTT \qquad (3)$$

In this case, the AP can easily find the AP buffer size and the RTT (we assume that we can measure RTT using ICMP packet, e.g. ping). Then we need to find the AP service rate ($\mu$) In table 1, we define the IEEE 802.11b parameter value that will be used in our simulation and then we calculate the AP service rate based on the defined parameter value. The AP service rate ($\mu$), TCP data transmission time ($T_{TCP\_data}$), and TCP ACK transmission time ($T_{TCP\_ACK}$) are as follows [6].

$$\mu = \frac{1}{T_{tcp\_data} + T_{tcp\_ack}} (packet/seconds) \qquad (4)$$

$$
\begin{aligned}
T_{TCP\_data} = \overline{CW} &+ T_{DIFS} + T_P + T_{PHY} + \frac{L_{RTS}}{R_{control}} \\
&+ T_{SIFS} + T_P + T_{PHY} + \frac{L_{CTS}}{R_{control}} + T_{SIFS} + T_P \\
&+ T_{PHY} + \frac{L_{MAC\_H} + L_{TCP/IP\_H} + L_{TCP\_data}}{R_{data}} \\
&+ T_{SIFS} + T_P + T_{PHY} + \frac{L_{MAC\_ACK}}{R_{control}}
\end{aligned} \qquad (5)
$$

$$
\begin{aligned}
T_{TCP\_ACK} = \overline{CW} &+ T_{DIFS} + T_P + T_{PHY} + \frac{L_{RTS}}{R_{control}} \\
&+ T_{SIFS} + T_P + T_{PHY} + \frac{L_{CTS}}{R_{control}} \\
&+ T_{SIFS} + T_P + T_{PHY} + \frac{L_{MAC\_H} + L_{TCP/IP\_H}}{R_{data}} \\
&+ T_{SIFS} + T_P + T_{PHY} + \frac{L_{MAC\_ACK}}{R_{control}}
\end{aligned} \qquad (6)
$$

In the above, $\overline{CW}$ is the average backoff time [7] given by

$$\overline{CW} = \frac{CW_{min} T_{slot}}{2} \qquad (7)$$

We can derive from expression (5), (6) the summation of TCP data transmission time and TCP ACK transmission time and the value is 3868 $\mu$s. Then, we can derive the AP service rate using expression (4) and the value is approximately 2.15Mbps. Hence, we can again define the maximum window size as follows.

$$W_{max} = AP\ buffer\ size + (2.15Mbps * RTT) \qquad (8)$$

We will use this expression for our scheme.

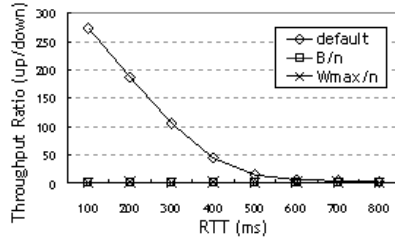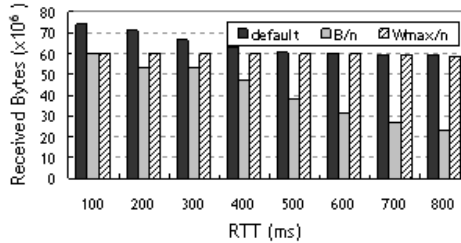**Fig. 6.** TCP throughput ratio for RTT



**Fig. 7.** TCP received bytes for RTT

### 5.3   Measuring TCP Flow Number

In order to implement our solution, we need to measure the number of current TCP flows in the link. This measurement is similar to the method in [1]. Practically, it is very difficult to know the exact number of active flows, especially in the case of flows in an open connection. In addition, determining the direction of data is very hard (uplink flow or downlink flow). However, these are not problems for our solution since we count only the number of active flows regardless of the direction and consider a network where only TCP flows are present. In order to count the number of active TCP flows, we keep monitoring a pair of IP address and port number in the header of TCP packets passing through the AP. If a new flow is observed, the variable storing the number of active flows is increased by 1 and if a particular flow is not observed during a regular period, the variable is decreased by 1.

## 6   Simulation Results

In this section, we evaluate the performance of the proposed scheme by ns-2 simulator. The simulation scenario is depicted in Figure 1 and runs on equal terms with topology used in previous simulation with the exception of the number of TCP sender. This scenario assumes a network where the number of TCP sender is equal to the number of TCP receivers.

In order to evaluate the performance, we have two important performance metrics: TCP fairness and total throughput.
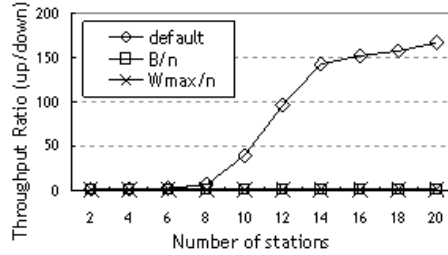
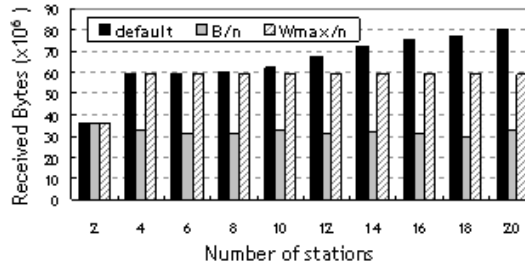**Fig. 8.** TCP throughput ratio for number of stations.



**Fig. 9.** TCP received bytes for number of stations

TCP fairness is the ratio of the TCP uplink throughput is the summation of uplink throughput and downlink throughput. The simulations run with various parameter functions such as RTT, number of stations and error rate. Further, we compare the performance metrics for three reference models: default, B/n and our solution (denoted in following as "$W_{max}/n$")

Figure 6 and 7 show the throughput ratio and total throughput for varying RTT when the mobile stations are organized into four TCP sender and four TCP receivers respectively. As shown in Figure 6 and 7, our solution ensures uplink/downlink fairness similar to that of the B/n scheme and total throughput reaches as much as the default and provides a throughput enhancement of approximately 180% higher than that of B/n scheme.

Figure 8 and 9 present results for the increase of the number of stations. When the number of stations is above 10, the ratio is suddenly increased and TCP receivers in WLAN cannot receive packets because of dropping at AP. However, our scheme ensures fairness as in the B/n scheme as well as shows a throughput enhancement of 2 times than B/n as shown in Figure 9.

Figure 10 and 11 show fairness and total throughput for the error rate of the wireless channel. When error rate is 0.2, both fairness and throughput of three reference models show no great difference since the TCP of uplink flow cannot arrive at the desired window size due to the high error rate. When the error rate is 0.2, our scheme is no different than other schemes, however, with a proper error rate not only the fairness is ensured like in the B/n scheme, but also the
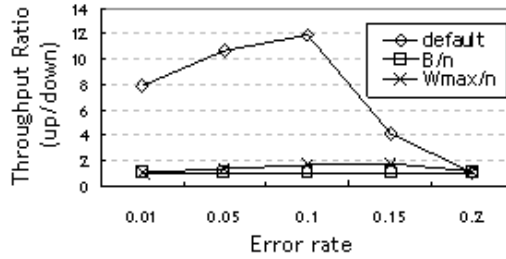
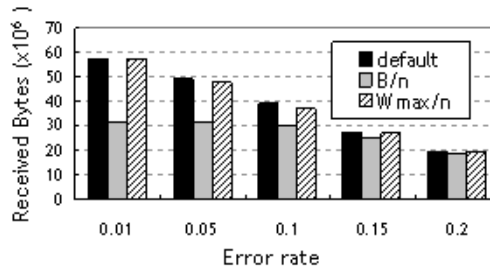**Fig. 10.** TCP throughput ratio for error rate



**Fig. 11.** TCP received bytes for error rate

total throughput always reaches that of the default. Therefore, the unfairness issues in infrastructure WLAN consider only when the channel error rate is 0.2 and below.

Finally, we confirm the fairness and throughput between competing uplink flows. As shown Figure 3 in section 2.2, the unfairness problem in IEEE 802.11b WLAN without applying any scheme is very serious. However, B/n and $W_{max}/n$ schemes fairly ensure the fairness among all mobile stations as shown Figure 12. Further, the received bytes for each TCP receiver in the case of using the $W_{max}/n$ scheme nearly show a throughput enhancement of 2 times than that of the B/n scheme.

## 7   Conclusions

In this paper, we proposed a new scheme that modifies the receiver window size on the basis of the maximum window size which is able to maximize the link utilization. Simulation results showed the fairness and total throughput compared with standard IEEE 802.11b and the case of applying a previous scheme. In the result, we confirmed that our scheme greatly improves not only the fairness between uplink and downlink flows, but also between competing uplink flows like that of the previous scheme, tagged B/n. In addition, we showed that the total throughput of our proposed scheme can reach as much as the default in all cases and approximately 2 times that of the B/n scheme. However, our
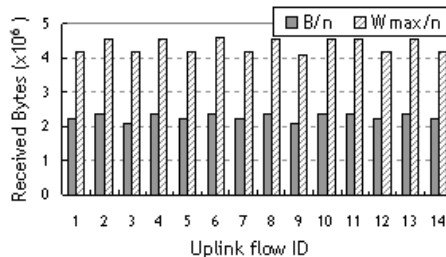
**Fig. 12.** TCP received bytes for each uplink flow ID

scheme suffers from the disadvantage that it has to modify the TCP packet at AP. This packet modification would violate the TCP end-to-end semantics. Therefore, this point must be considered in future research.

# References

1. Pilosof S, Ramjee R, Raz D, Shavitt Y and Sinhan P: Understanding TCP fairness over wireless LAN, In Proceedings of IEEE INFOCOM 2003, Volume 2, (April 2003) pp.863-872.
2. D.J. Leith, P. Clifford, D.W. Malone and A Ng: TCP fairness in 802.11e WLANs, IEEE Communication Letters, Volume 9, Issue 11, (Nov. 2005) pp.964- 966.
3. Blefari-Melazzi, N., Detti, A., Ordine, A. and Salsano, S.: Controlling TCP Fairness in WLAN access networks using a Rate Limiter approach, 2nd International Symposium on Wireless Communication Systems, (Sept. 2005) pp.375-379.
4. Xiaoyang Lin, Xiaolin CKaveg hang and Muppala,J.K: VQ-RED: An Efficient Virtual Queue Management Approach to Improve Fairness in Infrastructure WLAN, The IEEE Conference on Local Computer Networks, (Nov. 2005) pp. 632-638.
5. T.V. Lakshman and U. Madhow: The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Transaction on Networking, volume 5, Issue 3, (June 1997) pp.336-350.
6. Daniele Miorandi, Arzad A. Kherani and Eitan Altman: A Queueing Model for HTTP Traffic over IEEE 802.11 WLANs, Computer Networks, Volume 50, Issue 1, (Jan 2006) pp. 63-79.
7. Y. Xiao and J. Rosdahl: Throughput and delay limits of IEEE 802.11, IEEE Communications Letters, Volume 6, Number 8, (Aug. 2002) pp.355-357.
8. A. Detti, E. Graziosi, V. Minichiello, S. Salsano and V. Sangregorio: TCP Fairness Issues in IEEE 802.11 Based Access Networks, submitted paper.