

RESEARCH

Open Access

Cross-layer based adaptive wireless traffic control for per-flow and per-station fairness

Vasaka Visoottiviset^{1*}, Akkasit Trunganont¹ and Siwaruk Siwamogsatham²

Abstract

In the IEEE 802.11 wireless LANs, the bandwidth is not fairly shared among stations due to the distributed coordination function (DCF) mechanism in the IEEE 802.11 MAC protocol. It introduces the per-flow and per-station unfairness problems between uplink and downlink flows, as the uplink flows usually dominate the downlink flows. In addition, some users may use greedy applications such as video streaming, which may prevent other applications from connecting to the Internet. In this article, we propose an adaptive cross-layer bandwidth allocation mechanism to provide per-station and per-flow fairness. To verify the effectiveness and scalability, our scheme is implemented on a wireless access router and numerous experiments in a typical wireless environment with both TCP and UDP traffic are conducted to evaluate performance of the proposed scheme.

Keywords: wireless local area network, per-station fairness, traffic control, bandwidth allocation

1. Introduction

Nowadays, the IEEE 802.11 [1] wireless local area networks (WLANs) have become the common network infrastructure for most organizations. In typical wireless environments, the bandwidth will be shared among wireless devices, while wireless access points or routers act as hubs connecting these devices together. In general, the bandwidth may not be equally shared among users in the same WLAN, because some users may utilize greedy applications that consume much larger bandwidth and consequently prevent other applications from connecting to the Internet. These applications include, for example, video streaming applications, download accelerators that create many sessions for each download, and the P2P applications such as BitTorrent. Moreover, traffic from wireless stations mounting denial-of-service (DoS) attacks or infected by a virus may overwhelm the network. This problem is particularly crucial for the wireless network environments since the bandwidth is very scarce. To alleviate this problem, per-station fairness shall be guaranteed. By fairness, we mean each wireless client should be able to evenly obtain the maximum bandwidth.

Other than greedy applications, virus-infected hosts, and DoS/DDoS traffic, the distributed coordination function (DCF) mechanism [1], which is the mandatory media access control method in the 802.11 MAC protocol, also contributes to the unfair access problem among wireless stations. Essentially, the DCF mechanism, which relies on the carrier sense multiple access with collision avoidance (CSMA/CA) algorithm, is designed to grant equal transmission opportunities to all devices in the network including the access point (AP) and its clients. That is, each device including the AP itself on average essentially obtains about $1/N$ of the available bandwidth, when there are totally N active wireless devices in the network. Considering the bi-directional transmission scenario in which some wired stations communicate with $N-1$ wireless clients, it may be noted that $N-1$ downlink flows from the wired stations via the AP obtain only $1/N$ of the available bandwidth, while $N-1$ uplink flows from $N-1$ wireless clients totally obtain $(N-1)/N$ of the bandwidth. The problem is that the uplink flows obtain much larger bandwidth than the downlink flows, especially when there are a large number of wireless clients in the network. Essentially, this results in a per-flow unfairness problem in the wireless network. Note that, by flow, we mean the sequence of packets from one particular source to a particular station, which can identify by a 5-tuple of source address, destination address, source port, destination port,

* Correspondence: itvws@mahidol.ac.th

¹Faculty of Information and Communication Technology, Mahidol University, 999 Phuttamonthon 4 Rd, Salaya, Nakhon Pathom 73170, Thailand
Full list of author information is available at the end of the article

and the transport protocol type. With this imbalance communication problem, it is apparent that greedy applications such as multimedia streaming applications can essentially starve the wireless network. In fact, many previous research studies have focused on fair bandwidth allocation in wireless networks [2-9]. However, most of the existing schemes only focused on TCP fairness and could not maintain fair performance when UDP traffic exists.

The main objective of this research is to design a bandwidth allocation mechanism to achieve throughput-based per-flow fairness between uplink and downlink flows for each wireless client as well as throughput-based per-station fairness when both UDP and TCP traffic may concurrently exist as in a typical real network environment in order to restrain and/or control traffic from greedy applications, virus-infected wireless clients, DoS/DDoS wireless attackers. Our mechanism only requires implementation at the AP side and no modification is required at the clients. Thus, it can support all legacy wireless clients.

There are many challenges to achieve our goal. The first challenge is to achieve per-station fairness among wireless clients. Therefore, the access point must equally allocate bandwidth to each wireless client. The second challenge is to achieve per-flow fairness between the uplink and downlink flows within a wireless station. The third challenge is to adaptively adjust the allocation of the uplink and downlink bandwidth for each wireless station, in order to allow a given communication direction to obtain larger bandwidth than the other. Finally, the fourth challenge is to adaptively re-allocate the remaining bandwidth to bandwidth-hungry wireless stations in order to increase link utilization.

In this work, we propose a scheme that utilizes the hierarchical token bucket (HTB) queuing discipline [10] at the access point, and dynamically modifies the value of contention window (CW) of the access point. HTB is one of the queuing disciplines supported by the traffic control (tc) command [11], which is a well-known traffic control system on Linux. By using HTB, we can achieve per-station fairness. Combining with adaptively modifying the CW_{min} (the minimum of contention window) parameter of the AP, we can then achieve fairness between uplink and downlink flows.

The initial idea of this work can be found in [12] wherein only fairness of downlink traffic was originally considered, and the idea was implemented on the commercial Linksys wireless access router with OpenWRT [13] firmware. The key idea behind this initial work is to perform traffic shaping by using the token bucket (TBF) queuing discipline [11], which will be described later in Section 3. Our additional work [14] introduces a wireless LAN adaptive traffic control (WLAN-ATC) scheme, which is enhanced from the original algorithm and is

implemented on a PC-based wireless access router to address the fairness issue between uplink and downlink. That is, our mechanism can be implemented on a residential wireless router as well as a PC-based wireless router.

This article presents a more sophisticated method to prevent domination of uplink flows that affects per-station fairness, and covers additional implementation techniques that can handle a large network with many wireless clients. The evaluation results reveal that our solution is simple, yet effective. The main contributions of this article are: first, we can provide fair bandwidth allocation for each wireless client without any modification on the client devices. Only the wireless access point is required to upgrade its firmware. Second, our fair bandwidth allocation mechanism can alleviate the problems of greedy applications, unwanted traffic from virus-infected hosts and DoS/DDoS wireless attackers. Third, the bandwidth can be re-allocated dynamically based on the number of wireless clients. That is, we can dynamically adjust the bandwidth for each wireless client when a new client is associated with the access point or when a client is disassociated from the access point. Fourth, the access point can adaptively allocate the remaining or extra bandwidth to the bandwidth-hungry wireless stations in order to increase link utilization. In addition, to gain benefits from our proposed scheme, only the wireless access router is needed an upgrade. Both legacy and illegitimated wireless clients have no need to upgrade wireless device drivers, nor install our program. Finally, contrary to most related work, we evaluate our proposed scheme by implementing it on a real Linux-based system and conduct experiments on real IEEE 802.11 g wireless testbeds. Experiment results reveal that our scheme can be effectively deployed in the real-world environments. Note that, since we would like to create a prototype and study the feasibility of implementation in the real wireless network environments, we therefore focus on the implementation other than the simulations.

The rest of article is organized as follows. Section 2 surveys the related work about the performance fairness issue in wireless networks, and Section 3 provides some background on the IEEE 802.11 protocol. The proposed adaptive wireless bandwidth allocation scheme for per-station and per-flow fairness is presented in Section 4, and Section 5 describes the implementation and experiment configurations. Section 6 presents the evaluation methods and the results. Section 7 discusses the pros and cons of our proposed work, followed by the conclusion and future work in Section 8.

2. Literature review

Over the past many years, much research has focused on different aspects of bandwidth allocation and traffic

control in wireless networks. For example, Lin and Lai proposed adaptive QoS management schemes in the infrastructure WLAN [15] for IEEE802.11b [16] devices. They argued that the IEEE 802.11e [17] EDCF channel access mechanism could not adjust an arbitration inter-frame space (AIFS) for each access category in order to adapt to the environment variations. Therefore, to support different QoS requirements, they proposed two schemes consisting of wireless differentiation (WD) scheme for UDP flows and wireless differentiation with prioritized ACK scheme (WDPA) for TCP flows. The technique behind both schemes is to assign a priority level of each packet in the MAC header, which will affect the contention window of the medium access mechanism. A higher priority level implies lesser time to wait on average before each transmission. Hiraguri et al. [18] proposed two wireless traffic control schemes for QoS management and load balancing among multiple access points. This work was implemented on a separate device called wireless traffic control (WTC), which functions as a gateway for access points. To obtain a guaranteed level of QoS, WTC classifies packets into two different priority buffers based on information in the packet header. Packets in a guaranteed flow buffer have ability to be transmitted in a certain rate while packets in a best effort flow will be transmitted only when there is sufficient remaining bandwidth.

Apart from traffic control or QoS in wireless networks, many previous studies focused on fair bandwidth allocation [2-9]. However, most of them only focused on TCP fairness and could not maintain fair performance when UDP traffic exists [2-8]. Pilosoft et al. [2] focused on the study of TCP fairness in IEEE 802.11 networks in the presence of both mobile senders and receivers. They found that the AP buffer size played a critical role on the unfairness problem between uplink and downlink flows, via simulations. They proposed a simple solution to this problem by adjusting the TCP receiver window size advertised from the AP according to the number of flows. Given that the buffer size of the AP equals to B , and there are n flows in the system, the minimum size of the advertised receiver window in the ACK packets of all TCP flows for this technique is $\lfloor B/n \rfloor$.

Seyedzadegan et al. [3] reviewed research work about TCP fairness in wireless LAN covering both per-flow and per-station fairness. They mentioned that the per-flow unfairness problem was caused by the DCF mechanism that allowed uplink TCP flows to dominate downlink TCP flows. There are many techniques to achieve per-flow fairness, e.g. reducing the contention window of AP, expanding the buffer size of AP, using the dual queues at AP and dropping the uplink flow when it consumes more than one half of the total bandwidth. On the other hand, the per-station unfairness problem may be solved by

giving fair channel access durations to each wireless station. Seyedzadegan et al. also proposed a weighted window scheme by extending the work of [2] to provide fair TCP bandwidth allocation among wireless stations [4]. To manipulate the TCP window size of each station, the technique they proposed was to also consider the number of active wireless stations in addition to the number of flows and the buffer size of the access point. Therefore, the minimum size of the advertised receiver window of all TCP flows is $\lfloor (B/m)/n \rfloor$, where m denotes the number of active wireless clients. Their extended recent work [5] also presented a class-based weighted window method in order to support different levels of required bandwidth in each class.

The distributed access time control (DATC) scheme [6] was proposed to provide both per-flow and per-station fairness of TCP flows. This scheme was based on channel access time. When the sending rate of a wireless client during a sample period exceeds a fair rate, the wireless client would decrease its flow rate by adjusting the dropping probability. The channel access time of each station is fairly allocated by dividing the sample period by the number of active stations. However, this scheme cannot cope with UDP flows and implementation requires the computation on all wireless stations.

Park et al. [7,8] proposed a cross-layer feedback approach to assure per-station fairness in TCP over WLANs. They introduce the notion of channel access cost to quantify the traffic load and per-station channel usage. This channel access cost is estimated at the MAC layer in the AP by 'access cost estimator' (ACE). The high access cost is informed to the TCP sender by setting a bit in the packet's MAC header. After the TCP sender is informed about the high access cost, it adjusts its transmission rate based on this cost by utilizing the explicit congestion notification (ECN) mechanism [19].

Blefari-Melazzi et al. [9] proposed a rate-limiter mechanism to provide per-flow fairness between uplink and downlink traffic in wireless networks. The rate-limiter scheme controlled uplink traffic at a specific rate, raising the downlink bandwidth to achieve fairness. The rate of uplink traffic can be increased in order to reduce bandwidth waste, when downlink traffic was not greedy. The adaptation of bandwidth allocation was based on the number of lost packets at the downlink buffer. If there was no packet loss, it implied that no congestion occurred at the downlink buffer, thus the rate of uplink should be increased. Otherwise, the rate of uplink will be decreased to avoid domination of uplink traffic. The results from both simulations and real testbeds [20] confirmed that their proposed scheme could provide per-flow fairness and the controlled rate can be adapted to reduce bandwidth waste. Moreover, the rate-limiter was based on the IP-layer token bucket filter technique.

The recent work of Detti et al. [21,22] proposed the virtual shared bottleneck (VSB) scheme in order to grant per-station throughput fairness for TCP traffic. They defined the fairness level (i, j) as the ratio between the useful data-rate of the i th and j th stations. They also developed the wireless capacity estimator (WCE) based on the PING tool by using a BASH script. An excellent analytical model to evaluate fairness was presented and compared with the experiment measurements. This work was quite similar to ours in the manner that they used HTB scheduling mechanism and evaluate the system performance by means of experiment testbeds. Even though their major goal was to assure per-station throughput fairness with TCP traffic, it was pointed out also that their approach can handle UDP traffic by extending the structure of VSB as briefly discussed in Appendix IV of [22]. In addition to their primitive goal, our objectives are to achieve per-station fairness when both TCP and UDP traffic concurrently exist, and our scheme also attempts to re-assign the remaining bandwidth to greedy/bandwidth-hungry stations in order to increase link utilization.

“Dynamic Contention Window Control to Achieve Fairness between Uplink and Downlink Flows in IEEE 802.11 WLANs” [23,24] was proposed to provide per-flow fairness between uplink and downlink traffic by dynamically adjusting the CWmin parameter at the access point. By the nature of the 802.11 MAC control, uplink flows dominate downlink flows due to the DCF mechanism that grants equal chance of transmission for all wireless devices including the access point and its client. The idea of this research was to minimize the backoff time of the access point to increase the opportunities of downlink transmissions by adjusting the minimum contention window size (CWmin). The authors performed a mathematical analysis and found that in order to provide per-flow fairness, the optimal value of CWmin was a function of the number of downlink flows. When the number of downlink flows increases, CWmin of access point has to be decreased in order to grant more chances for downlink flows. Since the CW parameter is only adjusted at the access point, no modification is required in the MAC protocol of wireless clients.

Table 1 summarizes the related work and compares with our work, WLAN-ATC. We found that many related schemes are evaluated via simulations, and most of them are implemented in the data link layer, and thus require wireless device driver’s modifications. In contrast, our mechanism does not require any modifications at the wireless clients, thus it is very deployment-friendly. Even though WTC [16] and VSB [21,22] are similar to our work, their main objectives are to support fairness among TCP traffic, and not to support adaptive

bandwidth allocation for greedy or bandwidth-hungry applications.

3. Background

3.1 IEEE 802.11 wireless LAN

The IEEE 802.11 standard [1] is a set of protocols for the Physical, MAC and LLC layers for wireless LAN communications. The mandatory functionality of IEEE 802.11, namely DCF, largely relies on the CSMA/CA mechanism to share medium access among wireless stations. CSMA/CA employs the request-to-send (RTS) and clear-to-send (CTS) mechanism. When a station is allowed to transmit, it will broadcast a RTS packet to all stations. The RTS packet will tell the time duration that the media is accessed, thus each station can know how long it has to wait until the channel will become idle. Then the receiving station replies with a CTS packet that also contains the information about how long the channel will be used. Thus, the hidden node problem can be solved. After receiving a RTS packet, a station begins to transmit an actual packet immediately. For each packet transmission, a sender has to wait for an acknowledgment (ACK) from the receiver. If it does not receive the ACK packet during a timeout period, it will assume that collision occurs and retransmit after waiting for a backoff time as in (1)

$$\text{Backoff Time} = \text{Random}() \times \text{aSlotTime} \quad (1)$$

where $\text{Random}()$ denotes a random integer value and aSlotTime denotes the value of one slot duration specified in IEEE 802.11. The random value is randomly chosen from a range of integers between 0 and $\text{CW}-1$, where CW denotes Contention Window, which corresponds to a given integer in the range of CWmin and CWmax . The CW parameter takes the value of CWmin as an initial value and is typically doubled every time when a transmission does not succeed until this value reaches CWmax .

In the legacy IEEE 802.11 standard [25], there is only one backoff instance in a wireless station for each transmission attempt. However, in the IEEE 802.11e [17], there are multiple backoff instances in a wireless station which each backoff will be parameterized with specific traffic category parameters to achieve the QoS differentiate. These parameters consist of AIFS, CW, persistence factor (PF), and transmission opportunity (TXOP). From this approach, packets in a higher priority class have opportunities to be transmitted more frequently than those in a lower priority class.

3.2 Traffic control (tc)

To perform traffic control on a wireless access router, in this research we use the *tc* tool [11]. This tool provides an interface to the kernel that performs traffic shaping,

Table 1 Comparison of related work and our WLAN-ATC scheme

Scheme	Layer of deployment	Minimum required deployment	Goal	Techniques	Performance evaluation technique
Understanding TCP fairness over WLAN [2]	Transport	A wireless access point	Fairness among TCP flows	Manipulating the TCP advertised receiver window size on the AP according to the number of flows	Simulations, few experiments
Weighted window [4,5]	Transport	All wireless clients	Fairness among clients for TCP only	Manipulating TCP window size according to the number of stations and flows	Simulations
WTC [18]	Network	Additional machine	Quality of service for both TCP and UDP flows	Assign two different priority buffer queues	Experiment Testbed
VSF [21,22]	Network	Additional machine	Per-station throughput fairness for TCP traffic	Exploit the QDISC and FILTER traffic control tools of the iproute2 package on a Linux machine located between the gateway and AP	Experiment Testbed
Rate limiter [9,20]	Network	Additional machine	Per-flow fairness between uplink and downlink	Limit rate of uplink flow and detects packet loss at downlink buffer queue	Experiment Testbed
WLAN-ATC	Network and data link	A wireless access point	Per-station Fairness, Uplink and Downlink Fairness for both TCP and UDP. Also adaptively allocate remaining bandwidth to greedy clients	Dynamically limit rate of each station, rate of uplink and downlink flow within a station, and modify the CWmin of AP	Experiment Testbed
Access cost estimator (ACE) [7,8]	Network and data link	A wireless access point	Per-station fairness for TCP and ensure high channel utilization	Calculate channel access cost, and set a single it in the MAC header to notify high access and utilize ECN mechanism for reducing TCP traffic	Simulations
WD and WDPA [15]	Data link	All wireless stations including the access router	Adaptive bandwidth sharing for QoS for both TCP and UDP flows	Assign the priority of traffic in MAC layer	Simulations
DATC [6]	Data link	All wireless clients	Both per-flow and per-station fairness of TCP flows	Fairly allocate channel access time to each station	Simulations
Dynamic CW control [23,24]	Data link	A wireless access point	Per-flow fairness for both UDP and TCP	Optimize CWmin of AP	Simulations

scheduling, policing, and classifying. There are three main components of the ‘*tc*’ command, i.e., queuing disciplines (*qdisc*), *class* and *filter*.

The *qdisc* scheduler is the main component of the *tc* command which is simply a scheduler. Every output interface needs a scheduler to arrange the packets into a queue. Root *qdisc* is the primary egress queuing discipline on any device. The *qdisc* scheduler can be classified into two groups as classless *qdisc* and classful *qdisc*. The classless *qdisc* cannot classify traffic, so the transmitted data will be governed by the same policy. Typical examples of classless *qdisc* are first-in first-out (FIFO), stochastic fair queuing (SFQ), generic random early drop (GRED), and token bucket filter (TBF). On the other hand, classful *qdisc* can classify traffic to the predefined classes. Different methods are utilized to treat data in the queue for different traffic classes. The classful *qdisc* can organize the classes in the hierarchical structure, where a class can be a child of another class. Examples of classful *qdisc* are HTB, priority scheduler (PRIO), and class based queuing (CBQ).

The classful *qdisc* involves two components of *tc* consisting of filters and classes. The filter component is

attached to *qdisc* and acts as a classifier to classify packets for each predefined class based on information in the header, such as the source and destination IP addresses. Classes are the traffic categories where each class contains *ceil* and *rate*. The *rate* parameter is used to set the minimum desired speed which limits transmitted traffic, while *ceil* is used to set the maximum desired speed.

Token bucket filter (TBF) [26,27] is one of a classless *qdisc*, which can shape the transmitted traffic at a specific rate by using the token and bucket. The data can be transmitted if and only if there is an available token in the bucket. Thus, limiting the amount of tokens can also limit the rate of transmitted data. Moreover, the amount of tokens cannot exceed the bucket size. Thus, the size of bucket can limit the rate of burst traffic. Basically, the TBF scheduler can be used to shape traffic, but it cannot flexibly adjust the token rate.

Traffic shaping in TBF implies that when the traffic exceeds its demand rate, excess traffic will not be dropped but instead will be stored in a buffer and transmitted later. This kind of traffic control might increase the heavy load to the wireless router for storing and forwarding the packets into queues. To reduce loads on the access router, we

consider utilizing both traffic shaping and traffic policing mechanisms. For traffic policing, the excess traffic will be immediately dropped, thus the wireless router no longer needs to use any memory resource for buffering excessive packets.

In this research, we employ the HTB *qdisc* [10,28], which performs traffic policing. It is proposed to support the borrowing mechanism by extending features of TBF. HTB employs a number of token buckets arranged in a hierarchy. In the borrowing concept, a parent class can lend its own tokens, which is the bandwidth, to its child classes, when it has some remaining bandwidth. Therefore, the bandwidth utilization can be improved.

As the classful *qdisc*, the root *qdisc* contain one HTB class with two parameters: *rate* and *ceil*. A *ceil* will represent the absolute available bandwidth on a link. In HTB, *rate* means the guaranteed bandwidth available for a given class and *ceil*. A number of children classes can be created under the top-level class. Any bandwidth used between *rate* and *ceil* in each child class is borrowed from a parent class. In order to reserve a particular amount of bandwidth to a specific class, *ceil* and *rate* parameter values of each child class should not be the same as those of the parent class.

In our work, the main *qdisc* is HTB, while *ceil* and *rate* of top parent class is set to the assumed wireless capacity. Children classes will borrow bandwidth, i.e., tokens, from their parents, when they have exceeding rate. A child class may continue to borrow until it reaches *ceil*. When extra bandwidth becomes available, HTB can calculate the ratio of distribution of available bandwidth to the ratios of the classes themselves. By dynamically adjusting the value of *rate* for each child class, a greedy station can ‘borrow’ bandwidth from another station.

Example of a *tc* script using HTB *qdisc* is shown in Figure 1 below.

```
tc qdisc add dev eth1 root handle 1:0 htb
tc class add dev eth1 parent 1:0 classid 1:10
  htb rate 20mbit
tc class add dev eth1 parent 1:0 classid 1:11
  htb rate 5mbit ceil 20mbit
tc filter add dev eth1 parent 1:0 protocol ip
  u32 match ip dst 192.168.1.11/32 flowid 1:11
tc class add dev eth1 parent 1:0 classid 1:12
  htb rate 15mbit ceil 20mbit
tc filter add dev eth1 parent 1:0 protocol ip
  u32 match ip dst 192.168.1.12/32 flowid 1:12
```

Figure 1 Example of a *tc* script with HTB *qdisc*.

In this example, the parent class 1:0 has a rate of 20 Mbps. It contains two children classes with id 1:11 and 1:12, having rate of 5 and 15 Mbps, respectively. The *ceil* parameter of both children classes is configured as 20 Mbps, which is the absolute available bandwidth of the parent class. Figure 2 depicts the hierarchical view of HTB *qdisc* structure in our example.

3.3 Fairness and fairness index

As mentioned in [7], the unfairness problem in WLANs results from TCP-induced asymmetry and MAC-induced asymmetry. Fairness can be categorized in many aspects. First, fairness can be per-flow fairness, per-station fairness, or uplink-downlink fairness. Another aspect is time-based fairness and throughput-based fairness as mentioned in [29]. The objective of time-based fairness is to solve the ‘performance anomaly’ of IEEE 802.11 [30], as the throughput of wireless stations with high data rates will be restricted within the lowest rate used by a station. On the other hand, the goal of throughput-based fairness is normally to equally allocate bandwidth to each wireless client for per-station fairness.

In this article, we focus on the throughput-based per-station fairness. To evaluate the throughput-based per-station fairness, we use Raj Jain’s Fairness index [31]. Equation 2 presents the calculation of

$$\text{FairnessIndex} = \frac{(\sum x_i)^2}{n \sum x_i^2} \quad (2)$$

while x_i is the actual throughput of each wireless station and n is the number of active wireless station.

4. Proposed work

In this section, we describe our proposed scheme named ‘a Wireless LAN Adaptive Traffic Control (WLAN-ATC)’ for solving the unfair bandwidth allocation problem of the wireless uplink and downlink traffic, and also per-station fairness. This mechanism works on the IP layer and also utilizes the CW parameter in the MAC

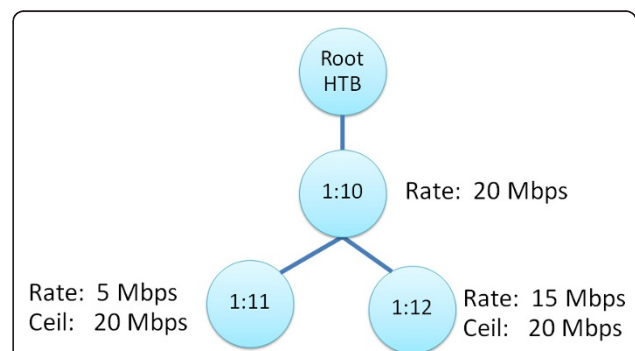


Figure 2 Hierarchical view of HTB *qdisc*.

layer. To solve the unfair bandwidth allocation, in the proposed scheme, the wireless access router must be able to analyze the current wireless traffic, so that it can know the number of wireless stations in the wireless network, how much bandwidth each station consumes and the greedy status of each station. In addition, the wireless access router must be able to control the traffic by policing so that it can prevent the greedy station from consuming much more bandwidth than other stations. Traffic policing is based on the information it analyzes. The wireless access router must also be able to configure the CW parameter in the MAC layer in order to prevent domination of uplink flows.

The system overview is illustrated in Figure 3. The wireless access router connects wireless clients to the Internet via the wired network. Each direction of traffic, upstream or downstream traffic, will be analyzed by the wireless router to calculate its throughput or bandwidth, and will be controlled in the fair manner by the wireless access router, so that each wireless station can equally consume bandwidth.

As illustrated in Figure 4, there are three main processes in our system: packet sniffing, traffic analysis, and adaptive bandwidth allocation. The packet-sniffing module aims to monitor data packets transmitted through the wireless interface of wireless access router. Essentially, it accesses into the IP header of each packet to find information such as the source/destination IP

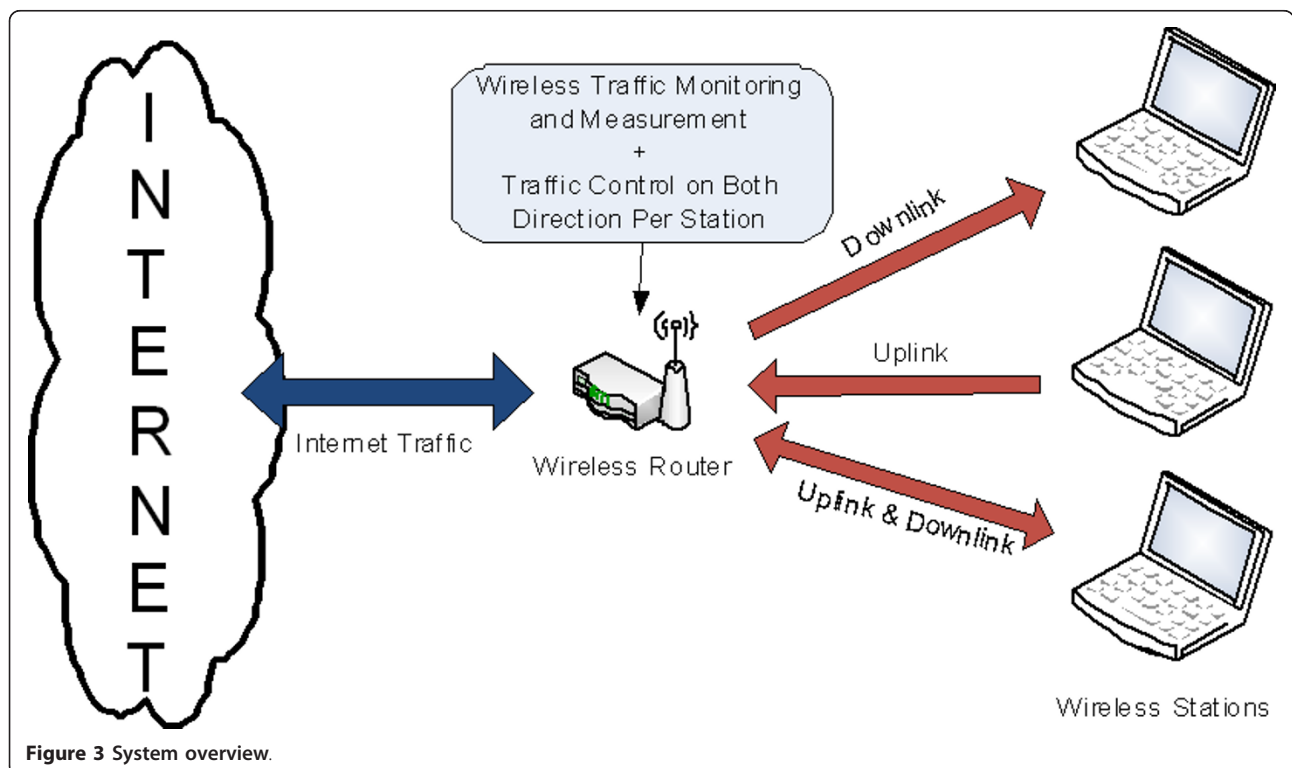
addresses, the source/destination ports, protocols, and the packet length.

4.1 Traffic control module

The traffic analysis module is implemented to acquire information about the current wireless traffic such as the number of wireless stations, the IP address of each station, how much bandwidth each station consumes, the greedy status of each station and the fair rate for each station.

A bandwidth consumption rate for each station is calculated by looking up the source/destination IP address and packet length information in the packets sniffed by the packet-sniffing module. If the source IP address of a packet matches to the IP address of the wireless client, it will be calculated as uplink consumption. Similarly, if the destination IP address of the packet matches to the IP address of the wireless station, it will be calculated as downlink consumption.

Next, the greedy status of each station can be determined by detecting the dropped or backlogged packets on both uplink and downlink queues in each station. Dropped and backlogged packets can be detected by periodically observing the statistics of each packet queue from the *tc* command on the access router. If there is a dropped or backlogged packet on any queue, it implies that the queue does not satisfy its desired rate and indicates that it is greedy.



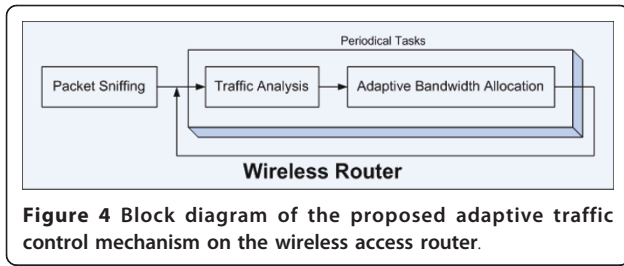


Figure 4 Block diagram of the proposed adaptive traffic control mechanism on the wireless access router.

Here, we define three levels for the greedy status, i.e., *intra-greedy*, *inter-greedy* and *non-greedy*. The *intra-greedy* status corresponds to the case when only the uplink flows or the downlink flows of a station are indicated as greedy. If both uplink and downlink flows of a station are indicated as greedy, the status of this station is deemed as *inter-greedy*. The status of a station is *non-greedy* if and only if neither of its uplink and downlink flows are indicated as greedy.

Finally, the *fair rate* can be calculated from the wireless capacity divided by the number of wireless stations. In short, this traffic analysis module will generate information about current wireless traffic in a bandwidth consumption table, which records the IP address, the bandwidth consumption rate in the unit of bps and the greedy status of each station.

It is worth mentioning that the traffic analysis and adaptive bandwidth allocation must be periodically performed in order to provide a semi real-time traffic control, while the packet-sniffing module must be executed at all time.

4.2 Adaptive bandwidth allocation module

There are four main ideas in our adaptive bandwidth allocation. First, the bandwidth must be equally/fairly allocated among stations. Thus, the given rate for each station is the maximum achievable capacity of the wireless network divided by the number of wireless stations. We call this rate as the *fair rate*. After each station obtains a fair rate, it will equally allocate this bandwidth to its uplink and downlink traffic. Therefore, if there are N wireless clients in the wireless network with the maximum achievable capacity of B bps, then each client is initially allocated $B/(N*2)$ bps, i.e., half of the fair rate, for each uplink and downlink communication. Note that this amount of bandwidth is for each direction per station. Although a station has multiple flows, its overall bandwidth is still the same as previously stated.

Second, the bandwidth can be adjusted between uplink and downlink communications within a specific station, when the station requires more bandwidth in either uplink or downlink direction. Uplink flows can borrow bandwidth from the downlink channel within the same station and vice versa.

Third, when a greedy station requires much higher total bandwidth of uplink and downlink flows than its portion (fair rate); while other stations do not fully consume their portions of bandwidth, the greedy stations can borrow bandwidth from non-greedy stations.

Finally, since TCP traffic transfers require ACK packets to be sent back, the TCP throughput will be severely degraded when either the bandwidth of the uplink or downlink flows is too small. Therefore, each station must maintain a *minimum guaranteed bandwidth* for both uplink and downlink direction in order to allow a flow to grow in the future. In addition, if the queue holds the bandwidth consumption rate as much as the minimum guaranteed bandwidth, it also implies that this queue is likely to be greedy.

Algorithm 1: Adaptive Bandwidth Allocation Algorithm

```

FOR ALL NON-GREEDY STATION
  IF minGuaranteeRate >  $\hat{C}.upConsume_i$ 
    remainingUpBWi =  $\hat{C}.upRate_i - minGuaranteeRate$ 
  ELSE
    remainingUpBWi = MAX( $\hat{C}.upRate_i - \hat{C}.upConsume_i - minGuaranteeRate$ , 0)
  ENDIF
  IF minGuaranteeRate >  $\hat{C}.dnConsume_i$ 
    remainingDnBWi =  $\hat{C}.dnRate_i - minGuaranteeRate$ 
  ELSE
    remainingDnBWi = MAX( $\hat{C}.dnRate_i - \hat{C}.dnConsume_i - minGuaranteeRate$ , 0)
  ENDIF
ENDIFOR
totalRemainingBW =  $\sum remainingUpBW_i + \sum remainingDnBW_i$ 
borrowRate = totalRemainingBW/stationNum
totalBorrow = borrowRate * greedyNum
FOR ALL Station C {
  CASE  $\hat{C}.greedyStatus$  OF
  INTRA-GREEDY:
    holdRatei =  $\hat{C}.upRate_i + \hat{C}.dnRate_i$ 
    IF Uplink is greedy
      C.dnRatei = MAX( $\hat{C}.dnRate_i - (holdRate_i * STEP\_RATIO)$ , minGuaranteeRate)
      C.upRatei = MIN( $\hat{C}.upRate_i + (holdRate_i * STEP\_RATIO)$ , holdRatei - minGuaranteeRate)
    
```


ELSE IF Downlink is greedy

$C.upRate_i = \text{MAX}(\hat{C}.upRate_{i-} - (\text{holdRate}_i * \text{STEP_RATIO}), \text{minGuaranteeRate})$

$C.dnRate_i = \text{MIN}(\hat{C}.dnRate_{i+} + (\text{holdRate}_i * \text{STEP_RATIO}), \text{holdRate}_{i-} - \text{minGuaranteeRate})$

ENDIF

INTER-GREEDY:

$C.upRate_i = \hat{C}.upRate_{i+} + (\text{borrowRate}/2)$

$C.dnRate_i = \hat{C}.dnRate_{i+} + (\text{borrowRate}/2)$

NON-GREEDY:

$C.upRate_i = \hat{C}.upRate_{i-} - (\text{totalBorrow} * ((\text{remainingUpBW}_i) / \text{totalRemainingBW}))$

$C.dnRate_i = \hat{C}.dnRate_{i-} - (\text{totalBorrow} * ((\text{remainingDnBW}_i) / \text{totalRemainingBW}))$

Default:

ENDCASE

}

Then, generate and execute the *tc* script according to $C.upRate_i$ and $C.dnRate_i$.

Algorithm 1 provides an overview of our bandwidth allocation algorithm. The bandwidth allocation process is based on the greedy status of each wireless station. First, the remaining bandwidth can be calculated by subtracting the current consumed bandwidth and the minimum guarantee bandwidth of non-greedy stations from their held bandwidth, i.e., initially half of the fair rate, on both uplink and downlink directions. The notation for current held bandwidth for uplink and downlink directions of station *i* and its newly allocated bandwidth for each direction are $\hat{C}.upRate_{i-}$, $\hat{C}.dnRate_{i-}$, $C.upRate_i$, and $C.dnRate_i$, respectively.

Next, *borrowRate* is calculated from the sum of the remaining bandwidth of all clients in both uplink and downlink directions, divided by the number of clients. This *borrowRate* will be given to the inter-greedy stations only.

The next part of the algorithm adjusts the allocated bandwidth for each direction of each station, according to their greedy status. For intra-greedy stations, the *holdRate_i* parameter, which is the sum of current held bandwidth for uplink and downlink directions, is calculated. Then, their rate will be adjusted between uplink and downlink directions within the station. If an uplink flow is greedy, the uplink rate will borrow the bandwidth from the downlink rate, and vice versa. However, in order to avoid the sharp fluctuation in the communication, the algorithm carefully adjusts the rate by using the STEP_RATIO parameter, which is a value between 0 and 1. Essentially, the adjusted rate for an intra-greedy station depends on this parameter.

For inter-greedy stations, they will borrow bandwidth from non-greedy stations. Both uplink and downlink

rate of inter-greedy stations will be equally increased by half of *borrowRate*.

For non-greedy stations, their bandwidth rate on both uplink and downlink flows will be proportionally decreased as a function of the ratio of the remaining bandwidth on each queue to the total remaining bandwidth in the wireless network.

After the bandwidth is completely allocated to each station, the consumption table will be updated. The new bandwidth rate will be added. This consumption table will be kept in the database and will be used again in the next period of traffic control process. Finally, a *tc* script will be generated in order to change the rate in traffic control rules and will be executed on the wireless access router.

4.3. Contention window setting

The previous adaptive bandwidth allocation for per-station fairness is based on traffic policing, which works on the network layer. However, this process can provide per-station fairness if and only if there is only downlink traffic in the network. If multiple of greedy uplink flows exist in the network, per-station fairness could not be achieved.

One reason behind this is because of the DCF mechanism provided in the IEEE 802.11 MAC layer, which aims to grant equal media access opportunity to all stations in the network including access point and its client. As mentioned earlier, it implies that downlink traffic is which transmitted from the access point may be dominated by uplink traffic because the bandwidth that the access point can utilize to accommodate all downlink flows is just $1/N$ of the total bandwidth. The domination of uplink flows is per-flow unfairness and could also give rise to per-station unfairness.

Thus, apart from traffic controlling for per-station fairness, which is done on the IP layer, the backoff time of the wireless access point should also be shorter than those of the wireless clients in order to increase the transmission opportunities for downlink traffic. As described in the background section, CW is a parameter used to control the backoff time. Therefore, to overcome the domination of uplink, we adopt the idea from [23] which minimizes the backoff time by decreasing the minimum value of contention window (CW_{min}) parameter in the MAC layer of the access point.

However, contrary to the previous work, to reduce a chance of fluctuation, we do not adaptively change CW_{min} according to the number of wireless clients. Here, CW_{min} is set to a constant value.

As suggested in [1], the initial value of CW_{min} should be 7. However, for the current IEEE 802.11 a/b/g/n wireless network interface cards (NIC), the CW_{min} parameter completely depends on manufacturers. Moreover, some wireless clients, attempting to generate

denial of service (DoS) attacks, may configure their CW parameter to a very small value. As we cannot know the exact CWmin value of all wireless clients, and some clients may be malicious, we suggest that CWmin of access point should be configured to possible minimum value to overcome uplink traffic from clients. Our suggested value of CWmin on the wireless router is 3. Experiments to derive this value are briefly described in Section 6.

5. Implementation and experiment configurations

To illustrate the effectiveness of our algorithm, we implement our scheme on a laptop-based wireless router with MadWifi driver, which is a set of Linux kernel drivers for Wireless LAN devices with Atheros chipsets. As mentioned earlier, there are three main modules in our system: packet sniffing, traffic analysis, and adaptive bandwidth allocation. Packet sniffing is implemented by using the *libpcap* library.

The traffic analysis and adaptive bandwidth allocation modules are implemented by using C-language and a Shell script. The number of active clients in the WLAN is automatically learnt by intercepting the result of the *wl* command [32] and the DHCP lease file or intercepting the result of the *arp* (address resolution protocol) command and the *wlanconfig* command [33]. The value of CWmin is modified by using the *iwprev* command [34], which is a utility on Linux for configuring parameters of a wireless network interface.

Our testbed consists of several desktops connected via a 10/100 switch to one laptop functioning as a wireless access router. All devices are located within 1-2 m apart from the access router. Each desktop is equipped with a Linksys WUSB54G wireless interface card, which supports IEEE 802.11 g. These desktops run Windows XP Service Pack 3, while the access router runs Linux Redhat Enterprise 5 with the MadWifi driver. The auto-rate function is also enabled, because it is the default configuration of most wireless clients and we would like to demonstrate the effectiveness of our solution when influenced by heterogeneous performance of wireless clients.

Specifications of hardware and software used in our experiments are described in Table 2.

The STEP_RATIO parameter is set to 20% of its hold rate. However, in a real deployment, the administrator

of the wireless access router can set this parameter to a specific value via our program interface.

The configurations of our *tc* script are described below.

- The *tc* script will be applied to both the wired Internet and wireless interfaces.
- The main *qdisc* is HTB.
- The number of classes and filters depends on the number of stations.
- Each class specifies the rate in a consumption table.
- The uplink rate is defined on the wired Internet interface and the downlink rate is defined on the wireless interface.
- The *qdisc* of each class is TBF with 1 ms of latency and 20 kb of burst setting.
- Each filter classifies traffic based on the IP address of the wireless station.

To measure the achieved throughput of each wireless station, the *Iperf* measurement tool version 1.7.0 is used. *Iperf* can run in the client or server modes. Therefore in our experiments, senders of any flow run *iperf* in the client mode, while the receivers run in the server mode. *Iperf* servers are configured to report the achieved throughput for every second.

In our adaptive bandwidth allocation algorithm, the default capacity of WLAN is assumed to be 20 Mbps, while the minimum guaranteed rate of each direction of a wireless station is defined as 0.5 Mbps. The estimated link capacity is derived from our preliminary experiments and observations. In addition, the interval time to periodically process our adaptive allocation algorithm is 10 s. All experiments are measured for 60-120 s and performed at least three times in order to obtain an average value.

6. Evaluation

To verify the effectiveness and scalability of our proposed scheme, we performed numerous experiments which can be categorized into six parts: (1) finding an appropriate contention window for the wireless access router, (2) per-station fairness of inter-greedy downlink-only traffic, (3) fairness of uplink and downlink flows, (4) fairness

Table 2 Hardware specifications

Specification	Wireless station	Wired station	Laptop-based wireless access router
Processor	Intel Core 2 Duo 2.0 GHz	Intel Core 2 Duo 2.0 GHz	Intel Pentium M 1.3 GHz
Memory	1 GB	1 GB	736 MB
Operating System	Microsoft Windows XP Service Pack 3	Microsoft Windows XP Service Pack 3	Linux Redhat Enterprise 5
Network Interface	Linksys WUSB54G	Ethernet 10/100 Mbps	TP-Link TL-WN510G
Network Driver	-	-	Madwifi 0.94

when multiple flows in the same direction exist in the same wireless client, (5) fairness among inter-greedy uplink and downlink flows, and (6) adaptive bandwidth allocation.

In our algorithm, UDP flows with the higher transmission rate than the fair rate and TCP flows are considered greedy. TCP flows are burst flows, since their additive increase/multiplicative-decrease (AIMD) algorithm, which is a feedback control algorithm, increases the transmission rate for every RTT (round trip time) until a packet loss is detected.

6.1. Finding appropriate contention window for the wireless access router

This part of experiment aims to study the relationship between the CWmin parameter and the achieved throughput in order to configure an appropriate value of CWmin on our wireless access router. As mentioned in Section 4.3, the suggested value of the initial CWmin is 7. Therefore, we focus on a lower value ranging between

1 and 5, while the value of maximum contention window (CWmax) is fixed to 10. The network topology consists of two wired computers and two wireless clients connecting via our laptop-based wireless access routers. UDP flows are transmitted at the rate of 30 Mbps. Four cases we consider: (1) one UDP uplink flow existing with one UDP downlink flow, (2) one TCP uplink flow existing with one UDP downlink flow, (3) one UDP uplink flow with one TCP downlink flow, and (4) one TCP uplink flow with one TCP downlink flow.

Figure 5 shows the average throughput of each flow in four cases. As we can observe from the graphs, the throughput of the downlink flow can overcome that of the uplink flow when CWmin at the access point is less than 5. The throughput of the uplink flow tends to increase as the value of CWmin increases. This phenomenon can be observed in all cases except when the uplink flow is TCP and the downlink flow is UDP. Therefore, in the remaining parts of our experiments, we configured the value of CWmin to below 5.

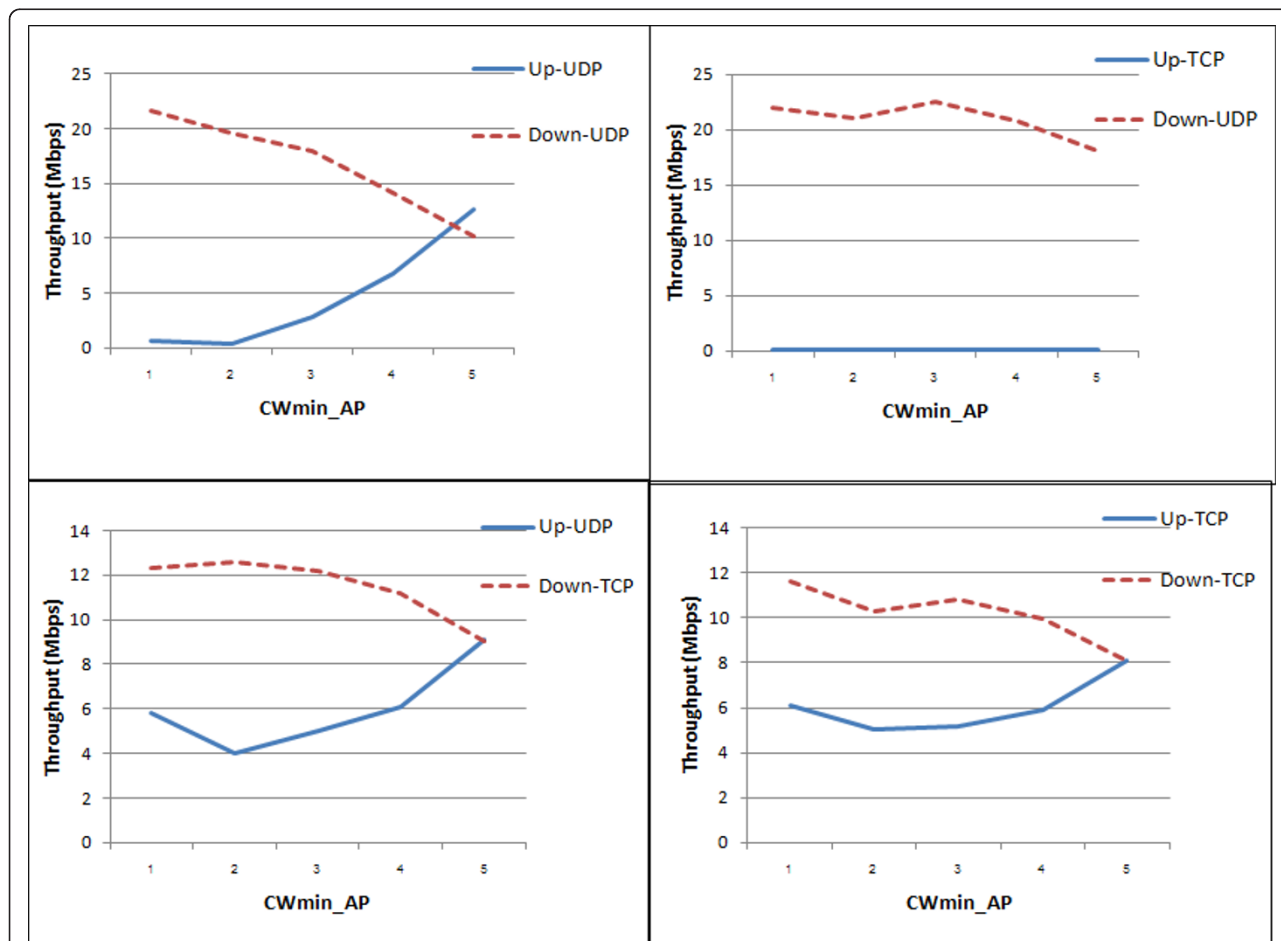


Figure 5 The relationship between CWmin of the access router and the average throughput of UDP/TCP flows in both directions.

6.2 Per-station fairness for inter-greedy downlink-only traffic

In this part of experiments, the value of CW_{min} at the wireless access router is set to 3. The fairness performance of our scheme is compared with that of the legacy DCF mechanism. Three experiment scenarios in this part are as shown in Table 3.

6.2.1. Scenario B-1: Five TCP downlink flows and five UDP downlink flows

In order to test performance and scalability of our proposed scheme, ten wireless clients are employed. Figure 6 illustrates the topology of this experiment. It consists of five desktop computers connected to a wireless access router via a 10/100-Mbps switch and ten desktop computers as wireless clients connected to the access router via the IEEE 802.11 g wireless network cards. Each wired computer transmits one TCP flow and one UDP flow with the rate of 4 Mbps to wireless clients. Since our estimated wireless capacity is 20 Mbps and there are ten wireless clients, the fair rate of each client is 2 Mbps.

The upper part and lower part of Figure 7 depict the average throughput of each TCP and UDP flow when our scheme is not applied, i.e., the legacy DCF scheme, and when our WLAN-ATC scheme is applied, respectively. Figure 7 reveals that, with the legacy DCF scheme all five UDP flows achieve the throughput above 3 Mbps, while all five TCP flows achieve less than 1 Mbps. However with our proposed scheme, all UDP and TCP flows obtain the throughput around 1-2 Mbps, which is close to the fair rate of this system. The results confirm that our scheme can ameliorate per-station fairness for TCP and UDP downlink flows even when there are many clients in the WLAN.

The reasons why TCP flows cannot obtain 2-Mbps throughputs as same as UDP flows are because of (1) the TCP congestion control mechanism, (2) the minimum guaranteed bandwidth, and (3) the unpredictable wireless capacity. For TCP congestion control mechanism, TCP flows are normally greedy flows, as they will double the congestion window, i.e., the transmission rate, whenever they do not detect congestion. In other words, there is some remaining bandwidth. On the contrary, it will decrease the rate by half when a packet loss is detected.

Table 3 Experiment setup for downlink-only traffic

Scenario no.	Number of flows		Details
	TCP	UDP	
B-1	5	5	Each UDP flow is 4 Mbps
B-2	1	9	All flows are greedy
B-3	-	10	Five flows start at 0 s, the rest starts after 60 s

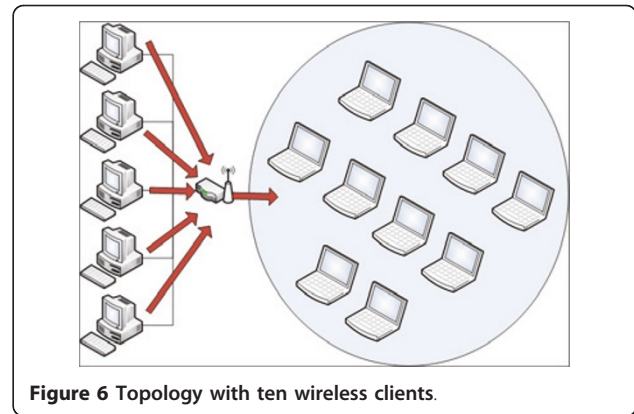


Figure 6 Topology with ten wireless clients.

As the minimum guaranteed bandwidth for each direction of each flow is 0.5 Mbps and there are ten downlink flows in this experiment, thus totally 5 Mbps of the link capacity is wasted for the minimum guaranteed bandwidth of ten uplink flows, which may occur subsequently. It is worth pointing out that the value of the minimum guaranteed bandwidth can be fine-tuned by the network administrator.

For the maximum link capacity, in the experiment we set the maximum capacity to 20 Mbps. However, since the wireless condition is unpredictable, the actual link capacity may sometimes be as low as 16 Mbps. Note that, the wireless capacity is unpredictable because of noise and influences from circumstances such as traffic from other APs or wireless stations using the same wireless channel.

6.2.2. Scenario B-2: Nine UDP downlink flows and one TCP downlink flow (all greedy flows)

In this experiment, there are totally ten wireless clients and five wired stations as in the previous experiment. However, all nine UDP downlink flows are transmitted at the rate of 3 Mbps, which is higher than the 2-Mbps fair rate. Another flow is the TCP downlink flow. Figure 8 compares the throughput performance of the system before and after applying our scheme. As shown in the upper part of the figure, the TCP flow seldom gains any throughput for the standard IEEE 802.11 scheme. However, we can observe in the lower part of the figure that with our WLAN-ATC scheme the TCP flow finally achieves the similar amount of bandwidth as those of UDP flows. The results confirm that our scheme can provide fairness even when there are a large number of wireless clients and all traffic flows are greedy.

6.2.3. Scenario B-3: Ten UDP downlink flows with the real-time adaptive allocation

The goal of this experiment is to evaluate the ability of our scheme to dynamically re-allocate the wireless bandwidth when the number of wireless clients increases during the time of measurements. The

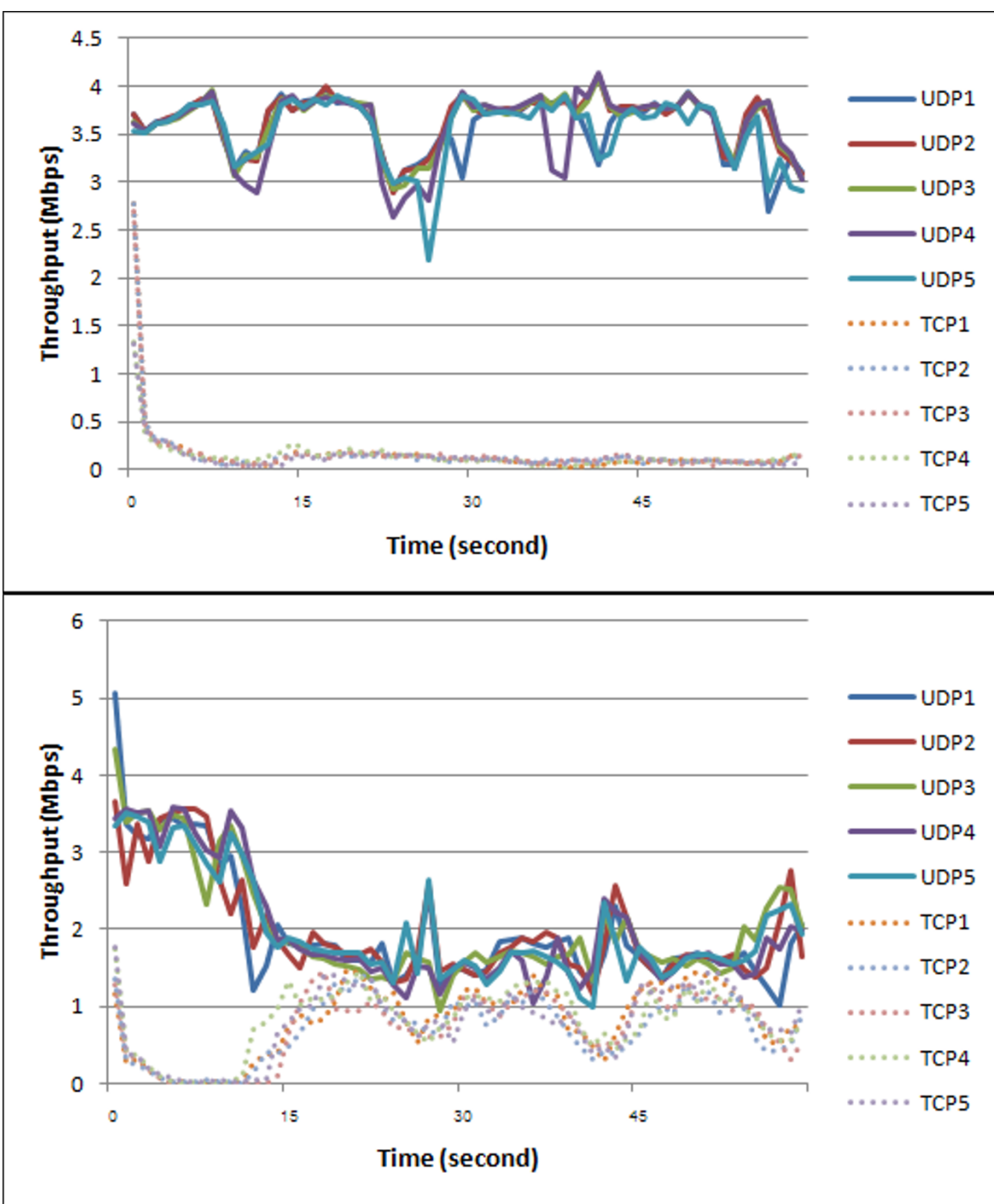


Figure 7 Average throughput of five UDP downlink flows and five TCP downlink flows (the upper part: with standard DCF scheme, the lower part: with our WLAN-ATC scheme).

experiment topology is the same as in Figure 6. However, only five UDP downlink flows with the rate of 4 Mbps are transmitted to five wireless clients at time 0 s. After 60 s elapsed, another five UDP downlink flows with the same rate begin to transmit. Figure 9 depicts

the change in the average throughput of each flow. At time 0 s, the average throughput of each flow is about 3-3.5 Mbps.

After the 60th-s of test time, the throughputs of five new coming UDP flows dominate the old UDP flows.

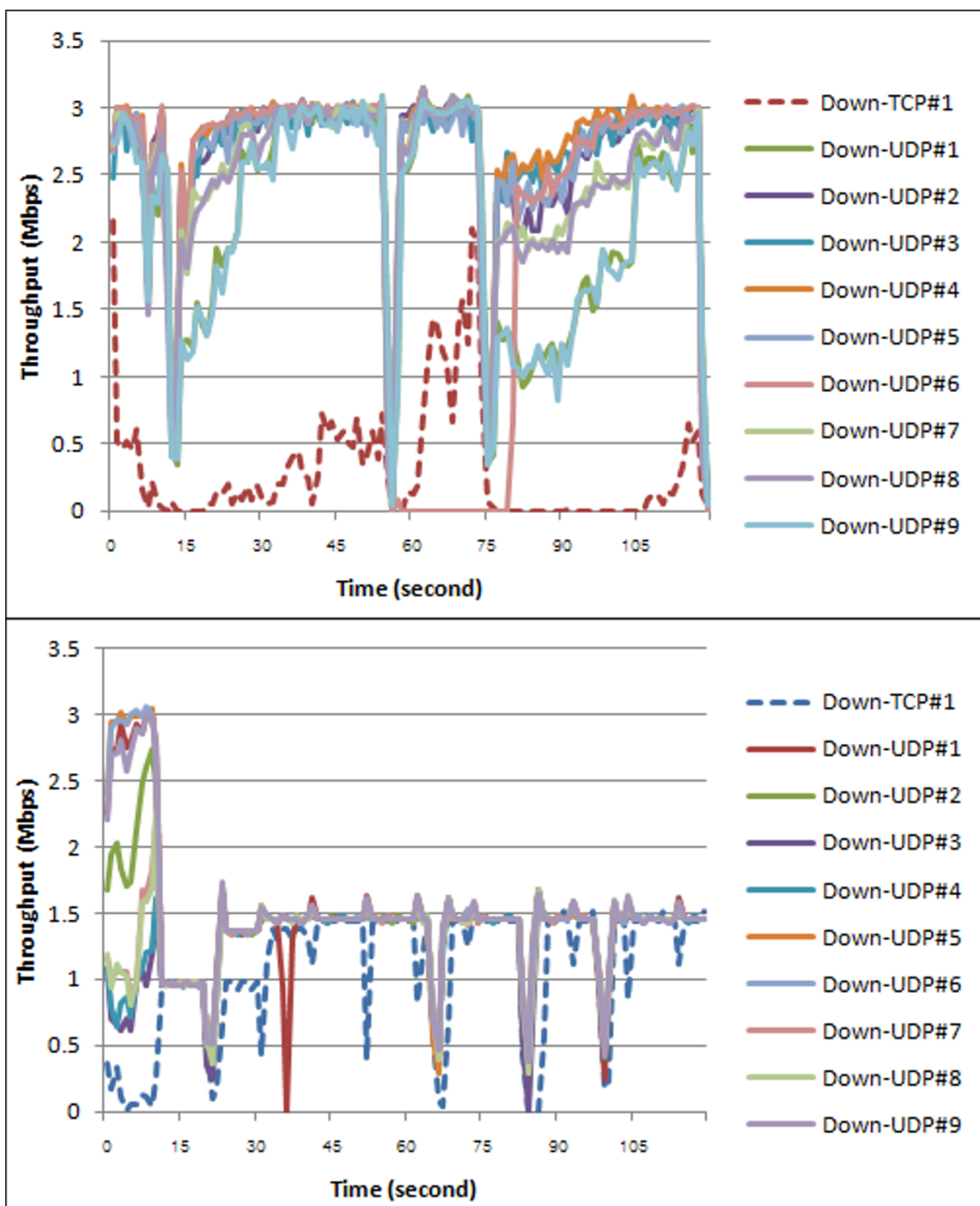


Figure 8 Average throughput of nine greedy UDP downlink flows and one TCP downlink flow (the upper part: with standard DCF scheme, the lower part: with our WLAN-ATC scheme).

However, after our scheme is applied at the 70th-s of test time, the fair rate of each flow is re-calculated, and the average throughput of each flow is stabilized around 1.5 Mbps, which is close to the 2-Mbps fair rate for ten wireless clients.

Table 4 summarizes the Jain's fairness index results for each scenario. The results confirm that our proposed

mechanism can significantly improve per-station throughput fairness.

6.3. Per-station fairness when both uplink and downlink flows exist

In this part of experiments, per-station fairness when both uplink and downlink flows exist is investigated.

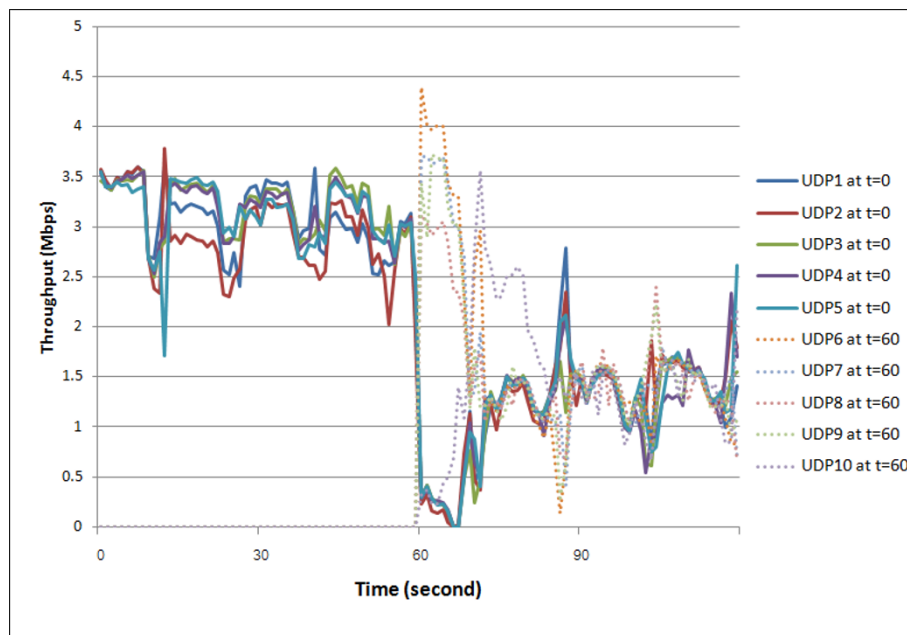


Figure 9 Average throughput of ten UDP downlink flows with our scheme, while five of them start transmission at the 60th second.

The value of CW_{min} at the wireless access router is set to 3. The network topology is the same as depicted in Figure 6 and there are ten wireless clients. There are two scenarios in this experiment: (1) one TCP uplink flow with nine greedy UDP downlink flows, and (2) one UDP uplink flow with ten greedy UDP downlink flows.

6.3.1. Scenario C-1: One TCP uplink flow and nine greedy UDP downlink flows (inter-greedy case)

In this experiment, nine UDP downlink flows with the rate of 3 Mbps, each of which is transmitted from five wired-network computers, while a TCP uplink flow is transmitted from a wireless client. Figure 10 compares the average throughputs of the system when the standard DCF scheme and our scheme are employed. In the standard DCF scheme, the TCP throughput is almost zero, while the throughput of each UDP downlink flows is highly fluctuated between 0.5 to 3 Mbps. The reason behind this large fluctuation is that all UDP flows are greedy when comparing to the fair rate of 2 Mbps. Therefore, there is not enough bandwidth for all stations. The results in the lower part of the figure with our scheme confirm that our scheme can ameliorate the

fairness performance since all flows including TCP flows can achieve throughput equally.

6.3.2. Scenario C-2: One greedy UDP uplink flow and ten greedy UDP downlink flows (intra-greedy case)

In this experiment, we emulate the situation that all ten stations are greedy; and one of ten stations contains both greedy uplink and downlink flows. Ten UDP downlink flows are transmitted to each of ten wireless stations at the rate of 3 Mbps. Moreover, from one of these stations, one UDP uplink flow is also transmitted at the rate of 5 Mbps. If the bandwidth is fairly allocated to each station, the fair rate will be 2 Mbps, since the estimated wireless capacity of 20 Mbps is divided by ten stations. Figure 11 compares the differences in the average throughput when the standard DCF scheme and our WLAN-ATC scheme are employed. The dotted line represents the total throughput of uplink and downlink flows of station 1. We can observe that with the standard DCF scheme, the only UDP uplink flow obtains a higher bandwidth than all downlink flows. However, with our WLAN-ATC scheme in the lower part of the figure, the results show a great improvement in fairness of the system.

Table 5 summarizes Jain’s fairness index of each scenario. The results verify the effectiveness of our scheme when both uplink and downlink flows of TCP and UDP exist.

Table 4 Jain’s fairness index for downlink only experiments

Scenario no.	Number of flows		Jain’s fairness index	
	TCP	UDP	DCF	WLAN-ATC
B-1	5	5	0.542	0.834
B-2	1	9	0.917	0.992
B-3	-	10	-	0.827

6.4. Fairness when multiple flows in the same client

In this part of the experiment, we intend to demonstrate the ability of our scheme even when multiple flows exist on the same client. Here, five wireless clients receive

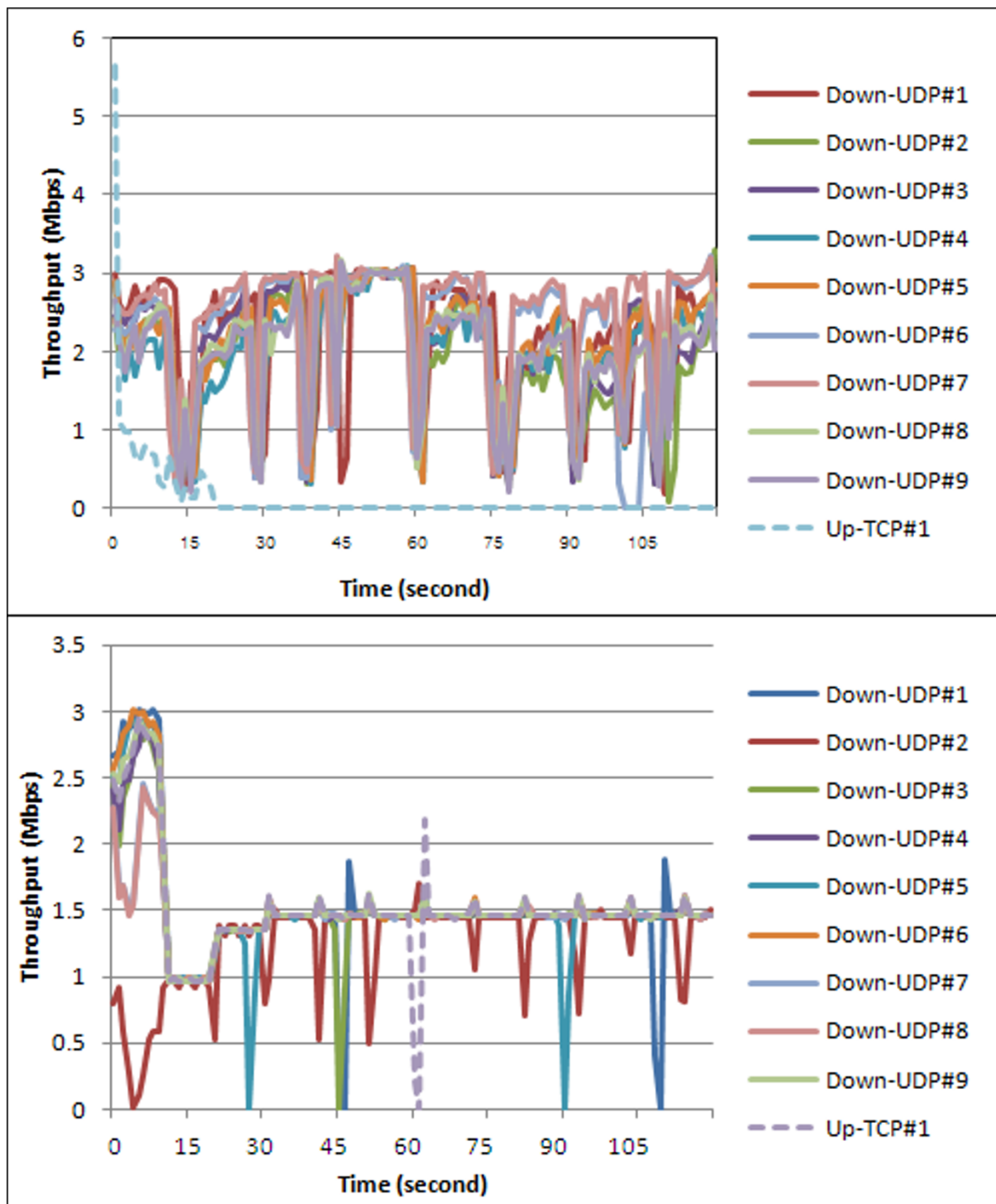


Figure 10 Average throughput of nine UDP downlink flows and one TCP uplink flow (the upper part: with standard DCF scheme, the lower part: with our WLAN-ATC scheme).

one UDP downlink flow, and other five clients receive two UDP downlinks. The transmission rate of each flow is 3 Mbps. Figure 12 compares the results before and after deploying our scheme. The notation '-2F' in the

figure represents 'two flows', as five clients from number 1 to 5 each receive two UDP flows. Moreover, the average throughput in these graphs is per-station, not per-flow.

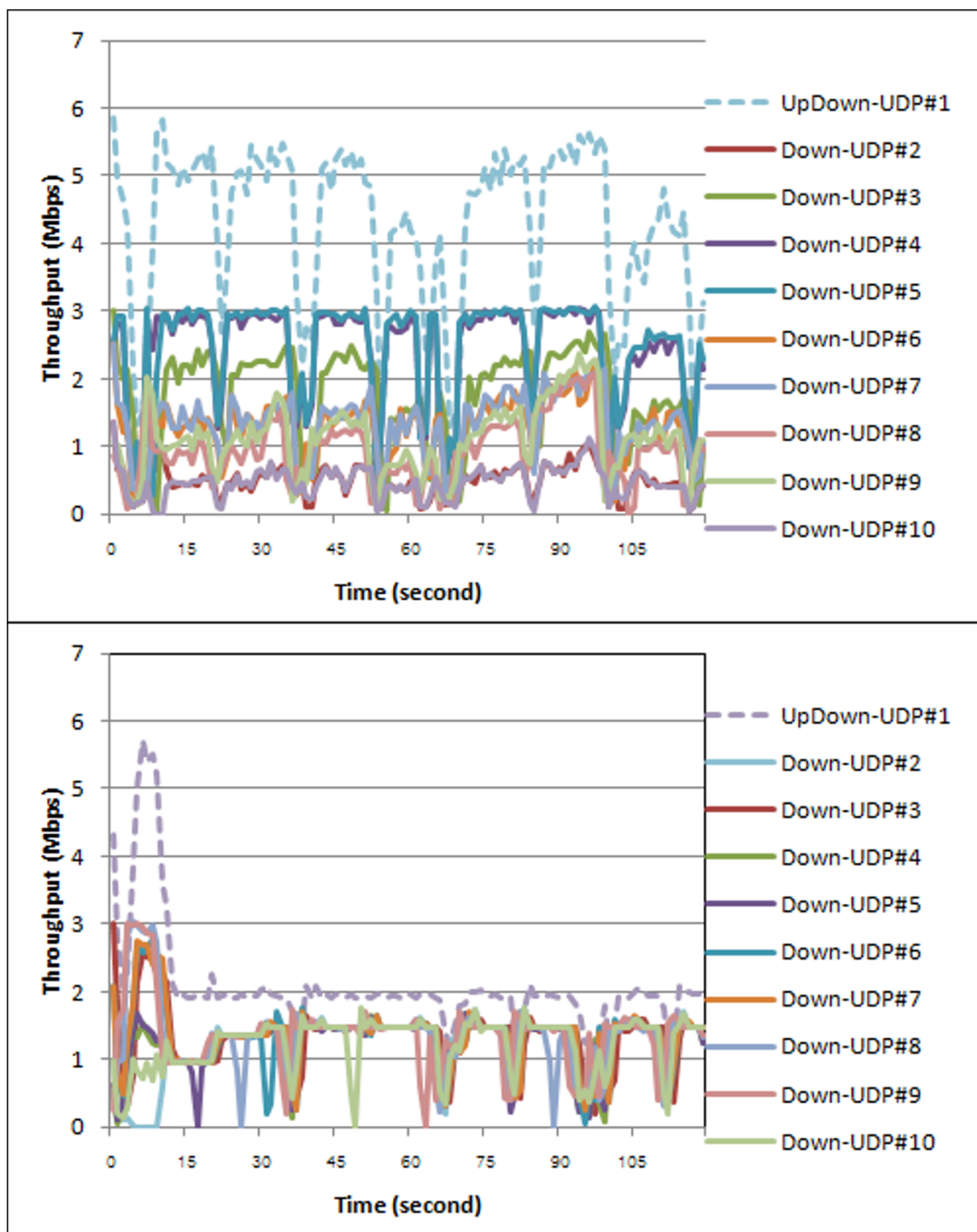


Figure 11 Average throughput of ten UDP downlink flows and one UDP uplink flows (the upper part: with standard DCF scheme, the lower part: with our WLAN-ATC scheme).

As in the upper part of Figure 12, for the standard DCF scheme, stations with multiple flows will gain higher total throughputs than stations with a single flow. However, as observed in the lower part of the

figure, with WLAN-ATC, all stations achieve similar average throughputs. The effectiveness of our scheme can be confirmed by the fairness index, since Jain's fairness index for the standard DCF scheme and our

Table 5 Jain’s fairness index for uplink and downlink experiments

Scenario no.	Number of flows		Jain’s fairness index	
	TCP	UDP	DCF	WLAN-ATC
C-1	1 Up	9 Down	0.907	0.997
C-2	-	9 Down, 1 UP + Down	0.695	0.970

WLAN-ATC scheme are 0.69997 and 0.95518, respectively.

6.5. Fairness of intra-greedy flows

This part of the experiments aims to illustrate performance of the proposed scheme when there exist multiple flows with different directions in the same client. The CWmin parameter of the wireless router is configured to 1. The network topology is illustrated in Figure 13 where two wired stations perform as transmitters and there are three wireless clients named A, B, and C. Each UDP flow is transmitted at the rate of 20 Mbps. Station A transmits both uplink and downlink flows, which are either TCP or UDP traffic.

There are four scenarios in this experiment. Figures 14, 15, 16 and 17 illustrate the experiment results for each of the four scenarios, respectively. The notation ‘Station A’ represents the total throughput of its uplink and downlink flow. On the other hand, the dashed line represents the throughput of each flow on the station A. Table 6 summarizes the experiment setup for all four scenarios and its fairness index based on our WLAN-ATC scheme.

The experiment results in Figures 14 and 15 reveal that our algorithm can equally share the bandwidth within the same station when the data in both directions are the same traffic type. However as observed in Figures 16 and 17, when there are both TCP and UDP flows in the same station, but in different directions, the TCP flow cannot acquire further bandwidth. This problem will be discussed later in Section 7. Moreover, as shown in Table 6, Jain’s fairness index of each scenario is higher than 0.99. Thus, our scheme can achieve the per-station throughput fairness.

6.6. Adaptive bandwidth allocation

In the last part of experiments, we validate the effectiveness of our adaptive bandwidth control algorithm, as a greedy station should be able to gradually obtain the remaining bandwidth. For a greedy flow, we consider a high bit rate of UDP flow or TCP flow, since TCP flow will increase the rate whenever they cannot detect a packet loss, i.e. there is some remaining bandwidth. We divide experiments into two scenarios: greedy among stations and greedy between uplink and downlink flow within a station.

6.6.1. Scenario F-1: greedy among stations when one greedy UDP downlink flow and one TCP downlink flow

In this experiment, there are totally ten wireless clients and five wired stations as depicted in Figure 6. Each wireless client transmits one flow: eight UDP downlink flows with the rate of 1 Mbps each, one UDP downlink flow with the rate of 5 Mbps, and one TCP downlink flow. Since the estimated link capacity is 20 Mbps and the number of wireless clients is ten clients, the fair rate for each client is 2 Mbps. Therefore, there are two greedy flows in this network: 5-Mbps UDP flow and TCP flow. The experiment lasts for 120 s.

As shown in Figure 18, at time 0 to the 10th-s when our scheme is not yet applied, the 5-Mbps UDP flow and TCP flow consume a large portion of the available bandwidth. After our algorithm is executed at the 10th-s, both flows obtain the equal rate of 2 Mbps. After the 20th-s of test time, our algorithm observes the remaining bandwidth. Then, it re-allocates the remaining bandwidth to the greedy 5-Mbps UDP flow and TCP flow. We can learn from the graph that both of them gradually obtain more bandwidth. The highest bandwidth they can obtain is about 3.5-4 Mbps.

6.6.2. Scenario F-2: Two small UDP flows and two greedy UDP flows within the same station

As illustrated in Figure 13, the network topology consists of two wireless stations, each receiving one UDP flow with 1 Mbps, and another wireless station transmits both UDP downlink and uplink flows at 20 Mbps in each direction. The results in Figure 19 show that after 10 s, the maximum bandwidth of each wireless station is equally allocated to the capacity of the network, i.e., 20 Mbps, divided by the number of stations, i.e., three stations. That is why the summation throughput of uplink and downlink flow of station A is about 7 Mbps. After that, our mechanism detects the remaining bandwidth. Therefore, we can observe from the figure that the throughput of station A, which is a greedy station, gradually increases.

The experiment results in both scenarios confirm that our adaptive bandwidth control algorithm can successfully re-allocate the remaining bandwidth to both greedy UDP flows and TCP flow.

7. Discussion

This section presents discussions about our proposed schemes

- Channel diversity (near-far effect): Normally, wireless stations experience different propagation paths and thus they observe different and time-varying transmission rates. However, we do not include this time-varying characteristic in our traffic control mechanism consideration, because the remaining

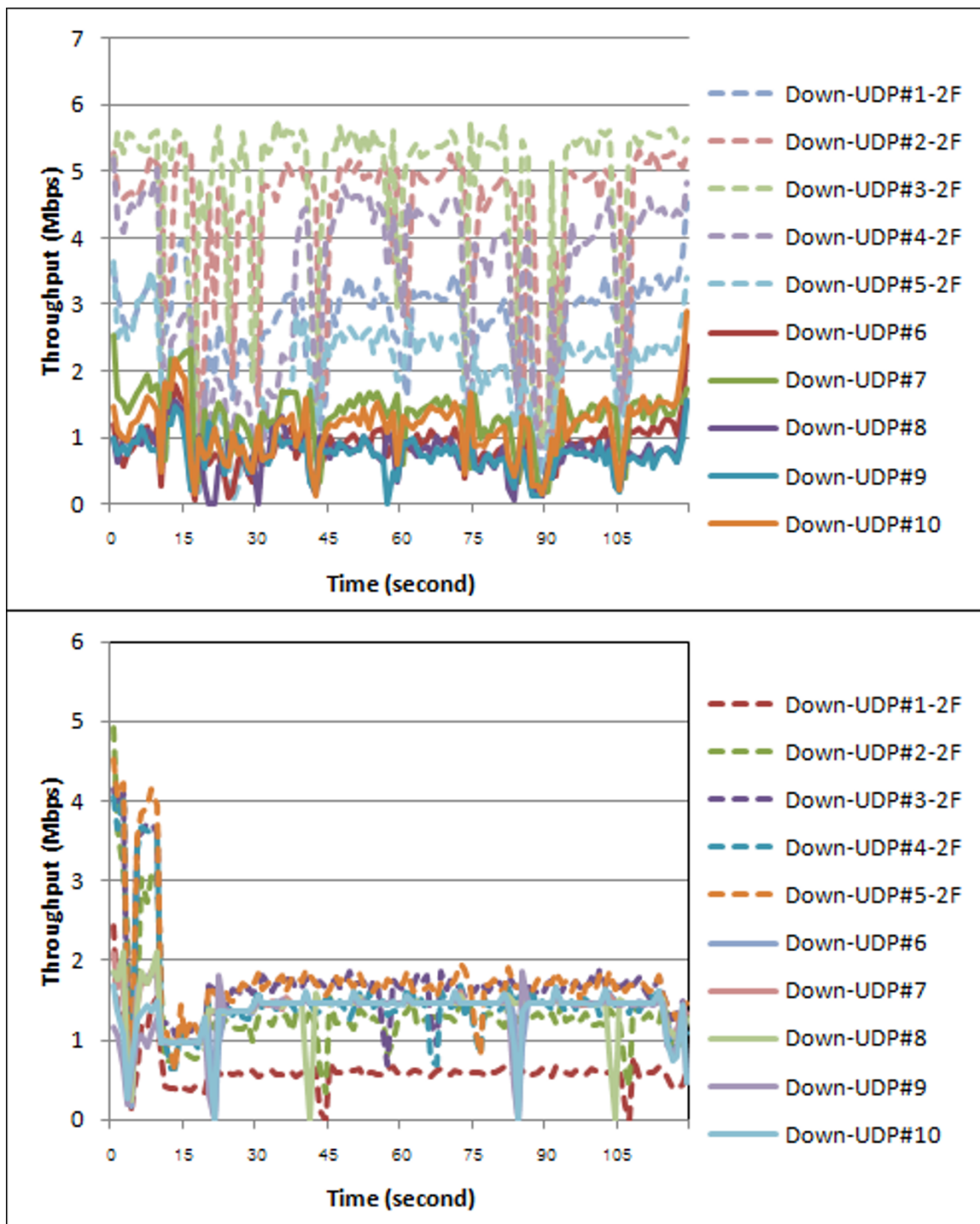


Figure 12 Average throughput of ten wireless clients, while five clients transmit two UDP downlink flows and the rest transmits one UDP downlink flow each (the upper part: with standard DCF scheme, the lower part: with our WLAN-ATC scheme).

bandwidth from stations with the lower transmission rate will be taken into account and will be allocated to other stations that require more bandwidth or can gain higher transmission rates.

○ Inactive wireless clients: When a wireless station associates with the access point, it will automatically acquire a fair rate, which our access router equally allocates to each client. In the case when there is no

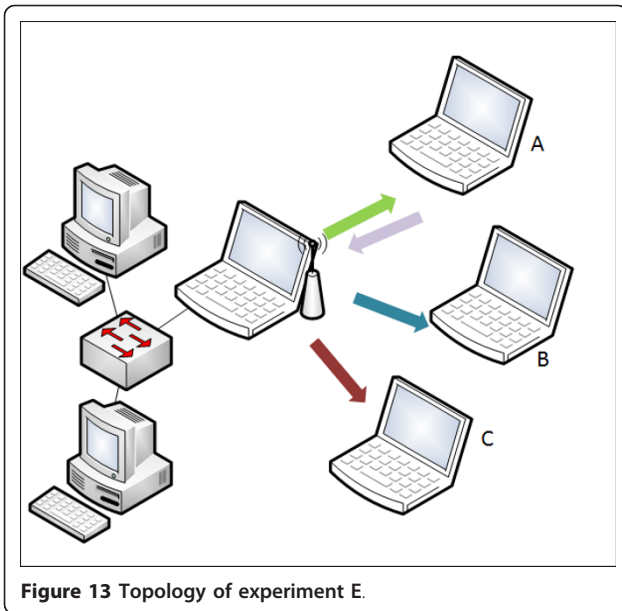


Figure 13 Topology of experiment E.

network activity, this amount of bandwidth will be useless. However, as we introduce the adaptive bandwidth allocation for greedy flows, its fair rate will be automatically allocated to other stations. The actual wasted bandwidth from the inactive wireless clients is just their sum of the minimum guaranteed bandwidth, which the WLAN administrator can adjust.

- Support legacy devices: Much work argues that IEEE 802.11e can solve the unfairness problem in the

wireless network. To achieve fairness, it is required that all stations must implement this standard. However, when a scheme needs the implementation on all nodes, we cannot force every node to deploy it. Therefore, we choose to solve the problem only on the access router. Even though newer standards are being developed, in order to handle legacy clients, the solution presented in this article can be used.

- Virus traffic or Dos/DDos attacks from malicious stations: There may also be some malicious nodes, i.e., virus-infected nodes or attacker nodes. Normally, traffic from virus-infected hosts or traffic from the DoS/DDoS attackers can be filtered out by deploying a firewall, an intrusion detection system (IDS), or a virus scanner at the wireless AP or between the AP and the gateway. However, they cannot constrain this unwanted flooding traffic from uplink flows. Therefore, this problem can be alleviated by our proposed scheme.

- Selfishly tuned devices: One kind of misbehaving nodes is a device selfishly tuned its contention window to gain advantage over the AP. Considering when the suggested CWmin for AP in this work is 3 and the selfish station set its CWmin as 1, its uplink flows may gain higher throughputs than downlink flows from the AP. However, in the TCP case, ACK packets are transmitted in the downlink flows to wireless stations. Therefore, the TCP uplink transmission rate will be constrained because the ACK packets forwarded from the AP are constrained. Figure 20 shows the results of our maximum throughput

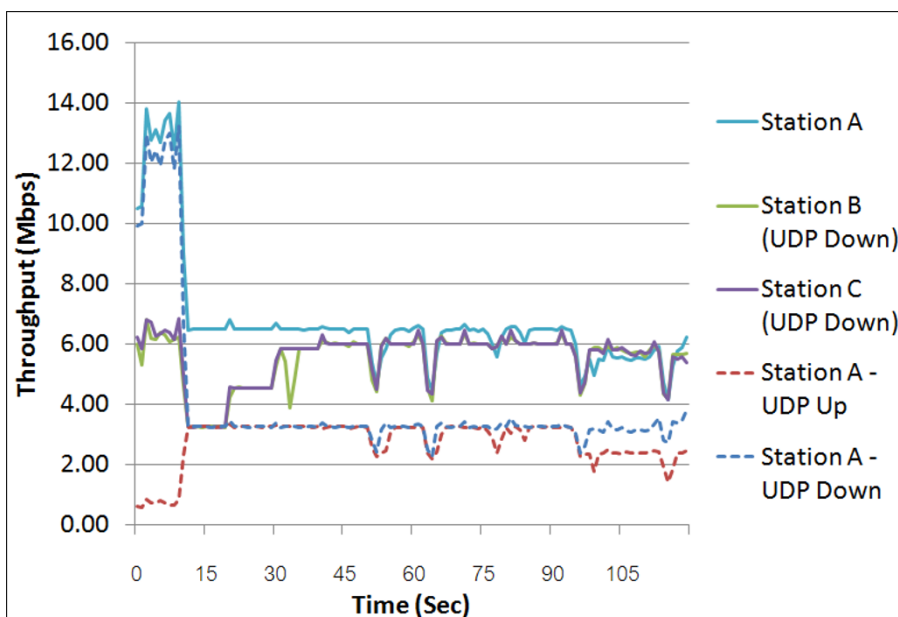


Figure 14 Average throughput of scenario E-1, when station A contains UDP flows in both uplink and downlink directions.

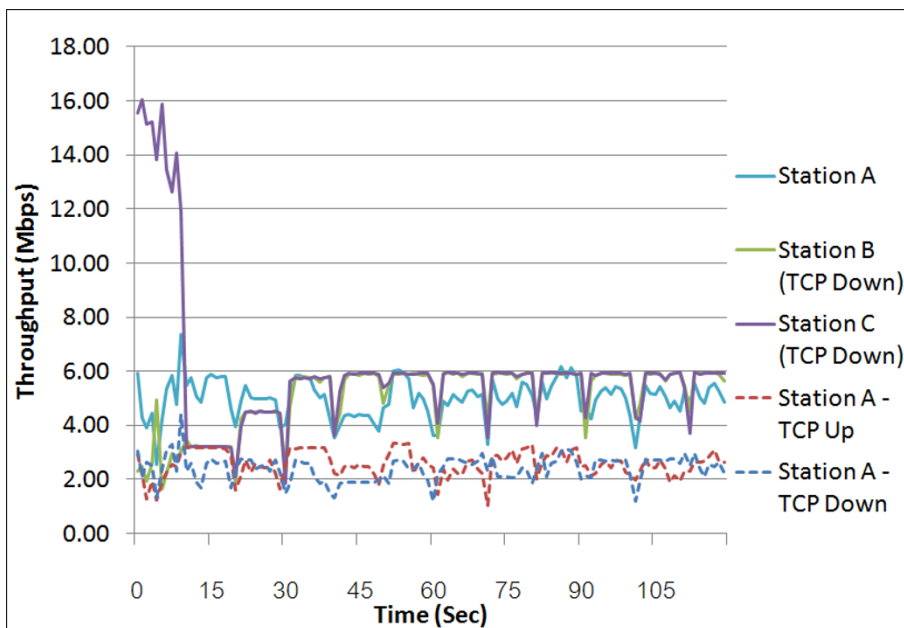


Figure 15 Average throughput of scenario E-2, when station A contains TCP flows in both uplink and downlink directions.

measurement based on Linksys WUSB54G clients. In each measurement, there is only one traffic flow between two devices: one AP and one client. We modified CWmin of the AP from 1 to 10, while CWmin of the client is set to its factory's default value. As shown in Figure 20, throughputs of all flows are dropped as CWmin of the AP decreases, no

matter how CWmin of the client is configured. Except in the UDP uplink case, modifying CWmin of the AP does not work so well. That is the reason why we have to add the traffic policing mechanism at the AP in order to drop excessive UDP uplink traffic.

○ Fluctuation: There may be a fluctuation of throughputs in our WLAN system because of the nature of

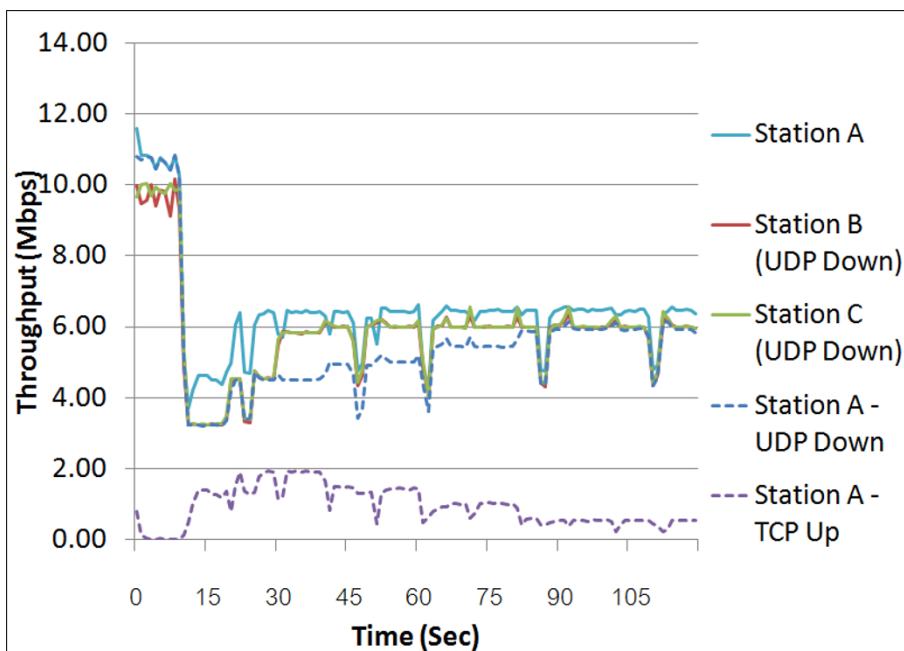


Figure 16 Average throughput of scenario E-3, when station A contains one TCP uplink flow and one UDP downlink flow.

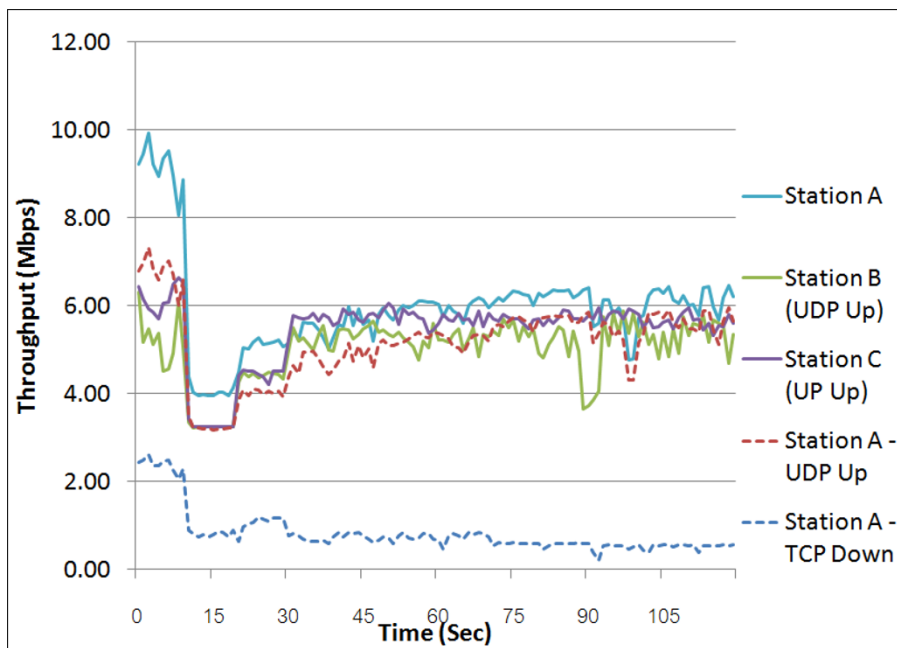


Figure 17 Average throughput of scenario E-4, when station A contains one UDP uplink flow and one TCP downlink flow.

WLAN, which is prone to noise and other interferences. Another factor is the time interval to execute our adaptive traffic control algorithm, in order to update the knowledge about wireless circumstances, e.g., the number of new wireless clients and the current consumed bandwidth. In all experiments, we configured the time interval as 10 s in order to update the rate. However, if the administrator desires a smoother communication, the time interval can be re-configured, such as 5 s, through our user interface. Note that, in the setting of time interval, the administrator should concern about the computational overhead, since the CPU load will increase when the time interval is shorter.

○ Cold start phase: In a multiuser scenario, at the beginning of our algorithm, every user will get a very small share of bandwidth which may induce poor performance on the running application. However, after the first round (first time interval), the algorithm will gradually allocate the remaining bandwidth to greedy applications.

○ Convergence: The convergence time of our proposed scheme depends on two parameters, which are the time interval and STEP_RATIO. As discussed above, in order to avoid fluctuations, the time interval should be set to a small value. This will also bring to a faster convergence in the network. The administrator can also configure STEP_RATIO to a higher ratio in order that the remaining bandwidth will be rapidly allocated to greedy stations.

○ UDP and TCP unfairness within a station: As shown in Figures 16 and 17, our scheme can be easily extended to support co-existence of the UDP and TCP flows within the same wireless client. Currently, we determine the greedy status by detecting the dropped or backlogged packets. However, it is difficult to accurately observe the TCP flow.

○ Link capacity: The effectiveness of the proposed scheme depends on the wireless capacity value. In this research, we define the wireless capacity as the specific and constant value, i.e., 20 Mbps, which conflicts to the fact that the wireless capacity is very

Table 6 Experiment setup and Jain’s fairness index for greedy flows within a station

Scenario no.	Number of flows and its link direction						Fairness index
	Station A		Station B		Station C		
	TCP	UDP	TCP	UDP	TCP	UDP	
E-1	-	1 Down, 1 Up	-	1 Down	-	1 Down	0.9910
E-2	1 Down, 1 Up	-	1 Up	-	1 Up	-	0.9919
E-3	1 Up	1 Down	-	1 Down	-	1 Down	0.9980
E-4	1 Down	1 Up	-	1 Up	-	1 Up	0.9949

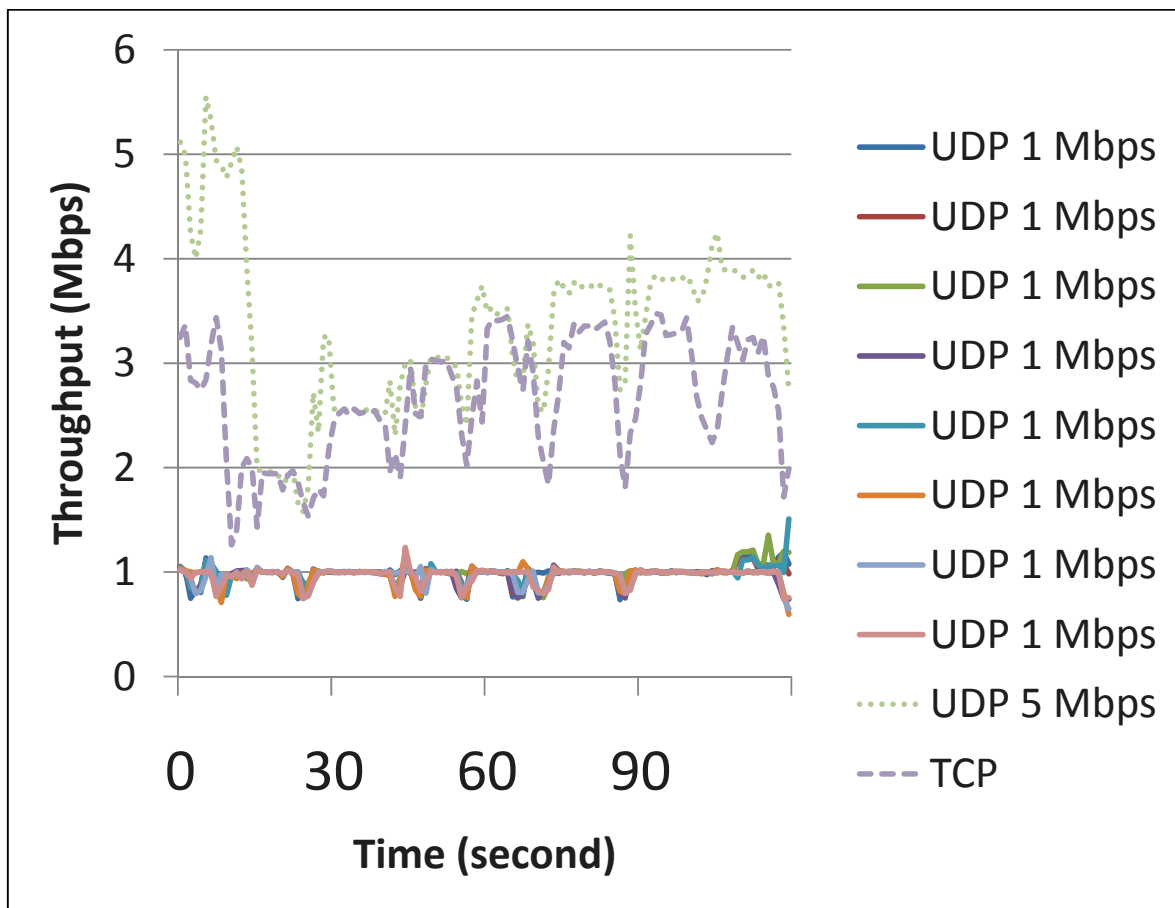


Figure 18 Average throughput of nine UDP downlink flows and one TCP downlink flow (two greedy flows).

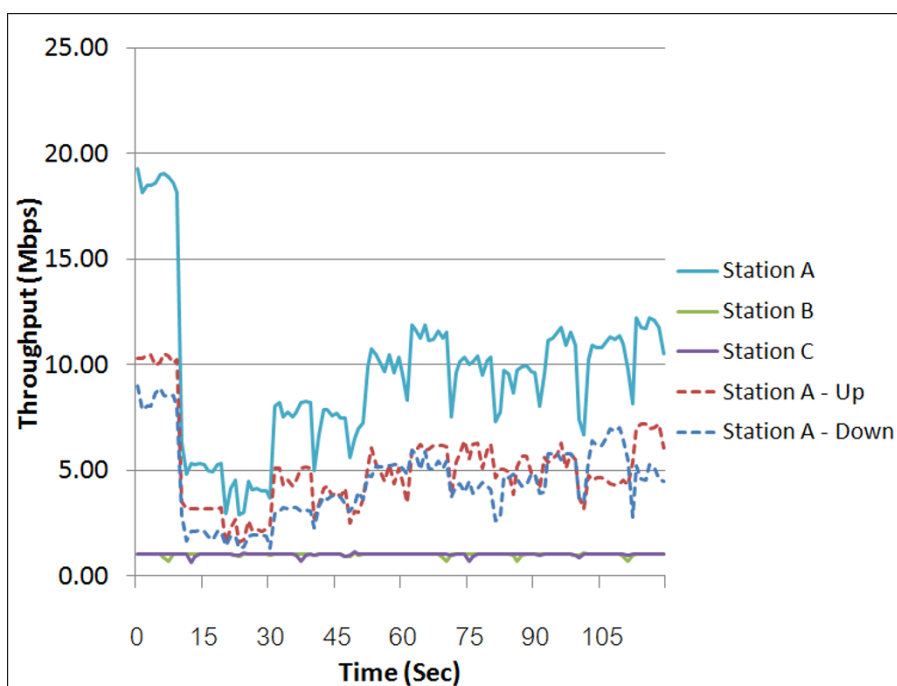


Figure 19 Average throughput of two small UDP flows and two large UDP flows within the same station.

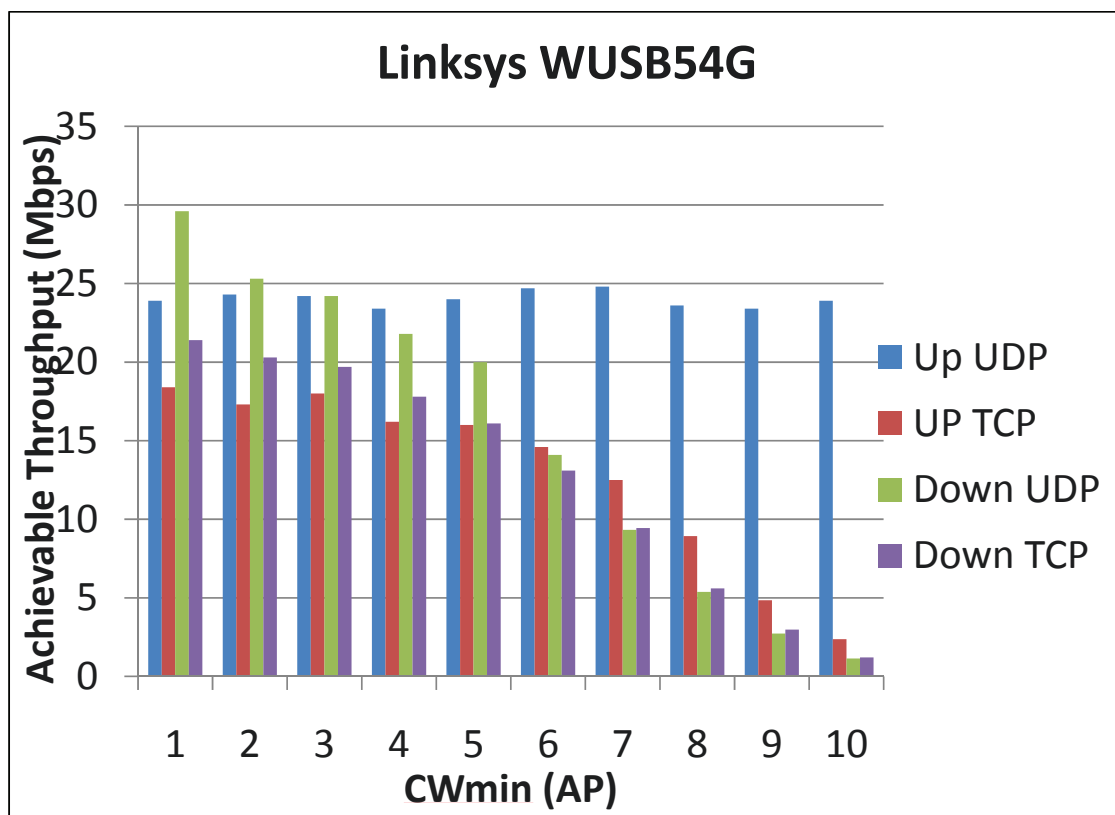


Figure 20 Average maximum throughput when altering the CWmin of AP.

unstable. The achievable capacity indeed is unpredictable, because it depends on the circumstances such as obstacles, noises and interferences from neighboring WLANs using the same frequency channel. The higher capacity setting could degrade per-station fairness while the lower capacity setting could waste the bandwidth. However, the major goal of this paper is not the wireless capacity prediction in the real-time manner. For future work, we aim to study the possibility of utilizing existing wireless capacity prediction tools such as WCE [21,22] and WBest [35] in our mechanism.

○ **Weighted Fairness:** Even though our scheme is proposed to deliver the throughput-based per-station fairness and uplink-downlink fairness, it is straightforward to extend the proposed work to support weighted fairness. For example, one wireless client may be assigned with a throughput that is two times higher than the throughput of the others.

8. Conclusion and future work

In this article, we proposed a novel scheme to provide per-station fairness and uplink-downlink fairness in the wireless network. By observing the current number of

wireless stations, each station will be assigned a fair rate for both uplink and downlink directions. However, the fair rate of each station can be adjusted between its uplink and downlink, if either direction requires more bandwidth than another. For example, the uplink channel can borrow bandwidth from the downlink channel within its station. Moreover, by observing the actual throughput of each station, our scheme can also adaptively allocate more bandwidth to a greedy station, when there is some remaining bandwidth. This technique can practically maximize link utilization.

Experiment results on the real testbed with a large number of stations also confirm that the proposed scheme can meet objectives of this work that it can provide per-station fairness, regardless of the traffic direction. In addition, the proposed scheme is compatible with legacy wireless devices due to all processes are centrally performed on the wireless router.

To overcome the uplink domination, the CWmin parameter on the access router is statically set to a small value. However, this static setting could penalize the overall performance due to the dynamic size of the wireless LAN. When the number of wireless stations increases, CWmin should be adapted to an optimal value to provide the best performance. Therefore, future

work can include fine tuning of STEP_RATIO, estimated wireless link capacity, and the CW_{min} parameter.

List of abbreviations

ACE: access cost estimator; ACK: acknowledgment; AIFS: arbitration interframe space; AIMD: additive increase/multiplicative-decrease; AP: access point; CBQ: class based queuing; CSMA/CA: carrier sense multiple access with collision avoidance; CTS: clear-to-send; CW: contention window; CW_{min}: minimum of contention window; DATC: distributed access time control; DCF: distributed coordination function; DoS: denial-of-service; ECN: explicit congestion notification; FIFO: first-in first-out; GRED: generic random early drop; HTB: hierarchical token bucket; PF: persistence factor; PRIO: priority scheduler; qdisc: queuing disciplines; RTS: request-to-send; RTT: round trip time; SFQ: stochastic fair queuing; TBF: token bucket filter; tc: traffic control; TXOP: transmission opportunity; VSB: virtual shared bottleneck; WCE: wireless capacity estimator; WD: wireless differentiation; WDPA: wireless differentiation with prioritized ACK scheme; WLAN-ATC: wireless LAN adaptive traffic control; WLANs: wireless local area networks; WTC: wireless traffic control.

Author details

¹Faculty of Information and Communication Technology, Mahidol University, 999 Phuttamonthon 4 Rd, Salaya, Nakhon Pathom 73170, Thailand ²National Electronics and Computer Technology Center, 112 Phahon Yothin Rd., Klong 1, Klong Luang, Pathumthani 12120, Thailand

Competing interests

The authors declare that they have no competing interests.

Received: 28 February 2011 Accepted: 17 September 2011

Published: 17 September 2011

References

1. IEEE Std 802.11™-2007, IEEE Standard for Information technology- Telecommunications and information exchange between systems. Local and metropolitan area networks, Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 12 June 2007
2. S Pulosoft, R Ramachandran, D Raz, Y Shavitt, S Prasun, Understanding TCP fairness over wireless LAN. *Proceedings of IEEE INFOCOM 2003*
3. M Seyedzadegan, M Othman, S Subramaniam, Z Zukarnain, The TCP fairness in WLAN: a review. *Proceedings of IEEE International Conference on Telecommunications, Malaysia*. 644–648 (2007)
4. M Seyedzadegan, M Othman, Weighted window for fair bandwidth allocation in WLANs. *European Journal of Scientific Research*. **29**, 481–490 (2009)
5. M Seyedzadegan, M Othman, Weighted window and class-based weighted window methods for per-station TCP fairness in IEEE 802.11 WLANs. *EURASIP J Wireless Commun Netw* **10** (2010). Article ID 593497, 2010
6. D-Y Kim, E-C Park, C-H Choi, Distributed access time control for per-station fairness in infrastructure WLANs. *IEICE Transactions on Communications*. **E89-B**, 2572–2579 (2006). doi:10.1093/ietcom/e89-b.9.2572
7. E-C Park, D-Y Kim, C-H Choi, Analysis of unfairness between TCP uplink and downlink flows in Wi-Fi hot spots. *Proceedings of IEEE Global Telecommunication Conference (Globecom '06)*. (2006)
8. E-C Park, D-Y Kim, H Kim, C-H Choi, A cross-layer approach for per-station fairness in TCP over WLANs. *IEEE Trans Mobile Comput*. **7**(7):898–911 (2008)
9. N Blefari-Melazzi, A Detti, I Habib, A Ordine, S Salsano, TCP fairness issues in IEEE 802.11 networks: problem analysis and solutions based on rate control. *IEEE Trans. Wireless Commun*. **6**, 1346–1355 (2007)
10. MA Brown, Traffic Control using tcng and HTB HOWTO. <http://tldp.org/HOWTO/Traffic-Control-tcng-HTB-HOWTO/> (2006)
11. MA Brown, Traffic Control Howto. <http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html> (2006)
12. A Trunganon, V Visoottiviset, Adaptive wireless bandwidth allocation for per-station fairness. *Proceedings of the 9th International Symposium on Communication and Information Technology 2009 (ISCIT 2009)*. (Incheon, Korea, 2009), pp. 1286–1291
13. OpenWRT Project. <http://openwrt.org/>
14. V Visoottiviset, A Trunganon, S Siwamogsatham, Adaptive bandwidth allocation for per-station fairness on wireless access router. *Proceedings of the 10th International Symposium on Communication and Information Technology 2010 (ISCIT 2010)*. (Tokyo, Japan, 2010), pp. 238–243
15. Y-C Lin, WK Lai, Adaptive bandwidth sharing mechanism for quality of service administration in infrastructure wireless networks. *Communications IET* **1**, 846–857 (2007). October 2007. doi:10.1049/iet-com:20060231
16. IEEE Std 802.11b-1999 (R2003), Supplement to IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan area networks-Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, (1999)
17. IEEE Std 802.11e™-2005, IEEE Standard for Information Technology- Telecommunications and information exchange between systems. Local and metropolitan area networks, Specific requirements-Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. (2005)
18. T Hiraguri, T Ichikawa, M Iizuka, S Kubota, Proposal of wireless traffic control schemes for wireless LANs. *IEICE Trans Commun*. **E91-B**(5):1340–1348 (2008). doi:10.1093/ietcom/e91-b.5.1340
19. S Floyd, TCP and explicit congestion notification. *ACM Comput. Commun. Rev.* **24**(5):10–23 (1994)
20. N Blefari-Melazzi, A Detti, A Ordine, S Salsano, A mechanism to enforce TCP fairness in 802.11 wireless LANs and its performance evaluation in a real test-bed. *IEEE World of Wireless Mobile and Multimedia Networks*. 1–7 (2007)
21. A Detti, N Blefari-Melazzi, I Habib, A Ordine, Per-station throughput fairness in a WLAN hot-spot with TCP traffic. *Elsevier Comp Netw*. **55**(8):1820–1833 (2011). doi:10.1016/j.comnet.2011.01.004
22. A Detti, N Blefari, I Habib, A Ordine, Per-station throughput fairness in a WLAN hot-spot with TCP traffic (Extended Version). http://netgroup.uniroma2.it/Andrea_Detti/VSB/STAFair_ext.pdf
23. BA Hirantha Sithira Abeysekera, T Matsuda, T Takine, Dynamic contention window control to achieve fairness between uplink and downlink flows in IEEE 802.11 WLANs. *Proceedings of IEEE Wireless Communications and Networking Conference 2007 (WCNC 2007)*. 2109–2114 (2007)
24. BA Hirantha Sithira Abeysekera, T Matsuda, T Takine, Dynamic contention window control mechanism to achieve fairness between uplink and downlink flows in IEEE 802.11 wireless LANs. *IEEE Trans Wireless Commun*. **7**(9):3517–3525 (2008)
25. ANSI/IEEE Std 802.11, 1999 Edition (R2003), Information technology- Telecommunications and information exchange between systems. Local and metropolitan area networks, Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2003)
26. Classless Qdisc. <http://www.bigwebmaster.com/General/Howtos/Traffic-Control-HOWTO/classless-qdiscs.html>
27. K Wagner, Short evaluation of linux's token-bucket-filter (TBF) queuing discipline. http://www.docum.org/docum.org/docs/other/tbf02_kw.ps (2001)
28. JL Valenzuela, A Monleon, I San Esteban, M Portoles, O Sallent, A hierarchical token bucket algorithm to enhance QoS in IEEE 802.11: proposal, implementation and evaluation. *Proceedings of Vehicular Technology Conference*. **4**, 2659–2662 (2004)
29. C Wang, T Tai, Achieving time-based fairness for VoIP applications in IEEE 802.11 WLAN using a cross-layer approach. *Proceedings of 2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. 1475–1480 (2010)
30. M Heusse, F Rousseau, G Berger-Sabbatel, A Duda, Performance anomaly of 802.11b. *Proceedings of IEEE INFOCOM*. **2**, 836–843 (2003)
31. R Jain, W Hawe, D Chiu, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. DEC-TR-301. <http://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf> (1984)
32. Wl Command. <http://dd-wrt.com>. http://www.dd-wrt.com/wiki/index.php/Wl_command
33. Wlanconfig Man Page, FreeBSD 6.2 Man Pages. <http://manpages.unixforum.co.uk/man-pages/unix/freebsd-6.2/8/wlconfig-man-page.html>
34. Linux Programmer's Manual, lwpriv. <http://www.unix.com/man-page/Linux/8/iwpriv/>

35. M Li, M Claypool, R Kinicki, WBest: a bandwidth estimation tool for IEEE 802.11 wireless networks. *Proceedings of 33rd IEEE Conference on Local Computer Networks (LCN)*. (Montreal, Quebec, Canada, 2008), pp. 374–381

doi:10.1186/1687-1499-2011-97

Cite this article as: Visoottiviset et al.: Cross-layer based adaptive wireless traffic control for per-flow and per-station fairness. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:97.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
