Wireless Netw
DOI 10.1007/s11276-006-6465-x

Temporal fairness guarantee in multi-rate wireless LANs for per-flow protection

Yongho Seok · Taekyoung Kwon · Yanghee Choi · Jean-Marie Bonnin

Published online: 8 May 2006

Abstract Wireless LAN technologies such as IEEE 802.11a and 802.11b support high bandwidth and multi-rate data transmission to match the channel condition (i.e., signal to noise ratio). While some wireless packet fair queuing algorithms to achieve the per-flow throughput fairness have been proposed, they are not appropriate for guaranteeing QoS in multi-rate wireless LAN environments. We propose a wireless packet scheduling algorithm that uses the multi-state (multi-rate) wireless channel model and performs packet scheduling by taking into account the channel usage time of each flow. The proposed algorithm aims at per-flow protection by providing equal channel usage time for each flow. To achieve the per-flow protection, we propose a temporally fair scheduling algorithm called Contention-Aware Temporally fair Scheduling (CATS) which provides equal channel usage time for each flow. Channel usage time is defined as the sum of the packet transmission time and the contention overhead time due to the CSMA/CA mechanism. The CATS algorithm provides per-flow protection in wireless LAN environments where the channel qualities of mobile stations are dynamic over time, and where the packet sizes are application-dependent. We also extend CATS to Decentralized-CATS (D-CATS) to provide per-flow protection in the uplink transmission. Using an NS-2 simulation, we evaluate the fairness property of both CATS and D-CATS in various scenarios. Simulation results show that the throughput of mobile stations with stable link conditions is not degraded by the mobility (or link instability) of other stations or by packet size variations. D-CATS also shows less delay and less delay jitter than FIFO. In addition, since D-CATS can coordinate the number of contending mobile stations, the overall throughput is not degraded as the number of mobile stations increases.

Keywords Temporal fairness · Per-flow protection · Packet scheduling · Wireless LAN · CSMA/CA

1. Introduction

Multi-rate wireless local area network (LAN) technologies such as IEEE 802.11a, 802.11b, and 802.11g, which support high bandwidth in wireless hotspot areas, are becoming widespread. These wireless LANs match their data transmission rates to the channel conditions. Higher speeds than the base data rate become possible when the signal-to-noise ratio (SNR) is sufficiently high. In IEEE 802.11a, the set of possible data rates is 6, 9, 12, 18, 24, 36, 48 and 54 Mbps; IEEE 802.11b allows transmission at 1, 2, 5.5 and 11 Mbps. Multi-rate wireless LANs can provide bandwidth up to tens of Mbits/s but they cannot guarantee a Quality of Service (QoS). Since new applications, such as voice over IP (VoIP) and video on demand (VoD) services, will soon be

This work was supported in part by the Brain Korea 21 project of Ministry of Education and in part by the National Research Laboratory project of Ministry of Science and Technology, 2004, Korea.

Y. Seok (✉) · T. Kwon · Y. Choi
School of Computer Science and Engineering, Seoul National University, San 56-1, Shillimdong, Gwanak-gu, Seoul, Korea
e-mail: yhseok@mmlab.snu.ac.kr

T. Kwon
e-mail: tk@mmlab.snu.ac.kr

Y. Choi
e-mail: yhchoi@mmlab.snu.ac.kr

J.-M. Bonnin
ENST Bretagne, 2, rue de la Chataigneraie – CS 17607,
35576 Cesson Sevigne Cedex – France
e-mail: jm.bonnin@enst-bretagne.fr

implemented over wireless LANs, the guarantee of QoS is a vital issue.

In this paper, we propose a packet scheduling mechanism capable of providing QoS "guarantees" in multi-rate wireless LAN environments. In particular, we attempt to solve two problems mentioned by previous authors [18, 19]. Berger-Sabbatel et al indicate that, when some mobile stations use a lower data rate than the others, the performance of all mobile stations is considerably degraded. This is became the basic CSMA/CA mechanism of wireless LANs guarantees an equal channel access probability to all mobile stations. When a mobile station with a low data rate captures the channel, it takes more time than a station with a higher data rate, and hence the utilization of the channel is degraded. The first requirement addressed in this paper is that, under stable radio-link conditions, the throughput of a mobile station should not depend on the data rate variations experienced by other mobile stations.

Garg and Kappes [19] present experimental studies on the throughput of IEEE 802.11b wireless networks for UDP and VoIP traffic. The average overhead per data frame using these wireless LANs is around 840 μs, regardless of the size of a data frame. This implies that the overhead per data frame becomes overwhelming as the packet size becomes smaller. For example, when a G.711 codec with 20 ms audio payload is used, an 802.11b cell can support only a limited number of simultaneous VoIP calls [20]. Since the VoIP application transmits a large number of small-sized packets, the contention overhead reduces the channel utilization significantly. This motivates the second requirement that we will address: the throughput of mobile stations should be independent of the packet size used by other applications in the same WLAN.

To solve these problems, we propose a per-flow protection mechanism called Contention-Aware Temporally fair Scheduling (CATS), which supports the following two characteristics in multi-rate wireless LAN environments:

- *Mobility independence* – The throughput of static (or stable) mobile stations is not affected by the mobility (or radio link instability) of other mobile stations. Since the quality and data rates of a channel vary as stations move about, mobility independence means that the throughput of static mobile stations is not influenced by data rate changes experienced by moving stations.
- *Application independence* – The throughput experienced by mobile stations using a constant packet size is not affected by the variation of the packet size used in other mobile stations.

Two recent proposals, Opportunistic Auto Rate (OAR) [14] and Wireless Credit-based Fair Queuing (WCFQ) [15], are related to our work. In OAR, it is suggested that different packet transmission policies could be used according to the data rate of the channel, with the aim of improving the overall throughput. This is achieved by allowing multiple back-to-back packet transmissions with a high data rate but the fairness of the policy is not considered. WCFQ provides a statistical fairness guarantee for a continuous channel model based on a general cost function, instead of the two-state discrete model adopted in previous works [6, 7, 9, 16]. The cost function is deduced from the channel condition and is the key element in achieving a statistical fairness bound and hence improving the throughput. However, it is difficult to guarantee fairness at the same time as mobility independence and application independence, since these multiple constraints make it difficult to formulate an accurate cost function.

In Section 2, we begin by presenting channel model for wireless packet networks and introduce the concept of *temporal fairness*. In Section 3, we define the temporal fluid flow model as a reference system, by extending the Generalized Processor Sharing (GPS) algorithm used in wired networks. Then we describe Temporally-Weighted Fair Queuing (T-WFQ), which is a preliminary version of the CATS algorithm. In Section 4, we describe CATS, and show how it provides temporal fairness and per-flow protection. In Section 5, we extend CATS to decentralized-CATS (D-CATS) for uplink traffic. In Section 6, we present simulation results which verify that the CATS algorithm provides fairness properties such as mobility independence and application independence.

2. Channel modeling of wireless packet network

2.1. Wireless network with a binary channel model

Previous authors [6, 7, 9, 16] have proposed the packet fair scheduling algorithm for wireless networks. They considered the link state dependent characteristics of wireless networks and introduced a channel model with just two states: good and bad. It can be represented as a two-state Markov chain, as shown in Fig. 1. When the channel state of a mobile station is bad, other mobile stations with good channel states will have a higher priority in packet scheduling. Lagging stations will be compensated when their channel states become good, in order to improve the overall fairness of the algorithm. This two-state channel model is not applicable to current wireless LANs such as 802.11a, 802.11b, and 802.11g because each mobile station can transmit packets at different rates by

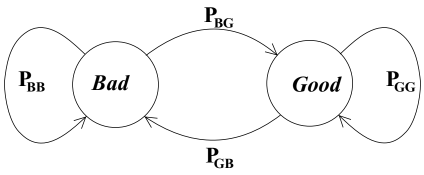


Fig. 1 A two-state Markov chain

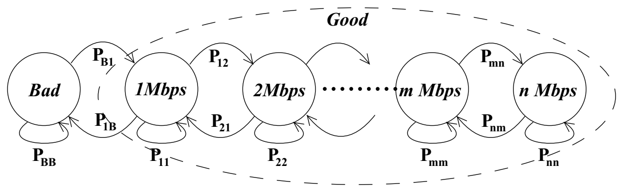


Fig. 2 An n-state Markov chain

adjusting either the transmission power or the modulation scheme depending on the state of the channel. For example, in a 802.11b wireless LAN, mobile stations support four data rates (i.e., 11 Mbps, 5.5 Mbps, 2 Mbps and 1 Mbps), which can be represented by a multi-state channel model [14, 15, 17], which we will now discuss.

2.2. Wireless network with multi-state channel model

Previous authors [14, 15] have proposed a wireless packet scheduling algorithm based on a continuous wireless channel model, which acknowledges the fact that a mobile station will experience different transmission costs depending on its current channel state. In wireless systems with variable rates (such as IEEE 802.11a, 802.11b and 802.11g), a mobile station with a high-quality channel can transmit packets at a higher data rate. The cost of transmissions in a system of this sort is proportional to the transmission time, since a mobile station with a low-quality channel will use the lower data rate and take longer. Figure 2 presents a channel model for the multi-rate wireless LANs which we will refer to as the n-state Markov chain.

3. Packet fair scheduling in multi-rate wireless LANs

Let us take consider IEEE 802.11b for purposes of discussion. It supports data rates of 11 Mbps, 5.5 Mbps, 2 Mbps or 1 Mbps depending on the wireless link condition of a mobile station [12, 13]. In [18], Berger-Sabbatel et al present performance anomalies in 802.11b wireless LANs. For example, a host transmitting at 1 Mbps dominates the shared-link occupation time, so that the throughput experienced by other hosts transmitting at higher bit rates (i.e., 5 or 11 Mbps) will be drastically reduced. Since the basic operation of IEEE 802.11b provides all mobile stations with the same channel access probability, the presence of a mobile station with a low data rate will reduce the overall throughput. Cantieni et al [17] also analyze the overall throughput achieved in IEEE 802.11b wireless LANs assigned a finite communication load. This analysis also shows that a mobile station with a low data rate degrades the overall system throughput of a wireless LAN.

To provide a defined QoS in a multi-rate wireless LAN, the throughput of every mobile station should be independent of other stations' transmission cost, which is defined as the channel usage time, which is in turn inversely proportional to the data rate. In previous work, per-flow protection has usually been provided by method of achieving max-min throughput fairness [5–9].

In multi-rate wireless LANs, the approach to max-min throughput fairness should be revised to accommodate: multi-rate packet transmission and contention based channel access. Because the transmission cost of an IEEE 802.11b wireless LAN are variable, temporal fairness should be taken into account in designing a packet scheduling algorithm. So, we will now define the temporal fairness model that we use in this paper.

3.1. Temporally fair fluid flow modeling

Generalized Processor Sharing(GPS) [5] is a technique, based on a fluid flow model, which guarantees that, during an arbitrary time window $[t_1, t_2]$, any two backlogged flows i and j are served in proportion to their weights if the unit of service time (δ) is infinitesimally small. This is represented as follows:

$$\frac{W_i(t_1, t_2)}{\phi_i} = \frac{W_j(t_1, t_2)}{\phi_j} \tag{1}$$

Here, $W_i(t_1, t_2)$ is the total service time received by flow i during the time window $[t_1, t_2]$ and ϕ_i is the weight of the flow i.

Since the capacity of each link changes and is different for each station, a monolithic GPS, that assumes a fixed link capacity, cannot provide per-flow protection. Therefore, in Fig. 3, we present a revised temporally fluid flow model that we call T-GPS. This model takes account of a different data rate ($C(i)$) for each mobile station. Here each flow i is served during ϕ_i time units. T-GPS provides the fairness we are looking for because each mobile station receives a length of service time proportional to its weight.

In T-GPS, the round number at time t, $R(t)$, is calculated using Eq. (2). It represents the normalized fair amount of service time that each flow should have received by time t.

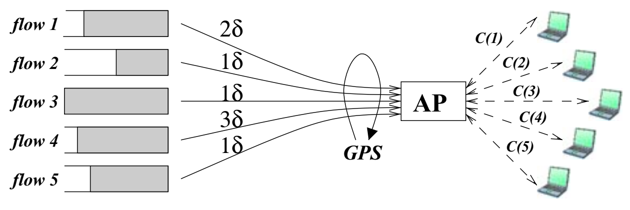


Fig. 3 Temporally fair fluid modeling

$$\frac{dR(t)}{dt} = \frac{1}{\sum_{i \in B(t)} \phi(i) \cdot \delta} \tag{2}$$

The variable δ is now the minimum service time of T-GPS and $B(t)$ is the set of backlogged flows.

As GPS is based on the fluid flow model, it cannot be applied to a real system. But a discrete version of GPS, packet-based weighted fair queuing, has been used in real systems. We extend weighted fair queuing (WFQ) in a similar way by considering multiple data rate in formulating the temporal weighted fair queuing (T-WFQ) algorithm. First, we will briefly summarize the WFQ algorithm, and then introduce the T-WFQ algorithm. The WFQ algorithm maintains a round number $R(t)$ and associates each flow i with a virtual start time $S(i, k)$ and a virtual finish time $F(i, k)$. $S(i, k)$ represents the normalized amount of service that flow i has received for serving the first $k - 1$ packets, and $F(i, k)$ represents the sum of between $S(i, k)$ and the normalized amount of service that flow i should receive in order to serve the k^{th} packet. The goal of the WFQ algorithm is to minimize the difference between $S(i, k)$ and $R(t)$. This is usually achieved by selecting the packet with smallest $S(i, k)$ or $F(i, k)$ to receive service first. Notice that the role of the round number $R(t)$ is to reset $S(i, k)$ whenever an unbacklogged flow i becomes backlogged again. More precisely,

$$S(i,k) = \begin{cases} \max\{R(t), S(i,k-1)\} & \\ & \text{flow } i \text{ become backlogged} \\ F(i,k-1) & (k-1)^{\text{th}} \text{ packet finish} \end{cases} \tag{3}$$

$$F(i,k) = S(i,k) + \frac{P(i,k)}{\phi(i)} \tag{4}$$

Here $P(i, k)$ is the packet size for the k^{th} packet of flow i, and $\phi(i)$ is its weight.

From Eq. (5), we obtain the virtual finish time for each flow in order to decide the T-WFQ scheduling flow order. The packet size is divided by the channel capacity that is offered to the flow (i.e., the data rate available for the station). The packet with the earliest virtual finish time (round number) is chosen for the next transmission.

$$F(i,k) = S(i,k) + \frac{P(i,k)}{\phi(i) \cdot \delta \cdot C(i)} \tag{5}$$

Here $C(i)$ is the channel capacity offered to flow i. In Eqs. (2) and (5), we can omit the δ value in both $R(t)$ and $F(i, k)$ altogether.

Figure 4 represents the worst scheduling scenario, in order to show the relative fairnessbound between two flows i and j. In this case, the first packet of flow i is served before the first packet of flow j, since the virtual finish time of the first packet of flow i is $R(t_1)$, which means that the round number computed by T-GPS at time t_1 is smaller than the virtual finish time of the first packet of flow j. With T-WFQ, the two flows i and j have a relative fairness bound, RFB(i, j), which is the upper limit of the difference between the channel usage times of the two flows. RFB(i, j) can be calculated from Eq. 6, and Lemma 1 shows the relative fairness bound provided by T-WFQ.

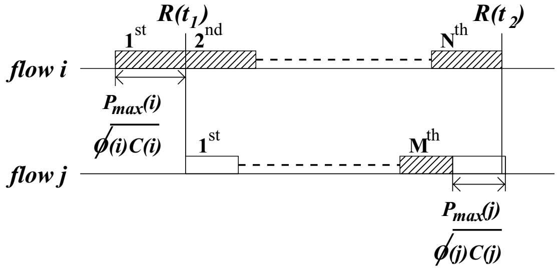


Fig. 4 Relative fairness bound of temporally weighted fair queuing

$$\left| \frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)} \right| \leq \text{RFB}(i,j) \tag{6}$$

Lemma 1. *For any two backlogged flows i and j, the difference in channel usage times in T-WFQ is bounded by the relative fairness bound,* RFB(i, j), $|\frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)}| \leq \frac{\text{P}_{\max}(i)}{\phi(i)\cdot C(i)} + \frac{\text{P}_{\max}(j)}{\phi(j)\cdot C(j)}$. *Here* $\text{P}_{\max}(i)$, $\text{P}_{\max}(j)$ *are the maximum packet sizes of the two flows, i and j respectively.*

Proof: Let $[t_1, t_2]$ be the time interval, as shown in Fig. 4. Flow i served the first packet with $R(t_1)$ finishing at time t_1, and the n^{th} packet with $R(t_2)$ finishing at time t_2. During the time necessary to serve the n packets of flow i, the round number of flow i increases by $R(t_2) - R(t_1) + \frac{\text{P}_{\max}(i)}{\phi(i)\cdot C(i)}$, and the round number of flow j increases by $R(t_2) - R(t_1) - \delta$. Since the maximum value of δ is $\frac{\text{P}_{\max}(j)}{\phi(j)\cdot C(j)}$, the difference in round number between flows i and j is bounded by $\frac{\text{P}_{\max}(i)}{\phi(i)\cdot C(i)} + \frac{\text{P}_{\max}(j)}{\phi(j)\cdot C(j)}$. □

3.2. Temporal fairness issues in multi-rates wireless LANs

Currently, the DCF mechanism in wireless LANs employs the carrier sense multiple access/collision avoidance (CSMA/CA) algorithm that incurs additional time (i.e. contention overhead) since it uses a back-off mechanism. To analyze the contention overhead in wireless LANs, we briefly review the Distributed Coordinated Function (DCF) from IEEE 802.11 [1].

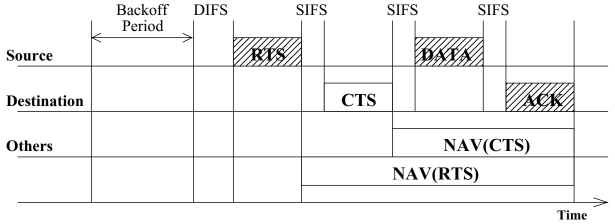


Fig. 5 Transmission timeline of DCF in IEEE 802.11

As described in the standard [1], a transmitting mobile station must first sense an idle channel for the duration of a specified period, the Distributed InterFrame Spacing (DIFS). This delay generates a random backoff time, chosen uniformly in the range $[0, w-1]$, where w is what is referred to as the contention window. At the first transmission attempt, w is set to the minimum contention window $\mathrm{CW}_{\min}$.

After the backoff timer reaches 0, the mobile station transmits a short request-to-send (RTS) message. If this message is received, the receiving mobile station responds with a short clear-to-send (CTS) message. Any other mobile stations that hear either the RTS or CTS message uses the duration field of this control message to update their network allocation vector (NAV), which contain information about the time during which the channel will remain busy. Thus, all mobile stations, including hidden nodes, can defer transmission and hence avoid collisions. Finally, a binary exponential backoff scheme is used, in which the value of w is set by evaluating $2^{n+5} - 1$ (where the retry counter $n = 0, \ldots, 5$), beginning with an initial value of 31, proceeding up to a maximum value 1023. Figure 5 shows the sequence in which packets are transmitted using DCF in IEEE 802.11.

We briefly calculate the contention overhead time based on the IEEE 802.11b system parameter (Table 1) to illustrate the effect of contention overhead in T-WFQ. A further detailed description of contention overhead is provided in Section 4, when collisions and wireless packet errors will be considered.

$$\text{Contention Overhead} = \begin{cases} \frac{\mathrm{CW}_{\min}}{2} + \mathrm{DIFS} + \mathrm{SIFS} + \frac{\mathrm{ACK+PHY\ header}}{\mathrm{Basic\ rate}} & \text{basic access mechanism} \\ \frac{\mathrm{CW}_{\min}}{2} + \mathrm{DIFS} + 3 \times \mathrm{SIFS} + \frac{\mathrm{RTS{+}CTS{+}ACK{+}PHY\ header}}{\mathrm{Basic\ rate}} & \text{RTS/CTS mechanism} \end{cases} \tag{7}$$

For now, we will assume that no collision with another data packet occurs and that the backoff time is chosen from a uniform distribution.

From Eq. (7), we calculate contention overheads: 866 μs for the basic access mechanism and 1542 μs for the RTS/CTS mechanism using Table 1. These contention times correspond to the transmission time of a 1190 bytes frame and a 2120 bytes frame at 11 Mbps.

Table 1 IEEE 802.11b system parameters

PHY header	192 bits
RTS frame	160 bits + PHY header
CTS frame	112 bits + PHY header
ACK frame	112 bits + PHY header
Basic rate	1 Mbps
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
$\mathrm{CW}_{\min}$	31

If we ignore this contention overhead, per-flow protection is not achieved and the relative fairness bound of T-WFQ is violated. By extending Lemma 1, we derive the following lemma (its proof is given in Appendix A). It represents the difference in channel usage times between two flows i and j in a CSMA/CA network.

Lemma 2. *When using T-WFQ in a CSMCA/CA wireless network, the difference in channel usage time between any two backlogged flows i and j is not bounded. As the backlogged flows are continuously served, the difference in channel usage time is continuously increased by*

$$\left| \frac{\mathrm{CO}(t)}{\phi(i)} \left(1 - \frac{C(j) \cdot P_{\mathrm{avg}}(i)}{C(i) \cdot P_{\mathrm{avg}}(j)} \right) \right| .$$

Here $\mathrm{CO}(t)$, $\mathrm{P}_{\mathrm{avg}}(i)$ *and* $\mathrm{P}_{\mathrm{avg}}(j)$ *are the contention overhead at time t, and the average packet sizes of flow i and j respectively.*

Lemma 2 shows that the channel usage time of a flow also depends on two variables which are determined by other flows. The first of these is the data rate, which is affected by the movement of mobile stations. The second is the average packet size, which depends on application characteristics. For example, let us suppose that the weights of two flows i and j are both 1, that the average packet sizes of these two flows, $\mathrm{P}_{\mathrm{avg}}(i)$ and $\mathrm{P}_{\mathrm{avg}}(j)$, are 1024 and 512 bytes respectively, and that the data rates, $C(i)$ and $C(j)$, are 5.5 Mbps and 11 Mbps respectively. The ratio of actual packet transmission time of the two flows, $\frac{\mathrm{P}_{\mathrm{avg}}(i)}{C(i)} / \frac{\mathrm{P}_{\mathrm{avg}}(j)}{C(j)}$, is equal to 4. T-WFQ will schedule 4 packets for flow j and 1 packet for flow i. The difference between channel usage times for the two flows is 2598 μs per packet of flow i, when the basic access mechanism is used. Flow j uses much more contention overhead time than flow i for transmitting 3 more packets.

Since the contention overhead in 802.11b is so high, it is very important to consider contention time in any packet scheduling algorithm. Therefore, we propose a contention-aware temporally fair scheduling algorithm called CATS. We will describe it in the next section.

4. The contention-aware temporally fair scheduling (CATS) algorithm

4.1. CATS algorithm description

CATS is based on the T-WFQ algorithm described in Section 3. The basic idea of CATS is to utilize contention overhead time that occur during the computation of the virtual finish time of a packet. The virtual finish time of the k^{th} packet of a flow i is given by the following extension of Eq. (5):

$$F(i,k) = S(i,k) + \frac{P(i,k)}{\phi(i)\cdot C(i)} + \frac{\mathrm{CO}(t)'}{\phi(i)} \tag{8}$$

Here $\mathrm{CO}(t)'$ is the exponentially weighted moving average of the contention overhead, $\mathrm{CO}(t)$, which is due to the CSMA/CA mechanism.

To calculate the contention overhead, we classify each time-slot into three states, *successful access*, *collision* and *backoff*, which correspond to the status of the wireless channel. The state *successful access* means that each mobile station successfully accessed the shared wireless medium to transmit the data packet. Time-slots within the *successful access* state are further classified into two sub-states: *successful transmission* and *packet corruption*, according to whether the packet is successfully transmitted or not. Since the number of time-slots required by the *successful transmission* and *packet corruption* show the same characteristics, the requirement of both states can be expressed by the following equation:

$$T_{\text{successful\_access}} = \begin{cases} \text{DIFS} + \text{SIFS} + \frac{\text{PHY Header}+\text{ACK}}{\text{Basic rate}} + \frac{\text{DATA}}{\text{Data rate}} & \text{basic access mechansim} \\ \text{DIFS} + 3\times\text{SIFS} + \frac{\text{RTS}+\text{CTS}+\text{PHY Header}+\text{ACK}}{\text{Basic rate}} + \frac{\text{DATA}}{\text{Data rate}} & \text{RTS/CTS mechanism} \end{cases} \tag{9}$$

The second, *collision* state occurs when at least two mobile stations attempt to transmit a packet simultaneously. In this case, the number of time-slots required is:

$$T_{\text{collision}} = \begin{cases} \text{DIFS} + \text{SIFS} + \frac{\text{PHY Header}}{\text{Basic rate}} + \frac{\text{DATA}}{\text{Data rate}}^{*} & \text{basic access mechansim} \\ \text{DIFS} + \frac{\text{RTS}}{\text{Basic rate}} & \text{RTS/CTS mechanism} \end{cases} \tag{10}$$

In, multi-rate wireless LANs, each mobile station has a different data-rate, depending on the channel condition, so the duration of a *collision* state in the basic access mechanism depends on the longest transmission time ($\frac{\text{DATA}}{\text{Data rate}}^{*}$) of the mobile stations involved in the collision.

The third, *backoff* state lasts for as long as mobile stations are decreasing their backoff counters. We use the variable T_{backoff} to denote the number of time-slots occurring in the *backoff* state between two consecutive *successful access* states.

We can now define the contention overhead as Eq. (11): the interval between two consecutive *successful access* states minus the transmission time for a data frame.

$$\mathrm{CO}(t) = T_{\text{backoff}} + \sum_{k=1}^{N_{\text{collision}}} T_{\text{collision}}(k) + T_{\text{successful\_access}} - \frac{\text{DATA}}{\text{Data rate}} \tag{11}$$

Here, $N_{\text{collision}}$ is the number of collisions that occurs between two consecutive successful access states.

The contention overhead $\mathrm{CO}(t)$ is adjusted when the number of active mobile stations competing for access to the shared wireless medium changes, because the number of active mobile stations determines both $N_{\text{collision}}$ and T_{Backoff}. A large variation of $\mathrm{CO}(t)$ has a negative effects on the scheduling policy since the packet priority is influenced by $\mathrm{CO}(t)$ in Eq. (5). If we can adjust the number of active mobile stations, we can reduce the variation of $\mathrm{CO}(t)$ and hence improve the total system throughput by reducing the number of collisions. In Section 5, we propose a mechanism to control the number of active mobile stations.

In order to compute the contention overhead in 802.11 wireless LANs, each mobile station updates two variables, *lastAck* and *currentACK*, whenever an ACK frame is transmitted or received: *currentACK* is either the arrival time of the ACK frame that is currently being received, or the departure time of the ACK frame that is currently being transmitted; and *lastACK* is either the arrival time of the most recently received ACK frame, or the departure time of the most recently transmitted ACK frame.

The contention overhead $\mathrm{CO}(t)$ of each mobile station is computed as follows:

$$\mathrm{CO}(t) = \mathit{currentACK} - \mathit{lastACK} - \text{PLCP LENGTH field} \tag{12}$$

Since an ACK frame is transmitted only if the data frame has been correctly received, the interval between two consecutive ACK frames is equivalent to the interval between two consecutive successful accesses. Each mobile station computes the contention overhead time whenever it has to send a data frame. The transmission time for a data frame is obtained from the physical layer convergence protocol (PLCP) sublayer. Length field in the PLCP frame indicates the number of microseconds required to transmit the MAC protocol data unit (MPDU).

The contention overhead given by Eq. (12) does not handle all cases. When we defined the contention overhead in Eq. (11), we included corrupted packets in the *successful access* state. But, in IEEE 802.11 wireless LANs, it is difficult to differentiate packet corruption that occurs because of a channel error and corruption resulting from a collision. To solve this problem, the RTS/CTS mechanism should be used instead of the basic access mechanism. When the RTS/CTS mechanism is used, each mobile station can estimate the completion time of a *successful access* using the duration field of CTS frame, regardless of any packet corruption. In this case the contention overhead is computed as;

$$\begin{aligned}\mathrm{CO}(t) = {} & (\mathit{currentCTS} + \mathit{currentNAV}) \\ & - (\mathit{lastCTS} + \mathit{lastNAV}) \\ & - \text{PLCP LENGTH field} \qquad (13)\end{aligned}$$

In Fig. 6, we can see an example of contention overheads, CO($t-1$) and CO(t), due to access-point (AP) transmissions.

The pseudo-code for the CATS algorithm is provided as Algorithm 1 and Algorithm 2, using the notation given in Table 2. Algorithm 1 explains the operations performed when a new packet arrives in flow i. If the target flow is not currently backlogged, the virtual start time of that flow is computed as the latest of the current round number and the virtual finish time of the previous packet (line 3 in Algorithm 1). Next, the virtual finish time of this flow is computed using Eq. (5). However, when the flow corresponding to an incoming packet is already backlogged, the packet is simply added to the queue.

Algorithm 2 describes the operations performed in transmitting a packet and shows how the round number, virtual start time and virtual finish time are updated. First, we find the flow that has the smallest virtual finish time among all eligible flows (line 1 in Algorithm 2). A flow is eligible if it is backlogged and its virtual start time is less than the round number. This eligibility constraint allows us to support a better quality of service [8], [9]. Assuming that the selected flow is continuously backlogged, we now dequeue the packet that is at the head of queue and update the virtual start time and virtual finish time of the flow. Finally, the round number is updated (line 8 in Algorithm 2).

Algorithm 1 *Enqueue*(i, P)
i : flow id, P : newly arriving packet, Q_i : queue for flow i

```
1: if  $Q_i = \emptyset$  then
2:     $Q_i \leftarrow P$;
3:     $s_i \leftarrow max(f_i, R(t))$;
4:     $f_i \leftarrow s_i + L/(C(i) \times \phi_i) + CO(t)/\phi_i$;
5:     $R(t) \leftarrow \max(\min_{k \in B(t)} s_k, R(t))$;
6: end if
7: insert (P,$Q_i$);
8: return;
```

Algorithm 2 *Dequeue*()

```
1: $i \leftarrow \min_{k \in B(t)\ and\ s_k \le R(t)} f_k$;
2: delete from(P,$Q_i$);
3: nextPacket $\leftarrow Q_i$;
4: if  $nextPacket \neq \emptyset$  then
5:     $s_i \leftarrow max(f_i, R(t))$;
6:     $f_i \leftarrow s_i + L/(C(i) \times \phi_i) + CO(t)/\phi_i$;
7: end if
8: $R(t) \leftarrow \max(\min_{k \in B(t)} s_k, R(t)$
         $+L/\sum_{k \in B(t)}(C(k) \times \phi_k) + CO(t)/\sum_{k \in B(t)} \phi_k)$;
9: transmit P;
```

4.2. Analysis of temporal fairness in CATS

Figure 7 shows the examples of the order in which packets are scheduled, and the throughputs of T-WFQ and CATS.

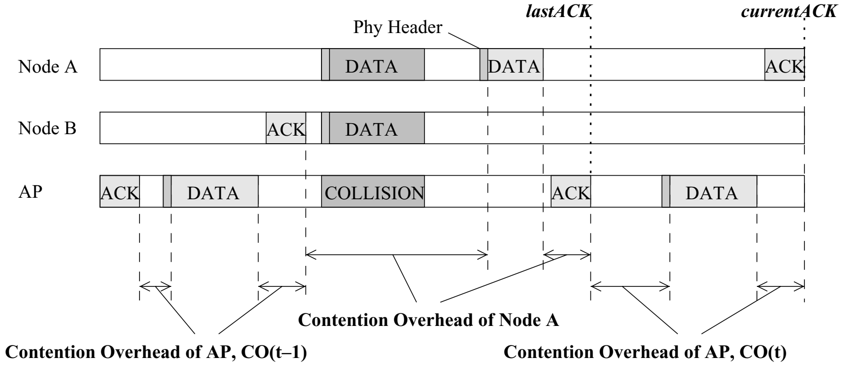

lastACK
currentACK
Phy Header
Node A
DATA
DATA
ACK
Node B
ACK
DATA
AP
ACK
DATA
COLLISION
ACK
DATA
Contention Overhead of Node A
Contention Overhead of AP, CO(t–1)
Contention Overhead of AP, CO(t)

Fig. 6 Contention overhead in the CSMA/CA MAC protocol

Table 2 Notation for the algorithm description

Notation	Description
$R(t)$	the round number at time t
s_i	the virtual start time for the flow i
f_i	the virtual finish time for the flow i
L	the length of the arrived and transmitted packet
ϕ_i	service weight for flow i
$C(i)$	channel data rate of flow i
$B(t)$	the set of backlogged flows at time t

Here, the contention overhead time of flows is 4 sec, the capacity of the link is 1 byte/sec (for illustration purposes) and the packet size of flow j is set to 8 bytes. When there are two flows i and j, we see how the packet scheduling order of two flows changes as the packet size of flow i decreases from 8 bytes to 2 bytes. Using T-WFQ, the corresponding throughput of flow j also decreases from 24/72 to 16/72. As we can see from the order in which packets are scheduled using T-WFQ, the number of packets served by flow i increases, and the corresponding increase in contention overhead for flow i penalizes the throughput of flow j. However, when we use CATS, the throughput of flow j is fixed at 24/72, regardless of the packet size for flow i. When the packet size of flow i is decreased, only the throughput of flow i is affected because the scheduling decision of CATS takes into account the total time consumed, which includes the contention overhead.

Figure 8 illustrates the worst-case scheduling scenario when using CATS. When flow i and j are continuously backlogged, the 1st packet of flow i is served at time t_1, since its virtual finish time is less than flow j's virtual finish time. CATS takes into account the contention overhead time in order to schedule the packet transmission and achieves temporal fairness with the bound given by the following lemma:

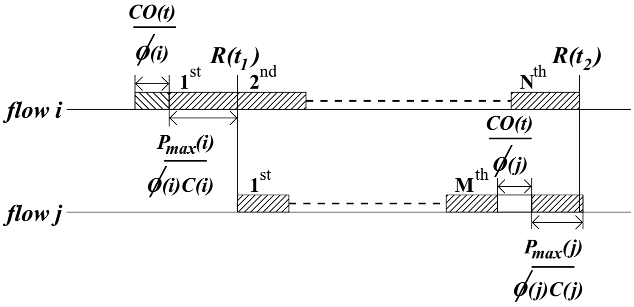


Fig. 8 Relative fairness bound of CATS

Lemma 3. *In a wireless network based on the CSMA/CA MAC protocol, for any two backlogged flows i and j, the difference in channel usage time using CATS is bounded by* $\frac{\mathrm{P}_{\max}(i)}{\phi(i)\cdot C(i)} + \frac{\mathrm{CO}(t)}{\phi(i)} + \frac{\mathrm{P}_{\max}(j)}{\phi(j)\cdot C(j)} + \frac{\mathrm{CO}(t)}{\phi(j)}$, *where* $\mathrm{P}_{\max}(i)$ *and* $\mathrm{P}_{\max}(j)$ *are the maximum packet sizes of flow i and j respectively.*

During the time interval $[t_1, t_2]$, a flow i is guaranteed to get channel usage time as follows:

$$W_i(t_1, t_2) = \frac{\phi(i)}{\sum_{k\in B(t)\phi(k)}}(t_2 - t_1) \pm \phi(i)\cdot\delta \tag{14}$$

The channel usage time of flow i depends on the weight of backlogged flows. It means that the throughput of flow i is also guaranteed and protected from the mobility of other mobile stations and from packet size variations. We can derive another lemma from the relative fairness bound given by Lemma 3.

Lemma 4. *During time interval* $[t_1, t_2]$, *when all flows are backlogged, the channel usage time of flow i is* $W_i(t_1, t_2) = \frac{\phi(i)}{\sum_{k\in B(t)\phi(k)}}(t_2 - t_1) \pm \phi(i)\cdot\delta$, *where* δ *is* $|\frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)}| \le$ RFB$(i, j) \le \delta, \forall(i, j) \in$ B(t).

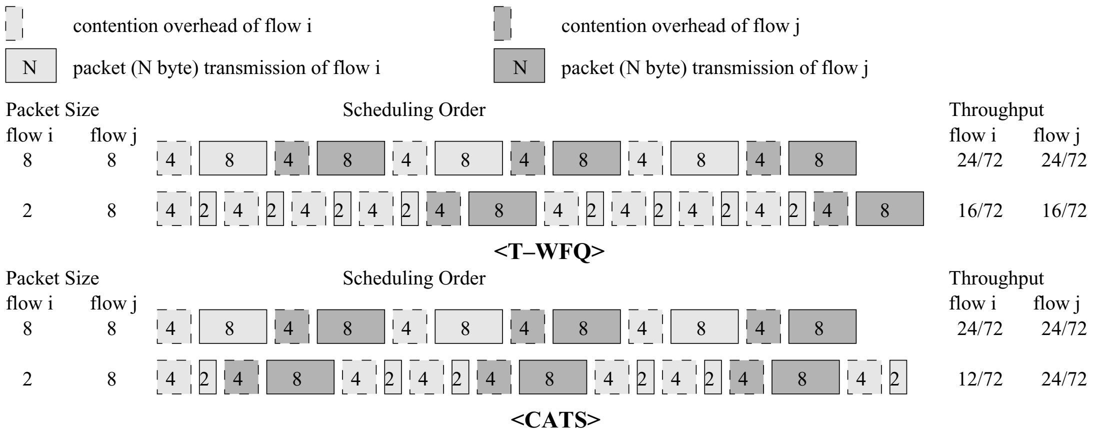


Fig. 7 Per-flow protection comparison between T-WFQ and CATS

5. Decentralized contention-aware temporally fair scheduling (D-CATS) algorithm

5.1. Description of the basic D-CATS algorithm

In wireless LANs using IEEE 802.11 DCF, per-flow protection can be provided for the downlink traffic by using CATS at the access point (AP). But it is not possible to guarantee temporal fairness for the uplink traffic because of the random access characteristics of the 802.11 DCF. To achieve temporal fairness in the DCF mode, CATS has to involve the mobile stations in the scheduling process and therefore must operate in a decentralized manner.

In this section, we propose a decentralized version of our contention-aware temporally fair scheduling algorithm (D-CATS) in order to achieve per-flow protection in IEEE 802.11 DCF mode. In D-CATS, each mobile station records the arrival time of all ACK frames transmitted from other mobile stations. Since D-CATS only deals with uplink DATA frame transmissions, all mobile stations can overhear the corresponding downlink ACK frames sent by the AP. Whenever an ACK frame is overheard, D-CATS updates the channel usage time of the mobile station that has successfully transmitted a packet. Meanwhile, the AP records the departure time of ACK frames, instead of the arrival time, and updates the channel usage time of each mobile station as ACK frames are transmitted. In this way, D-CATS knows the address of the mobile stations that have issued the DATA frame from the destination address in the ACK frame.

Each station now decides individually which are the next eligible mobile stations. Eligible stations will be allowed to compete to gain access to the uplink channel. In D-CATS, *R* stations with a lower channel usage time than other stations become eligible (line 7 in Algorithm 3). D-CATS allows *R* stations to compete to transmit their packets, in order to increase the resource utilization. Unfortunately this also reduces slightly the fairness achieved by the scheduling algorithm.

In this paper, *R* is set to 8 stations. In order to choose an appropriate value of *R*, we referred to the throughput analysis of 802.11 DCF wireless LANs by Bianchi [10]. From Eq. (15), we compute the number of stations (n) which will achieve the maximum throughput, using numerical techniques.

$$\begin{cases} \tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \\ p = 1-(1-\tau)^{n-1} \\ \tau = \frac{\sqrt{[n+2(n-1)(T_c^*-1)]/n-1}}{(n-1)(T_c^*-1)} \approx \frac{1}{n\sqrt{T_c^*/2}} \end{cases} \tag{15}$$

Here, τ is the probability that a station transmits in a randomly chosen time-slot and p is referred to as the conditional collision probability. This means that a packet being transmitted on the channel has a probability p of experiencing a collision. For more details about these equations, please refer to Eqs. (7), (9), (28) in Bianchi [10].

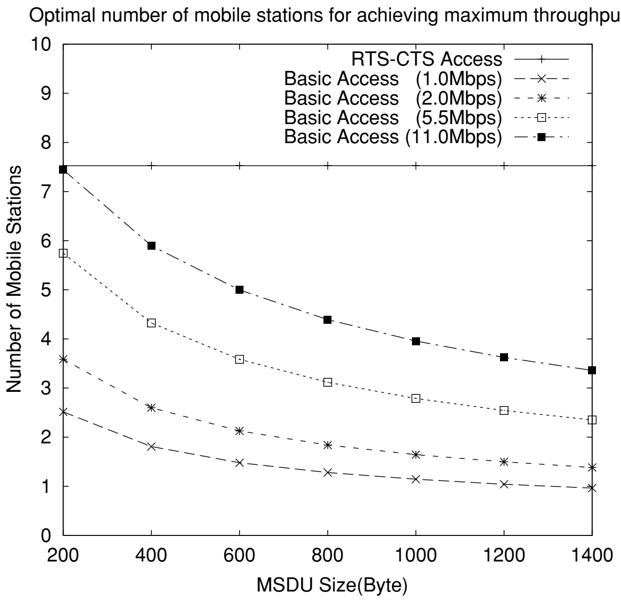


Fig. 9 The optimal number of stations to obtain the maximum achievable throughput in 802.11 wireless LANs

Using Eq. (15), Fig. 9 illustrates the optimal number of stations as the size of transmitted frame (the size of MAC service data unit) changes. There are five cases and two access mechanisms. For the purpose of this analysis, all mobile stations using the basic access mechanism have the same data rate in each case. When using the basic access mechanism in multi-rate wireless LANs, the collision time is determined by the longest transmission time of all mobile stations involved in a collision, as shown in Eq. (10); and we also see from Fig. 9 that the optimal number of stations varies depending on the data rate and the packet size. When the data rate increases and the packet size decreases, the optimal number of stations becomes larger because the packet transmission time is smaller and the wasted time due to the packet collision is consequently reduced. However, when the RTS-CTS access mechanism is used, the optimal number of stations is fixed to 7.526. In Eq. (10), the collision time using the RTS-CTS access mechanism does not depend on the data rate or on the size of the data frame, since collisions can only occur during RTS-CTS handshaking when using a fixed frame size and the basic rate. From Fig. 9, we can conclude that *R* should be adaptively determined to a value between 1 and 8, depending on the data rate of mobile stations, the average size of transmitted frames and the type of channel access mechanism. In the following discussion, we will set *R* to 8 stations since the RTS-CTS access mechanism is used in performance evaluations.

To compensate for the loss of short-term fairness, we put an additional constraint on the eligiblity of stations: each mobile station must require that the difference between the minimum channel usage time and its own channel usage time is less than $\delta/dataRate_{\text{index}}$. In Algorithm 3, we describe the pseudo-code of D-CATS. If the *leadingFlag* of a mobile station is set, it does not compete for packet transmission. However, when a mobile station frequently switches between the backlogged and unbacklogged states, some changes in the set of eligible stations could occur. In other words, it is possible that a mobile station which has previously sent a data frame when it was eligible is no longer eligible. If this station is currently performing the backoff process, that process should be stopped because the station no longer belongs to the set of eligible stations. An advantage of the D-CATS algorithm is that it can operate without any change in the 802.11 MAC protocol. Unfortunately, to stop the backoff process we would have to modify the 802.11 MAC, which is why we choose to ignore this priority inversion. But, it is not a real problem since D-CATS already allows several (R) stations compete to transmit their packets.

Algorithm 3 *receivedACK*(*dst*, *currentACK*)
dst : receiver address field in the received ACK frame
currentACK : arrival time of currently received ACK frame

```
$usageTime \leftarrow (currentACK - lastACK)/slotTime$;
$lastACK \leftarrow currentACK$;
$f_{dst} \leftarrow s_{dst} + usageTime/\phi_{dst}$;
$s_{dst} \leftarrow \max(f_{dst}, R(t))$;
$R(t) \leftarrow \max(\min_{k \in B(t)} s_k, R(t) + usageTime)$;
$count \leftarrow |k| : f_{k \in B(t)} \leq f_{index}$;
if $count \leq R$ and $s_{index} \leq (R(t) + \delta/dataRate_{index})$ then
    $leadingFlag \leftarrow 0$;
else
    $leadingFlag \leftarrow 1$;
end if
return;
```

5.2. Extended D-CATS+: an algorithm for error-prone wireless LANs

The D-CATS algorithm has some problem in error-prone wireless LANs. This is caused by the specific packet error characteristics of the wireless channel. In wireless networks, each mobile station has location-dependent and time-varying packet error characteristics. To reduce the packet error rate, several modulation schemes and error control schemes have been proposed. For example, in 802.11a, 802.11b, 802.11g, rate selection algorithms such as ARF [12] and RBAR [13] choose the adaptive data rate on the basis of the instantaneous link quality; while in 802.11e draft [4], the forward error correction (FEC) scheme is being considered.

When mobile stations experience a high packet error rate, D-CATS can guarantee neither per-flow protection nor temporal fairness. If the packet error rate of a mobile station increases, the contention overhead also increases, as shown in Fig. 10. Because error-free mobile stations have the same contention overhead, their throughput will also decrease as the packet error rate of other mobile stations increases.

Therefore, we put forward D-CATS+, which is an alternative solution using the optional RTS-CTS frame exchange proposed in the 802.11 DCF. In Algorithm 4, we present the pseudo code of D-CATS+. When a CTS frame is received, the *receivedCTS* function is called. It updates the channel usage time based on the duration field given in the CTS frame (lines 1 and 2 in Algorithm 4).

By means of the RTS-CTS frame exchange, we know which mobile station has the chance to transmit a data frame, and then we can update its channel usage time by adding in the arrival time of the CTS frame and the duration found in the CTS frame. By updating the channel usage time in this way, it is possible to maintain the channel usage time and the contention overhead time more exactly. As a consequence, the contention overhead of each mobile station becomes independent of the packet error rate of other mobile stations. In Fig. 11, we can see that the *lastACK* corresponding to the corrupted packet can be computed precisely using the *Duration* field found in the previously received CTS frame.

We can now compute the contention overhead time, CO(t), when 8 mobile stations use D-CATS+ (i.e., n is 8).

$$\begin{cases} P_{tr} = 1 - (1-\tau)^n = 0.2840,\ P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = 0.8601 \\[2ex] T_{\text{backoff}} = \lim_{k\to\infty} \sum_{i=1}^{k} i \cdot \sigma \cdot (1 - P_{tr})^i \\ \qquad\quad = \frac{\sigma\cdot(1-P_{tr})}{P_{tr}^2} = 177.5\mu s \\[2ex] T_{\text{backoff}} + N_{\text{collision}} T_{\text{collision}} \\ \quad = T_{\text{backoff}} + \lim_{k\to\infty} \sum_{i=1}^{k} i (T_{\text{backoff}} + T_{\text{collision}}) \\ \qquad \times (P_{tr} \cdot (1 - P_s))^i \\ \quad = T_{\text{backoff}} + \frac{(T_{\text{backoff}} + T_{\text{collision}}) \cdot P_{tr} \cdot (1-P_s)}{(1 - P_{tr}\cdot(1-P_s))^2} \\ \quad = 202.5\mu s \\[2ex] \text{CO}(t) = T_{\text{backoff}} + N_{\text{collision}} T_{\text{collision}} \\ \qquad\quad + T_{\text{successful\_access}} - \frac{\text{Data}}{\text{DataRate}} \\ \qquad\quad = 1434.5\mu s \end{cases} \tag{16}$$

Since D-CATS+ is based on the RTS-CTS access mechanism, we have obtained a value of 0.0409 for τ from Eq. (15). P_{tr} is the probability that there is at least one transmission in a randomly chosen time slot; it is the complement of the probability that there is no transmission in a randomly

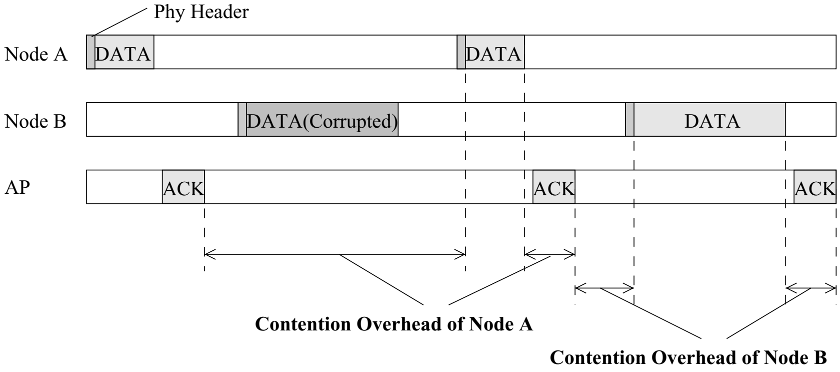


Fig. 10 Contention overhead in an error-prone wireless LAN

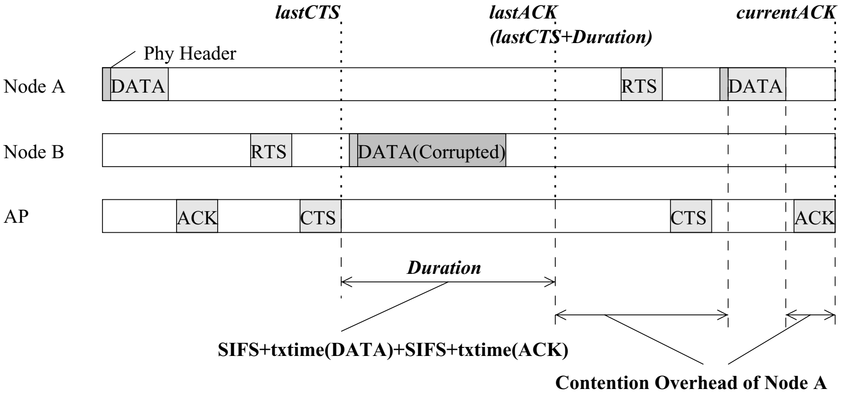


Fig. 11 Contention overhead using the RTS-CTS access mechanism in an error-prone wireless LAN

chosen time slot. P_s is the probability that a transmission on the channel is successful; it is given by the probability that exactly one station transmits on the channel, given that at least one station transmits.

Using D-CATS+, the contention overhead time is approximately 1434.5 μs as obtained in Eq. (16). This is independent on the packet error rate observed in the wireless channel. In Fig. 25 (Section 6.3), we also present the contention overhead times of both D-CATS and D-CATS+ as a function of the packet error rate. These results were obtained using a simulation tool, as explained in the next section. Both results show that D-CATS+ guarantees per-flow protection in error-prone wireless LANs because the contention overhead time does not depend on the packet error rate. On the other hand, the process that decides which are the next eligible mobile stations is the same as in D-CATS (lines 3–12 in Algorithm 3).

When a mobile station is associated with an AP for the first time, it synchronizes the channel usage time of all mobile stations using the control information received from the AP. The AP periodically broadcasts the channel usage table of all mobile stations in order to initialize the channel usage table of newly associated mobile stations. It also re-synchronizes the invalid channel usage tables of mobile stations that have experienced ACK frame losses or CTS frame losses, because of hidden terminal problems or the bad channel condition.

Algorithm 4 *ReceivedCTS*(dst, $duration$, $currentCTS$)
dst : receiver address field in the received CTS frame
$duration$: duration field in the received CTS frame
$currentCTS$: arrival time of currently received CTS frame

```
$usageTime \leftarrow$
  $(currentCTS + duration - lastACK)/slotTime$;
$lastACK \leftarrow currentCTS + duration$;
$f_{dst} \leftarrow s_{dst} + usageTime/\phi_{dst}$;
$s_{dst} \leftarrow \max(f_{dst}, R(t))$;
$R(t) \leftarrow \max(\min_{k \in B(t)} s_k, R(t) + usageTime)$;
$count \leftarrow |k| : f_{k \in B(t)} \leq f_{index}$;
if $count \leq R$ and $s_{index} \leq (R(t) + \delta/dataRate_{index})$ then
    $leadingFlag \leftarrow 0$;
else
    $leadingFlag \leftarrow 1$;
end if
return;
```

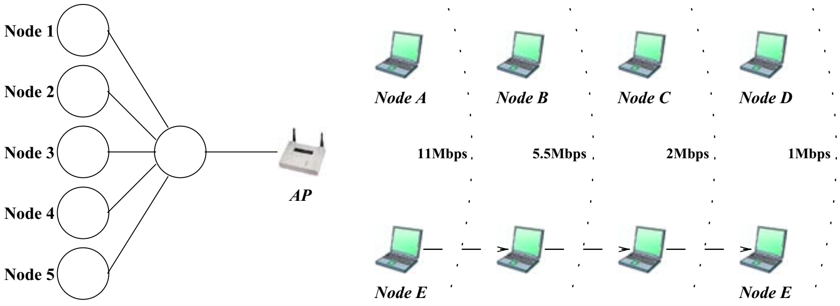
To implement D-CATS and D-CATS+, it is necessary to define a mechanism to announce the backlogged states of mobile station whenever they change. We can consider two mechanisms: (a) using a special control packet to announce the backlogged states and (b) utilizing the 802.11e standard, as follows. Figure 12 depicts the general MAC frame format defined in 802.11e and details the QoS field. The sub-fields

Fig. 12 The MAC frame format of IEEE 802.11e

octets:2	2	6	6	6	2	6	2	n	4
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	Feame Body	FCS

Bits 0–3	Bit 4	Bits 5–6	Bit 7	Bits 8–15
TID	FEC	ACK Policy	reserved (0)	Queue size unit of 256 octets

Fig. 13 Basic simulation scenario



of the QoS field are used for different purposes in various scenarios. We just describe how the QoS field is used when QoS data (non-null) frames are sent by stations. The queue size is filed as a 8-bit field that indicates the amount of buffered traffic for a given traffic identifier (TID) at the station sending this frame. A queue size value of 0 is used solely to indicate the absence of any buffered traffic in the queue used for the specified TID. Although this queue size field is not designed for the extended DCF mode, it can nevertheless be used to announce the backlogged states at mobile stations.

6. Performance evaluation

We evaluate the performance of the CATS algorithm, together with that of the other two scheduling algorithms, FIFO and T-WFQ. We show that the CATS algorithm supports per-flow protection by demonstrating that the throughput of a flow remains unchanged even when there is movement (link instability) of other mobile stations or changes in the packet sizes used by various applications. We will also define a fairness index [11] that represents the relative fairness achieved for each flow. It is calculated by measuring the amount of time that each flow has used for its transmissions.

$$\text{fairness index} = \frac{\left(\sum_{i\in B(t)} \frac{W_i(t_1,t_2)}{\phi_i}\right)^2}{\text{number of flows} \times \sum_{i\in B(t)} \left(\frac{W_i(t_1,t_2)}{\phi_i}\right)^2} \tag{17}$$

Our simulations were run on the NS-2 (ns-allinone-2.26) simulator [23], with the simulation topology given in Fig. 13. Each mobile station, including the AP, has a protocol stack, as illustrated in Fig. 14. In this protocol stack, the routing protocol uses the NO Ad-Hoc Routing Agent (NOAH) and the interface queue is located between the link layer (LL) and the Medium Access Control (MAC) protocol. The MAC protocol that we used conforms to the IEEE 802.11b specification, and we use the Ricean model [24] of propagation. When there is a dominant stationary signal component present, such as a line-of-sight propagation path, the small-scale fading envelope has a Ricean distribution. This is often described in terms of a parameter k, which is defined as the ratio between the deterministic signal power and the variance of multi-path fading. If k is equal to 0, the Ricean distribution reduces to the Rayleigh distribution, in which the signal only is transmitted by reflection. For a large value of k, the received signal strength is stable since the variation in the multi-path fading is very weak. In this paper, we choose a large value of k (256) because we want to assume that the channel condition of fixed mobile stations is stable, in order to demonstrate

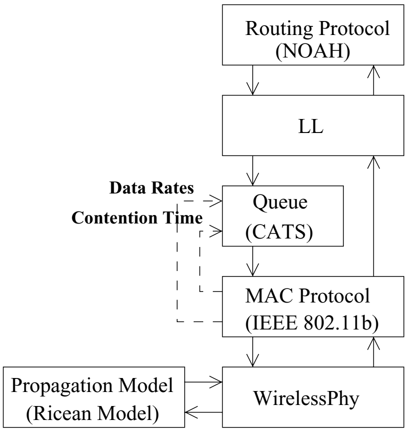


Fig. 14 Protocol stack

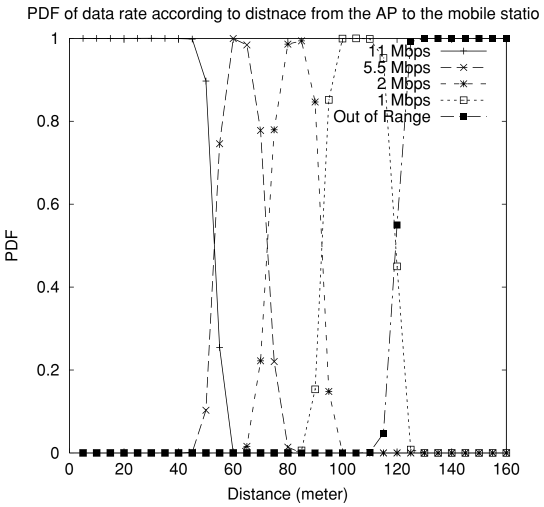


Fig. 15 PDF of data rate against distance from the AP to a mobile station

the per-flow protection achieved by the proposed algorithm. When k is 256, Fig. 15 shows how the probability density function for each data rate depends on the distance between the AP and the mobile station. We see that the transmission ranges for each data rate (11 Mbps, 5.5 Mbps, 2 Mbps and 1 Mbps) are about 50 m, 70 m, 90 m and 115 m, which accords with a commercial data sheet [25]. The automatic data rate selection algorithm of the MAC protocol uses the receiver-based auto rate (RBAR) [13].

6.1. Fairness vs. mobility of mobile stations

The purpose of this simulation is to show that the throughput of the stable mobile station(s) will be guaranteed even though the link conditions of other mobile stations are fluctuating, mainly due to their mobility. The five mobile stations, Node A–E are located at 25 m, 60 m, 80 m, 102.5 m and 25 m from the AP, and initially connected to the AP with data rates of 11 Mbps, 5.5 Mbps, 2 Mbps, 1 Mbps and 11 Mbps respectively.

While the simulation is running, Node E is moving away from the AP at 0.1 m/s. In the meantime, the data rate of Node E changes successively from 11 Mbps to 5.5 Mbps, 2 Mbps and 1 Mbps. In this environment, we first generate 5 CBR connections, which are modeled as 2 Mbps flows (the packet size is 1024 bytes). The five CBR connections are flow 1 (from Node 1 to Node A), flow 2 (from Node 2 to Node B), flow 3 (from Node 3 to Node C), flow 4 (from Node 4 to Node D) and flow 5 (from Node 5 to Node E).

Parts (a), (b) and (c) of Fig. 16 show the throughput of the three algorithms: FIFO, T-WFQ and CATS. As the destination (Node E) of flow 5 moves away from the AP, we can see the throughput of the 5 flows change over time. Figure 16 (a) shows the throughput when FIFO is used as the queuing policy in the mobile stations. The throughputs of all 5 flows is almost the same because the mobile station with the lowest data rate degrades the throughput of the other mobile stations.

In Fig. 16 (b), using T-WFQ as the queuing algorithm, the throughput of each flow is differentiated and dependent on data rate. Since T-WFQ takes into account the data rate of each flow, it can provide different throughputs for each flow while the data rate of flow 5 remains stable (at 11 Mbps for about 400 sec). However, T-WFQ cannot support per-flow protection. As the data rate of flow 5 is decreased, the throughputs of the other flows change and are not protected. The decreased data rate of flow 5 means a decrease of the number of packets served by flow 5, because T-WFQ schedules packets to satisfy fairness between flows in terms of the packet transmission time. As the number of packets served by flow 5 drops, the time consumed by contention becomes a free resource and is reallocated to all the other flows. Therefore, in Fig. 16 (b), the throughput of the other flows increases as the mobile station E moves away from the AP.

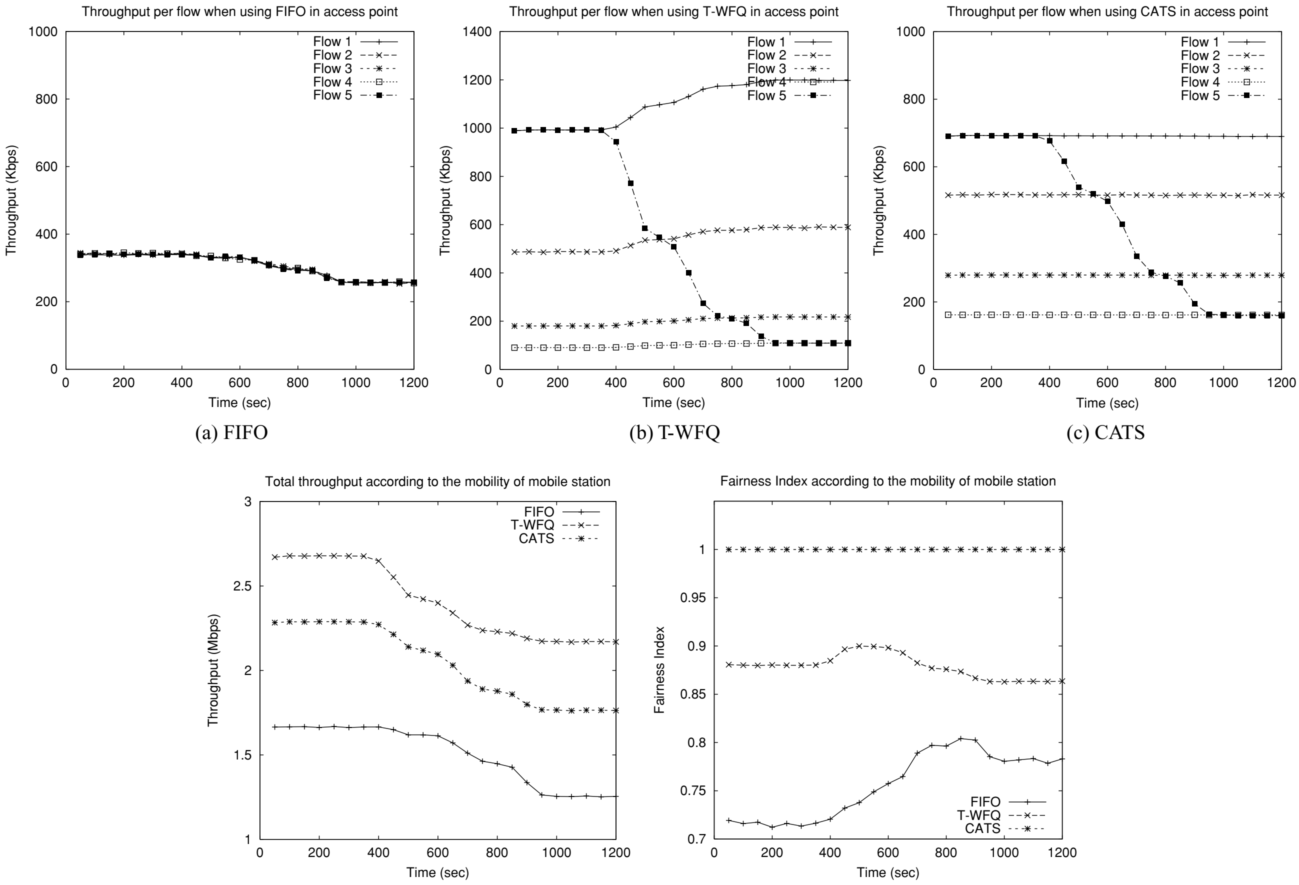
When the CATS algorithm is used, the throughput of each flow is not changed and the per-flow protection of throughput is guaranteed. As seen from Fig. 16 (c), the throughput of the other mobile stations remains unchanged even though the data rate of flow 5 changes.

Figure 16 (d) shows the total throughput of each scheduling algorithm: FIFO, T-WFQ and CATS. When using CATS, the total throughput is larger than that of FIFO but less than that of T-WFQ, which can achieve the high throughput because a high date-rate flow serves more packets. However, CATS can also improve the overall throughput by increasing the weights of flows with high data rates.

Figure 16 (e) shows the fairness indices of the three algorithms. When using CATS, we show that the fairness index remains close to 1, despite the mobility of the mobile stations. That means that all flows receive almost the same channel usage time. Both FIFO and T-WFQ have a fairness index less than 0.9, which means that temporal fairness is not guaranteed. Although the fairness index of T-WFQ ($0.85 \sim 0.9$) is higher than that of FIFO, it has a larger deviation than that of CATS.

Let us now evaluate D-CATS. (We will not consider T-WFQ since that algorithm is not operating in a distributed fashion.) Recall that D-CATS provides temporal fairness for uplink packet transmission. To evaluate the performance of D-CATS, we will transpose the source and destination of 5 CBR connections, as follows; flow 1 (from Node A to Node 1), flow 2 (from Node B to Node 2), flow 3 (from Node C to Node 3), flow 4 (from Node D to Node 4) and flow 5 (from Node E to Node 5).

As shown in Fig. 17 (a), with FIFO the throughput of each flow shows almost the same phenomena as in Fig. 16 (a). Because the mobile station E is moving away from the AP, the



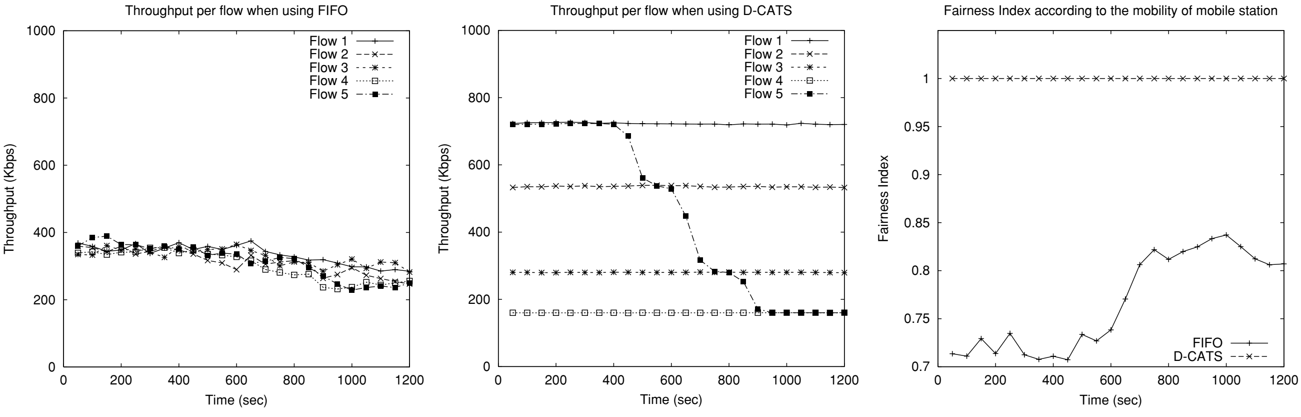
(a) FIFO (b) T-WFQ (c) CATS

(d) Total Throughput (e) Fairness Index

Fig. 16 Throughput and fairness indices of scheduling algorithms for downlink packet transmission

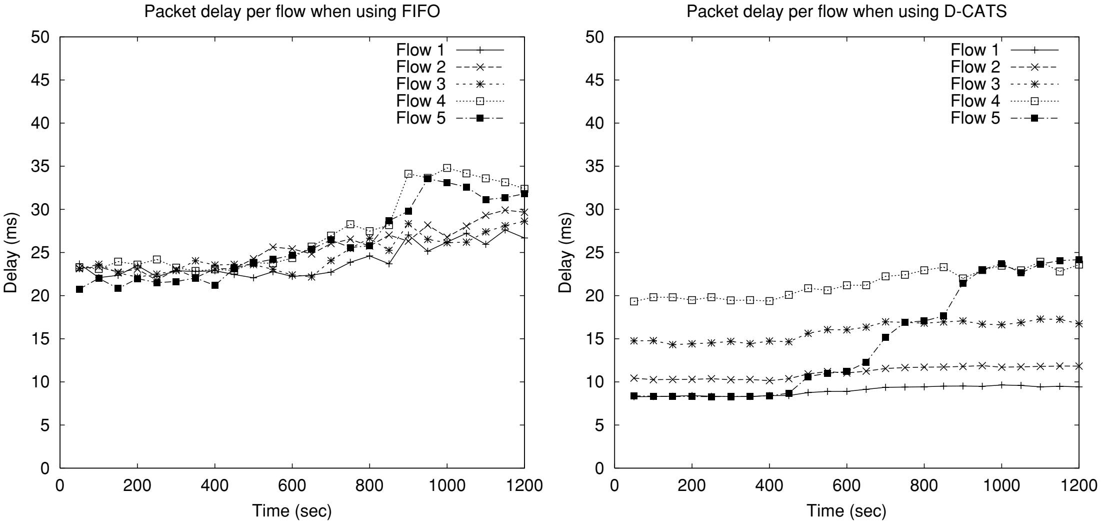
throughput reduces slightly. However, as shown in Fig. 17 (b), with D-CATS the throughput of the other mobile stations is protected regardless of the movement of the mobile station E. The throughput of flow 5 is degraded because the datarate of mobile station E reduced, but the throughputs of the other flows are not affected because the reserved channel usage time is guaranteed. Since mobile stations with high data rates receive guaranteed channel usage time in proportion to their weights, the total throughput is greater than with FIFO.

Figure 17 (c) shows the fairness index of FIFO and D-CATS in the case of the uplink packet transmission. Using D-CATS, the fairness index is nearly 1 and it is independent of the mobility of stations, while FIFO has a much lower fairness



(a) FIFO (b) D-CATS (e) Fairness Index

Fig. 17 Throughput and fairness indices of scheduling algorithms for uplink packet transmission



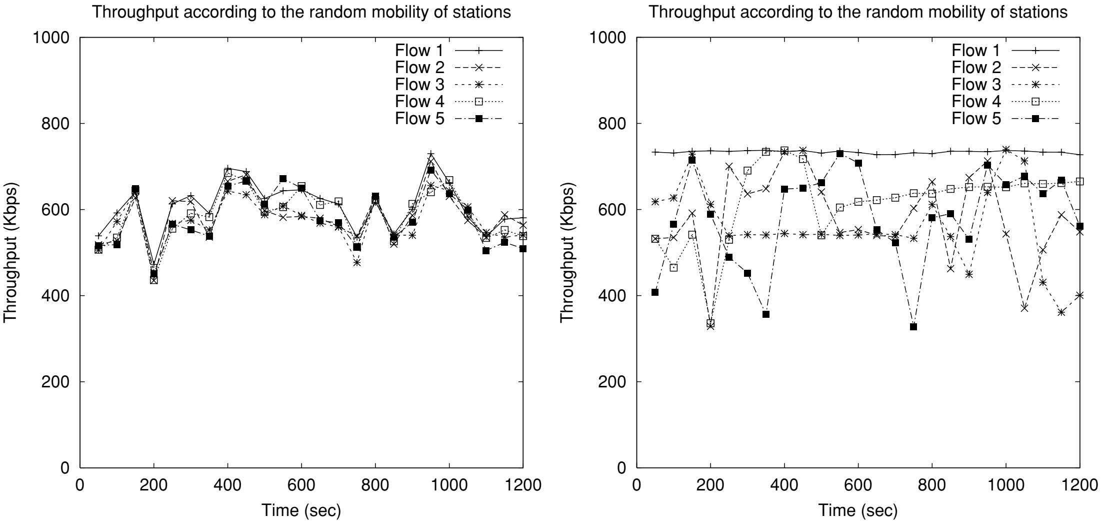
(a) FIFO (b) D–CATS

Fig. 18 Packet delay of scheduling algorithms for uplink packet transmission

index. Using FIFO, the fairness index is very poor because 802.11 DCF provides the same channel access opportunity to all mobile stations. The flows with high data rates get less channel usage time and the flows with lower data rates get more.

Figure 18 shows the packet delay of flows in the MAC layer when the uplink packets are transmitted using FIFO and D-CATS. When we use FIFO, the packet delay of each flow shows nearly the same value, but that increases by about 10 ms as the mobile station moves away from the AP. However, when we use D-CATS, the packet delay of each flow is much lower. In particular, the packet delays of flow 1 (data rate is 11 Mbps) and flow 2 (data rate is 5.5 Mbps), are about 8 ms and 10 ms, respectively. Using D-CATS, the delay of flow 1 and flow 2 is halved compared to that experienced with FIFO. We can also see that the delay jitter is reduced.

Figure 19 shows the throughput of each flow in the case of uplink packet transmissions. But in this simulation we use a random waypoint mobility model instead of a deterministic mobility model. To evaluate the per-flow protection, Node A, the source of flow 1, is located 25 m from the AP. The other four mobile stations, Node B - E, are moving around the AP according to a random waypoint model. The pause time, the maximum speed and the simulation time used in the model are 2 seconds, 10 m/s and 1200 second respectively, which represents a high-mobility scenario. In the simulations shown in Fig. 19 (a) and (b), the throughput of flow 1 is not protected when using FIFO, but, using D-CATS, the throughput



(a) FIFO (b) D–CATS

Fig. 19 Throughput of scheduling algorithms for uplink packet transmission with a random waypoint mobility model

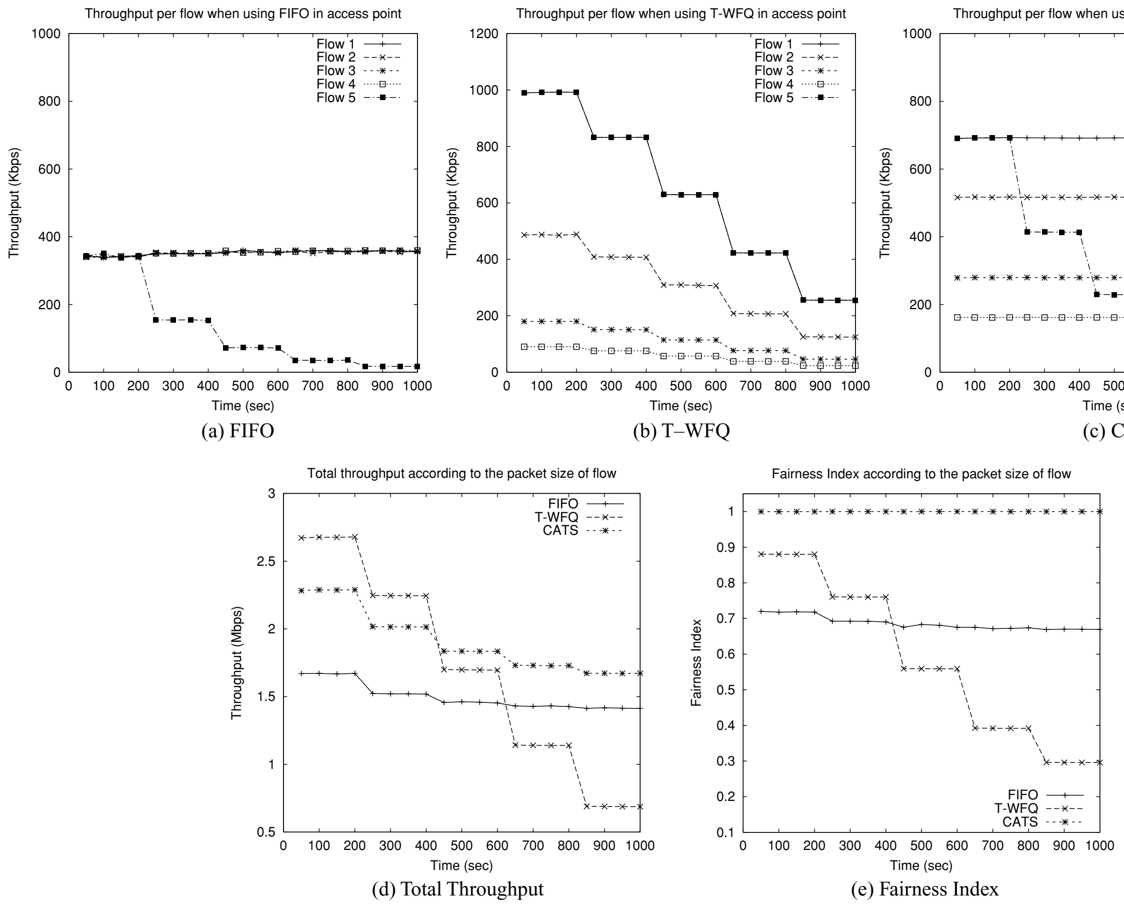
of flow 1 is protected regardless of the mobility of other mobile stations. Since per-flow temporal fairness is guaranteed, the throughput of each flow varies in a more independent manner.

6.2. Fairness vs. packet size of applications

We will now look at the effect of the packet size used by different applications. We initially simulate the same topology locate Node E at the same position as Node A, 25 m from the AP. The traffic pattern used in simulation is also the same (recall that each flows has 2 Mbps traffic) but the packet size for Node E is changed from 1024 bytes to 512 bytes, 256 bytes, 128 bytes, 64 bytes at 200 sec, 400 sec, 600 sec and 800 sec respectively.

Parts (a), (b), (c) of Fig. 20 show the throughput of three algorithms in this case. Using the FIFO algorithm [Fig. 20(a)], the throughput of flow 5 falls as its packet size is reduced. However other flows show throughputs similar to those in Fig. 16 (a), because the flow with the lowest data rate degrades the throughput of all mobile stations. Figure 20 (b) shows the throughput obtained when we use the T-WFQ algorithm. As the packet size of flow 5 falls, as mentioned in the above, the throughputs of the other flows also drop. Because T-WFQ performs packet scheduling in terms of packet transmission time, more packet transmissions by flow 5 are necessary to obtain an equal packet transmission time across all the flows. This is due to the fact that the packet transmission time of flow 5 is smaller than those of the other flows. To transmit more packets, flow 5 uses more contention overhead time. Other flows compromise their allocated resource to provide enough contention overhead time for flow 5. The throughputs of these other flows drop as the packet size is reduced for flow 5.

The overall performance of wireless LAN is especially degraded using T-WFQ, because it does not consider the contention overhead per packet. Consequently, it is important to protect the per-flow throughput while making allowance for the contention overhead, which is can be higher than the packet transmission time in 802.11b. Figure 20 (c) shows the throughput of CATS, which takes the contention overhead per packet into accounts. It is shown that the throughputs of the four flows are not changed even though the packet size of flow 5 is reduced. We can see that the CATS algorithm provides per-flow protection independent of the packet size of the application.



(a) FIFO

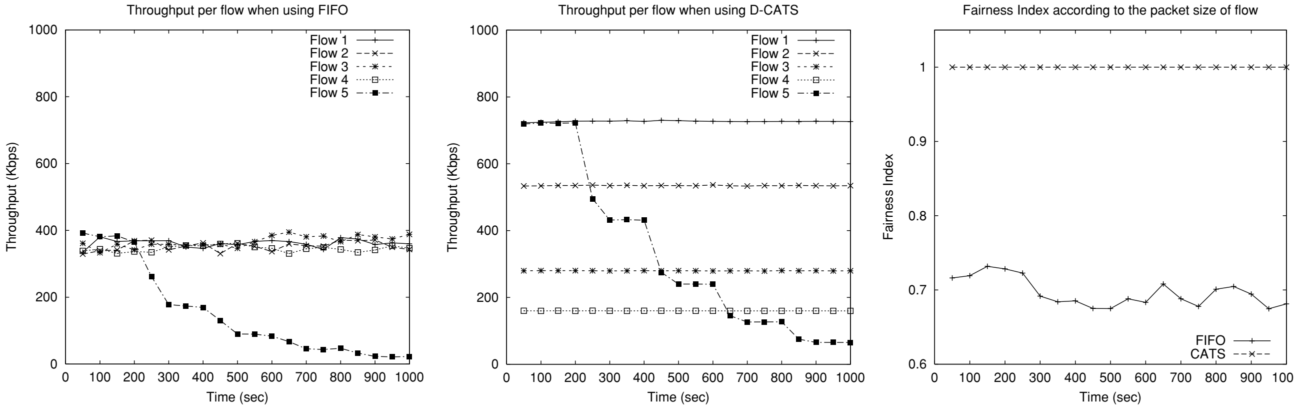
(b) T–WFQ

(c) CATS

(d) Total Throughput

(e) Fairness Index

Fig. 20 Throughput and fairness indices of scheduling algorithms for downlink packet transmission



(a) FIFO (b) D–CATS (c) Fairness Index

Fig. 21 Throughput and fairness indices of scheduling algorithms for uplink packet transmission

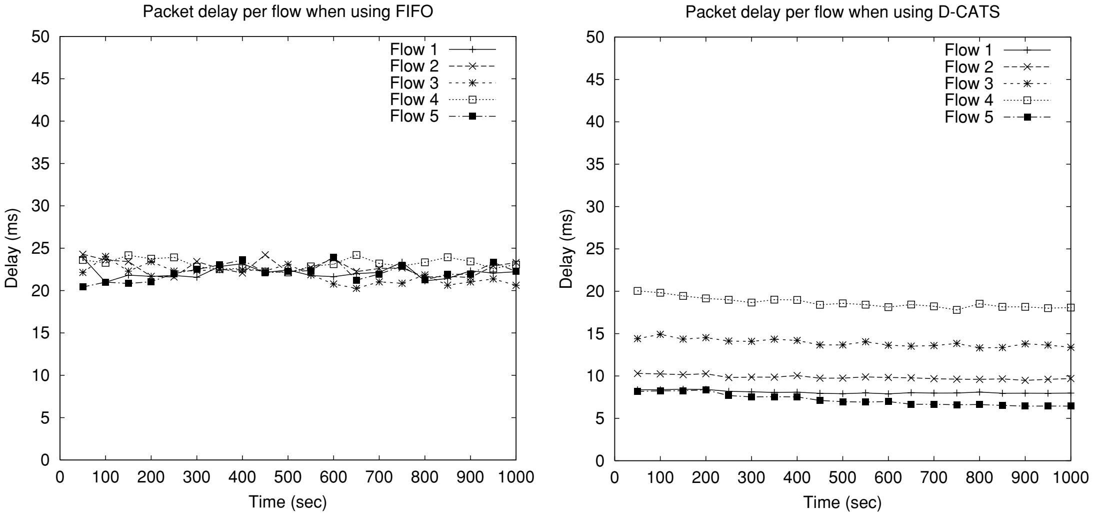
Figure 20 (d) shows the total throughput of the three scheduling algorithms. When CATS is used, the total throughput is slightly decreased as the throughput of flow 5 is decreased, but CATS always outperforms FIFO. Until 450 sec, the total throughput of T-WFQ is the highest, but it drops sharply as the packet size of flow 5 is reduced. For example, when the packet size of flow 5 becomes 64 bytes, the total throughput of T-WFQ becomes about 700 Kbps, which is half of the throughput with FIFO.

Figure 20 (e) shows the fairness index of the three algorithms. Using CATS, the fairness index remains at 1, which means that it is not affected by variations in packet size. However, FIFO and T-WFQ show poor fairness indices. In particular, using T-WFQ, the fairness index decreases in proportion to the change in packet size. As the packet size decreases, T-WFQ is supposed to transmit more packets in order to achieve the temporal fairness; but the contention overhead time is not considered in the design of T-WFQ, which therefore gives more channel usage time to a flow with small-sized packets.

We have also evaluated the temporal fairness of D-CATS when the uplink flows share the wireless link. We transpose the source and the destination of each flow, as in the previous simulation, to generate the uplink traffic. The throughput of flow 5 is reduced as its packet size decreases. When we use the FIFO algorithm [Fig. 21(a)], the throughputs achieved by all flows are quite similar and the throughput of flow 5 is reduced, as with D-CATS, when the packet size is reduced. However, FIFO does not provide per-flow protection.

Figure 21 (c) shows the fairness index for both FIFO and D-CATS. The fairness index of D-CATS is almost 1 but the fairness index of FIFO is very poor. By maintaining temporal fairness among the flows, we can support per-flow protection in mixed traffic of various packet sizes.

Figure 22 shows the packet delay for uplink packet transmissions. When D-CATS is used, the packet delays are less



(a) FIFO (b) D–CATS

Fig. 22 Packet delay of scheduling algorithms for uplink packet transmission

than 20 *ms*, which are lower than for FIFO. Flows, flow 1 and 2, which have higher data rates (11 Mbps and 5.5 Mbps), show especially short delays of 8 *ms* and 10 *ms*. Since the D-CATS algorithm guarantees per-flow temporal fairness, the delay experienced by each flow is determined by its data rate and protected against packet size variations in the other flows. When the packet size of flow 5 decreases, other flows are not affected, but the transmission delay is reduced. Additionally, in this case the delay jitter of each flow is very small.

6.3. Further design issues

All these simulation results have shown that CATS provides temporal fairness for each flow regardless of mobility and packet size. D-CATS, also provides temporal fairness. Although we have shown that CATS and D-CATS provide per-flow protection for the downlink and uplink flows, respectively, additional issues must now be considered in order to embody these algorithms in real 802.11 wireless LANs.

First, we need to evaluate the scalability of both CATS and D-CATS. In Fig. 23, we present total throughput against the number of active stations. In this simulation scenario, all mobile stations are located within 25m of the AP, and each mobile station generates CBR traffic to the AP.

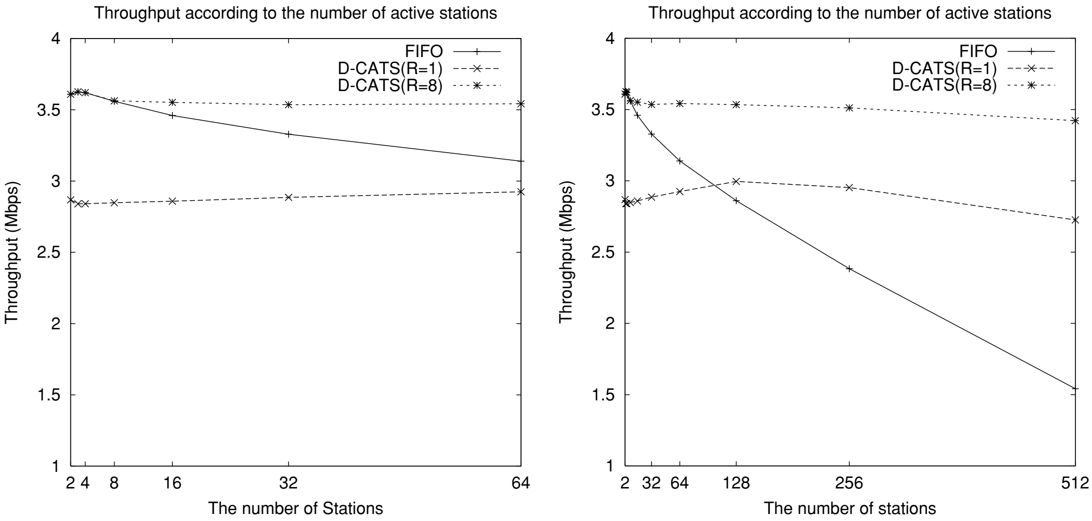
In Fig. 23 (a), the number of active mobile stations is increased from 2 to 64. We compare the total throughput achieved by FIFO and by D-CATS with its *R* parameter (line 7 in Algorithm 3) set to 1 and to 8. *R* is the number of active mobile stations that can contend for shared link access. When *R* is 1, the overall throughput is lower than with FIFO. This is due to the fact that just one mobile station is allowed to transmit a frame at any one time. But the throughput achieved by FIFO decreases as the number of mobile stations increases, because the probability of a collision increases. Although the throughput of D-CATS is not degraded in proportion to the number of active mobile stations, we still need to improve the throughput in this situation.

In Algorithm 3, we introduced the parameter *R* in order to solve this problem. As shown in Fig. 23 (a), the overall throughput of D-CATS becomes better when we set *R* to 8. It reaches about 3.5 Mbps and always outperforms FIFO. Figure 23 (b) shows the simulated result when the number of active mobile stations increases from 2 to 512. Even when the number of stations is very large, D-CATS outperforms FIFO and remains almost stable throughput; and under these conditions D-CATS (*R* is 8) shows a significant throughput advantage.

Kim and Hou [22] also proposed a model-based frame scheduling scheme (MFS) to improve the overall throughput achievable when the number of mobile stations is large. But because the MFS delays packets according to the network utility model to decrease the number of collision when there are more active mobile stations, it has some limitations. The network utility model of the MFS uses the collision count in each mobile station to estimate the number of active mobile stations. If the packet error rate does not reflect the real number of collisions, MFS misunderstands the number of active mobile stations, which reduces the overall throughput.

Similarly, veres at al [21] develop a delay model that takes into account channel utilization in turning the size of the contention window. They suggest two methods to estimate the channel utilization: virtual MAC and virtual source. But, their work focuses on a service differentiation mechanism similar to that of IEEE 802.11e.

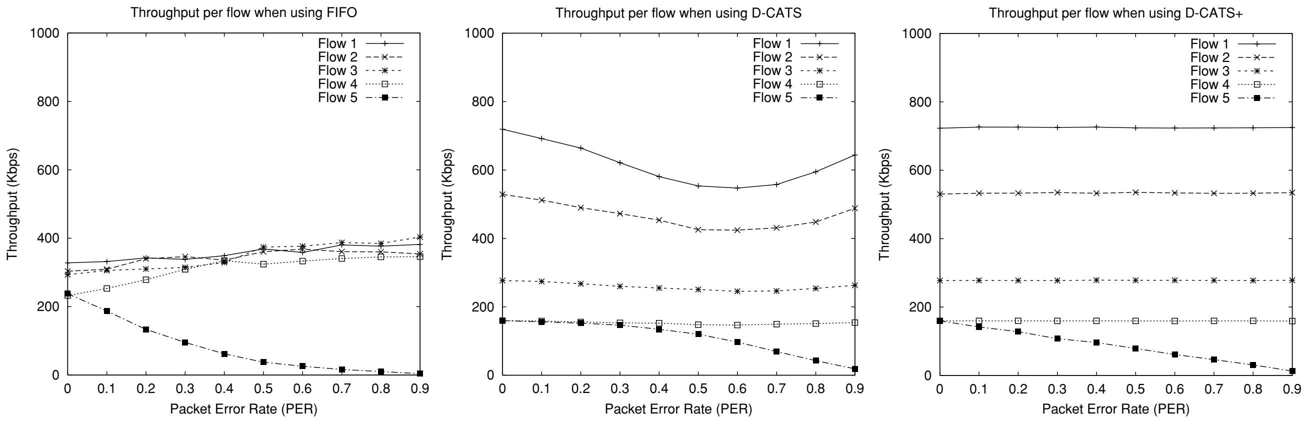
Second, we need to consider packet errors occurring in the wireless channel. Since IEEE 802.11 does not support



(a) Small active mobile stations

(b) Large active mobile statins

Fig. 23 Throughput of scheduling algorithms against to the number of active mobile stations



(a) FIFO (b) D–CATS (c) D–CATS+

Fig. 24 Throughput of three scheduling algorithms against packet error rate

any collision detection mechanism, each mobile station performs the same process for a collision as for a packet error. Consequently, both the temporal fairness and the collision overhead, as defined in this paper, are affected by the packet error rate.

To analyze the relationship between the packet error rate and temporal fairness, we performed a simulation. We use the topology shown in Fig. 13, except that the mobile station E is fixed at 115m from AP (its data rate is 1 Mbps). The traffic generation scenario is also the same, with 5 uplink CBR flows. The packet error rate of flow 5 ranges from 0 to 0.9, while the other flows have no packet error. Figure 24 shows how the throughput of the three algorithms are affected by packet error rate.

Using FIFO [Fig. 24(a)] the throughput of the error-free flows is not degraded by the error-prone flow (flow 5). But, using D-CATS [Fig. 24(b)] the error-prone flow degrades the throughput achieved by the error-free flows. The throughput of Flow 1 is particularly badly affected when the packet error rate increases. The packet errors in flow 5 are caused by the increase in contention overhead, although the mobile stations believe that the transmission errors are caused by collisions. So, as the contention overhead increases, the throughput of the error-free flows falls with their channel usage time. D-CATS+ solves this problem by using the optional RTS-CTS sequence of the 802.11 DCF. Now the throughputs of the error-free flows, are not degraded regardless of the packet error rate of the error-prone flow.

Figure 25 shows how the contention overhead time of the error-free flows (i.e., flows 1–4) varies with the packet error rate. Using D-CATS, the contention overhead time grows with the packet error rate. As shown in Fig. 24(b), the throughput degradation becomes very important at the 0.6 packet error rate. This error rate also corresponds to the highest contention overhead time. However, when we use the D-CATS+, the contention overhead is about 1.45 *ms* regardless of the packet error rate, which means that D-CATS+ provides better per-flow protection and guarantees temporal fairness even in the presence of transmission errors.

We also simulated a situation in which every mobile station has an unstable wireless channel due to small-scale fading (such as multi-path fading). For this purpose, we change a Ricean parameter of the propagation model from 256 to 6. With this value of the Ricean parameter k is 6, the probability density function of each data rate depends on the distance between the AP and the mobile station as shown in Fig. 26. We also changed the location of each mobile station. Node A - E are now positioned 5 m, 65 m, 90 m, 115 m and 5 m, respectively. As shown in Fig. 26, only Node A is always connected to the AP with a 11Mbps data rate. Other mobile stations have the unstable channel conditions and frequently change their data rates depending on the link condition. While the simulation is running, Node E is moving away to a new location, at a distance of 155m from the AP, with a speed of 0.1m/s.

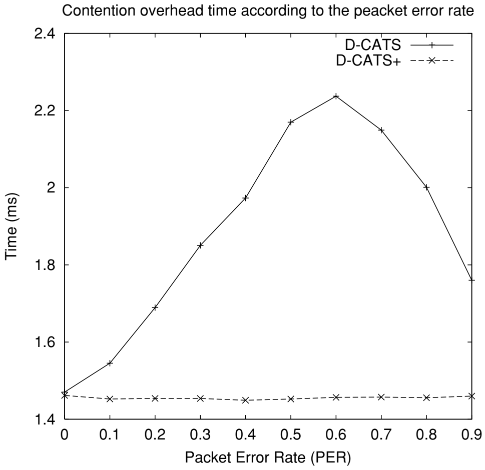


Fig. 25 Contention overhead time of scheduling algorithms in an error-prone wireless LAN

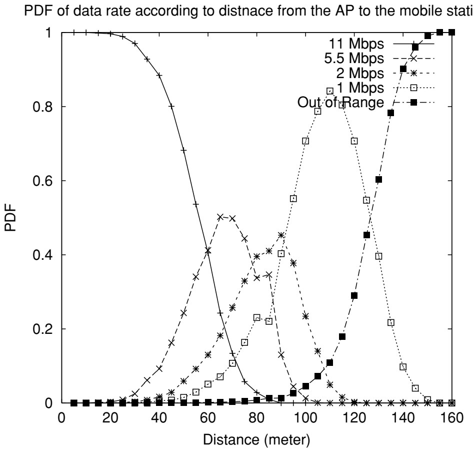


Fig. 26 PDF of data rate against distance from the AP to the mobile station

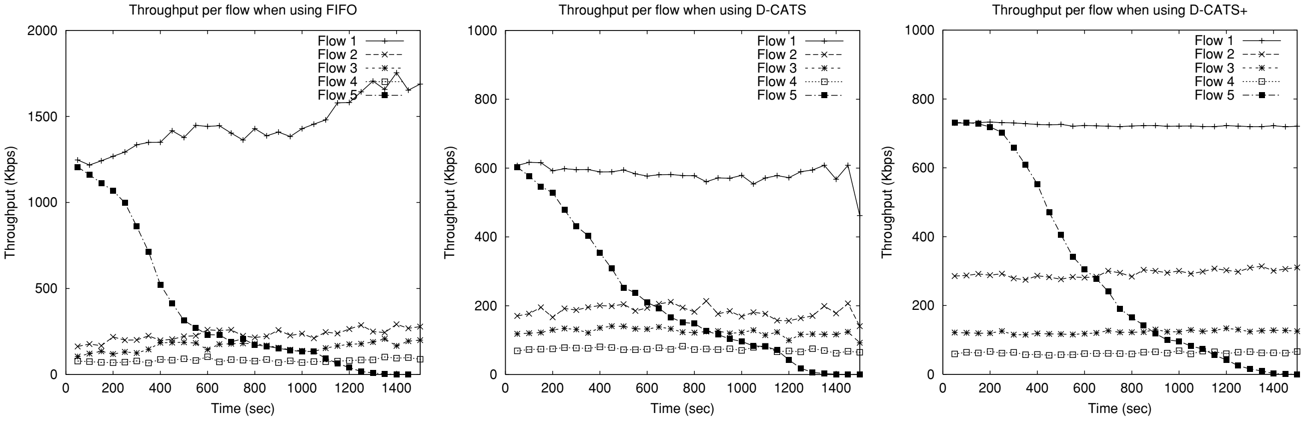
Figure 27 shows the throughput of the three algorithms, FIFO, D-CATS, D-CATS+ in this case. Using FIFO, the throughput does not show the performance anomaly seen in Figs. 16 (a) and 19 (a). Flow 1 shows the highest throughput while flows 2–4 show much smaller throughputs. Since the channel conditions for these flows are fluctuating due to the small-scale fading, their packet error rates are very high. The average packet error rates of flow 2, flow 3 and flow 4 are 0.097, 0.195 and 0.715, respectively. Because of the unstable channel conditions, these mobile stations have a lower channel access probability than flow 1, because of the exponential backoff mechanism of IEEE 802.11. While Node E is moving away from the AP, the throughput of flow 1 increases because the packer error rate of flow 5 is also increasing. Consequently [Fig. 27(a)] the mobile station with the stable channel condition has a higher performance than those with unstable channel conditions.

Figures 27(b) and 27(c) show throughputs of D-CATS and D-CATS+ in the same situation. We can see two differences between the performance of D-CATS and D-CATS+. First, using D-CATS, the average throughput of each flow is less than the throughput of the same flow with D-CATS+. When packet errors happen, D-CATS adds the packet corruption time into the contention overhead time of the flow that successfully transmits the next packet. The increase in contention overhead time caused by packet corruption means that the priority of this flow becomes lower. Thus, a flow competing with other flows with bad channel conditions is penalized by being made to share its resource with other flows. Second, the throughput variation of flows using D-CATS is higher than that with D-CATS+. In particular, by comparing the throughput variation of flows 1 and 2, we can see that D-CATS+ provides a tighter short-term fairness. The reason is that the variation of contention overhead time with D-CATS is much larger than that with D-CATS+. Even with short-term fading D-CATS+ guarantees per-flow protection.

7. Conclusion

Currently, wireless LANs using IEEE 802.11a and 802.11b support high bandwidth but the quality of service for applications is not guaranteed. The motivation of this paper is to design scheduling algorithms that provide a better support for quality of service by maintaining per-flow protection. To realize per-flow protection, we need to satisfy two requirements: *mobility independence* and *application independence*. Mobility independence means that the throughput of fixed stations is not affected by the mobility of other mobile stations. Application independence means that the throughput of an application with a fixed packet size is not affected by the packet size variation of other applications.

We have proposed a contention aware temporally fair scheduling algorithm (CATS) to satisfy these two requirements in CSMA/CA based 802.11 wireless LANs. The main design principle of CATS is that time should be fairly allocated despite the variable data rate and packet size of mobile



(a) FIFO

(b) D–CATS

(c) D–CATS+

Fig. 27 Throughput of scheduling algorithms in a wireless channel with small-scale fading

stations. CATS determines the scheduling order of packets as their virtual finish time, which is calculated by taking into account the data rate, the packet size and the contention overhead of each flow. We have also proposed a decentralized contention aware temporal packet scheduling algorithm (D-CATS), which is decentralized version of CATS. Both CATS and D-CATS can easily be implemented without changing the existing 802.11 MAC protocol.

We have also verified that both CATS and D-CATS support mobility-independent and application-independent QoS, using an NS-2 simulator. In a decentralized environment, D-CATS shows reduced packet delay and delay jitter, since it guarantees a fixed share of the wireless link resource to each flow. In addition, D-CATS outperforms the FIFO algorithm in terms of throughput, since D-CATS can control the number of mobile stations that are permitted to contend for the shared wireless link. Lastly, D-CATS can support per-flow protection in environments with a high packet error rate by using the D-CATS+ algorithm, which utilizes RTS and CTS frame exchange.

In future work, we will investigate how to extend CATS to address the fairness problem in multi-hop wireless networks which use the 802.11 MAC protocol. We also plan to implement our algorithms in a real system.

Appendix

Proof of Lemma 2

Proof: In the CSMA/CA wireless network, the channel usage time of each flow should include the contention overhead time per packet. So, the relative fairness bound in CASMA/CA network, $\mathrm{RFB}_{\mathrm{CSMA/CA}}(i, j)$, is as the follow,

$$\left|\frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)}\right| \le \mathrm{RFB}_{\mathrm{CSMA/CA}}(i,j)$$

$$\left|\frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)}\right| = \left|\frac{\sum_{k=1}^{n} P(i,k)}{\phi(i)\cdot C(i)} + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)} - \frac{\sum_{k=1}^{m} P(j,k)}{\phi(j)\cdot C(j)} - \frac{m\cdot \mathrm{CO}(t)}{\phi(j)}\right|,$$

where n and m are the number of packets served in flow i and j, respectively. We can obatin the $\mathrm{RFB}_{\mathrm{CSMA/CA}}(i, j)$ thrugh deriving the relation between n and m by using $\mathrm{RFB}(i, j)$ of the Lemma 1.

(i) When $\mathrm{RFB}(i, j) = 0$,

$$\frac{\sum_{k=1}^{n} P(i,k)}{\phi(i)\cdot C(i)} - \frac{\sum_{k=1}^{m} P(j,k)}{\phi(j)\cdot C(j)} = 0$$

$$\sum_{k=1}^{m} P(j,k) = \sum_{k=1}^{n} P(i,k)\frac{\phi(j)\cdot C(j)}{\phi(i)\cdot C(i)}$$

$$m = n\cdot \frac{\phi(j)\cdot C(j)\cdot \mathrm{P}_{\mathrm{avg}}(i)}{\phi(i)\cdot C(i)\cdot \mathrm{P}_{\mathrm{avg}}(j)}$$

$$\therefore \left|\frac{\sum_{k=1}^{n} P(i,k)}{\phi(i)\cdot C(i)} + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)} - \frac{\sum_{k=1}^{m} P(j,k)}{\phi(j)\cdot C(j)} - \frac{m\cdot \mathrm{CO}(t)}{\phi(j)}\right|$$

$$= \left|\mathrm{RFB}(i,j) + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)}\left(1 - \frac{C(j)\cdot \mathrm{P}_{\mathrm{avg}}(i)}{C(i)\cdot \mathrm{P}_{\mathrm{avg}}(j)}\right)\right|$$

(ii) When

$$\mathrm{RFB}(i,j) = \frac{\mathrm{P}_{\max}(i)}{\phi(i)\cdot C(i)} + \frac{\mathrm{P}_{\max}(j)}{\phi(j)\cdot C(j)},$$

$$\frac{\sum_{k=1}^{n} P(i,k)}{\phi(i)\cdot C(i)} - \frac{\sum_{k=1}^{m} P(j,k)}{\phi(j)\cdot C(j)} = \frac{\mathrm{P}_{\max}(i)}{\phi(i)\cdot C(i)} + \frac{\mathrm{P}_{\max}(j)}{\phi(j)\cdot C(j)}$$

$$\sum_{k=1}^{m} P(j,k) = \sum_{k=1}^{n} P(i,k)\frac{\phi(j)\cdot C(j)}{\phi(i)\cdot C(i)} - \frac{\phi(j)\cdot C(j)\cdot \mathrm{P}_{\max}(i)}{\phi(i)\cdot C(i)} - \mathrm{P}_{\max}(j)$$

$$m = \frac{C(j)\cdot\phi(j)}{\mathrm{P}_{\mathrm{avg}}(j)} \times \left(\frac{n\cdot \mathrm{P}_{\mathrm{avg}}(i)}{C(i)\cdot\phi(i)} - \frac{\mathrm{P}_{\max}(i)}{\phi(i)\cdot C(i)} - \frac{\mathrm{P}_{\max}(j)}{\phi(j)\cdot C(j)}\right)$$

$$\therefore \left|\frac{\sum_{k=1}^{n} P(i,k)}{\phi(i)\cdot C(i)} + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)} - \frac{\sum_{k=1}^{m} P(j,k)}{\phi(j)\cdot C(j)} - \frac{m\cdot \mathrm{CO}(t)}{\phi(j)}\right|$$

$$= \left|RF + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)}\left(1 - \frac{m\cdot\phi(i)}{n\cdot\phi(j)}\right)\right|$$

$$= \left|RF + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)}\left(1 + \frac{C(j)}{\mathrm{P}_{\mathrm{avg}}(j)} \times \left(\frac{\frac{\mathrm{P}_{\max}(j)}{n} - \mathrm{P}_{\mathrm{avg}}(i)}{C(i)} + \frac{\phi(i)\cdot \mathrm{P}_{\max}(j)}{n\cdot\phi(\mathrm{j})\cdot C(j)}\right)\right)\right|$$

$$\simeq \left|RF + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)}\left(1 - \frac{C(j)\cdot \mathrm{P}_{\mathrm{avg}}(i)}{C(i)\cdot \mathrm{P}_{\mathrm{avg}}(j)}\right)\right|$$

From case (i) and (ii),

$$\mathrm{RFB}_{\mathrm{CSMA/CA}}(i,j) = \left|\mathrm{RFB}(i,j) + \frac{n\cdot \mathrm{CO}(t)}{\phi(i)}\left(1 - \frac{C(j)\cdot \mathrm{P}_{\mathrm{avg}}(i)}{C(i)\cdot \mathrm{P}_{\mathrm{avg}}(j)}\right)\right|$$

we can derive the following differential equation.

$$\begin{aligned}&\frac{\partial|W_i(t_1,t_2)/\phi_i - W_j(t_1,t_2)/\phi_j|}{\partial n}\\ &\quad = \left|\frac{\mathrm{CO(t)}}{\phi(i)}\left(1-\frac{C(j)\cdot \mathrm{P}_{\mathrm{avg}}(i)}{C(i)\cdot \mathrm{P}_{\mathrm{avg}}(j)}\right)\right| \qquad \square\end{aligned}$$

Proof of Lemma 4

Proof: The maximum of RFB(i, j) is defined as δ,

$$\left|\frac{W_i(t_1,t_2)}{\phi(i)} - \frac{W_j(t_1,t_2)}{\phi(j)}\right| \leq \mathrm{RFB}(i,j) \leq \delta, \forall (i,j) \in B(t)$$

$$\left|\frac{\phi(j)\cdot W_i(t_1,t_2)}{\phi(i)} - W_j(t_1,t_2)\right| \leq \phi(j)\cdot\delta$$

Since the total serviced time of all backlogged flows is equal to $(t_1 - t_1)$,

$$\begin{aligned}&\left|W_i(t_1,t_2) - \frac{\phi(i)}{\sum_{k\in B(t)}\phi(k)}(t_2-t_1)\right|\\ &\quad \leq \frac{\phi(i)}{\sum_{k\in B(t)}\phi(k)}\sum_{k\in B(t)}\phi(k)\cdot\delta\\ &\therefore W_i(t_1,t_2) = \frac{\phi(i)}{\sum_{k\in B(t)}\phi(k)}(t_2-t_1) \pm \phi(i)\cdot\delta \qquad \square\end{aligned}$$

References

1. IEEE WG, Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications," IEEE 802.11 Standard (1999).
2. IEEE WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band, IEEE 802.11 Standard (1999).
3. IEEE WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer Extension in the 2.4 GHz Band, IEEE 802.11 Standard (1999).
4. IEEE WG, IEEE 802.11e/D4.0, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS) (November 2002).
5. A. K. Parekh and R. G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single node case, IEEE/ACM Transactions on Networking 1(3) (June 1993).
6. V. Bharghavan, S. Lu and T. Nandagopal, Fair queuing in wireless networks: issues and approaches, IEEE Personal Communication Magazine 6(1) (Februrary 1999).
7. T. Ng, I. Stoica and H. Zhang, Packet fair queuing algorithms for wireless networks with location-dependent errors, IEEE INFOCOM 1998, San Francisco, USA (1998).
8. J. Bennett and H. Zhang, Hierarchical packet fair queuing algorithms, IEEE/ACM Transactions on Networking 5(5) (October 1997) 675–689.
9. Y. Yi, Y. Seok, T. Kwon, Y. Choi and J. Park, W^2F^2Q: Packet fair queuing in wireless packet networks, in: *Proc. ACM WoWMoM 2000* (Boston, USA, 2000).
10. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18(3) (March 2000).
11. R. Jain, G. Babic, B. Nagendra and C. Lam, Fairness, call establishment latency and other performance metrics, Tech. Rep. ATM Forum/96-1173, ATM Forum Document (August 1996).
12. A. Kamerman and L. Monteban, WaveLAN II: A high-performance wireless LAN for the unlicensed band, Bell Labs Technical Journal (Summer 1997) 118–133.
13. G. Holland, N. Vaidya and P. Bahl, A rate-adaptive MAC protocol for multi-hop wireless networks, in: *Proceedings of ACM MOBICOM 2001* (Rome, Italy, 2001).
14. B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly, OAR: an opportunistic MAC for multirate ad hoc networks, ACM Journal of Mobile Networks and Applications (MONET) (2004).
15. Y. Liu, S. Gruhl and E. Knightly, WCFQ: an opportunistic wireless scheduler with statistical fairness bounds, IEEE Transactions on Wireless Communication 2(5) (September 2003).
16. M. Bottigliengo, C. Casetti, C. Chiasserini and M. Meo Short-term fairness for TCP flows in 802.11b WLANs, in: *Proc. IEEE INFOCOM 2004* (Hong Kong, China, 2004).
17. G. Cantieni, Q. Ni, C. Barakat and T. Turletti, Performance analysis of finite load sources in 802.11b multirate environments, INRIA Research Report No. 4881 (July 2003).
18. G. Berger-Sabbatel, F. Rousseau, M. Heusse and A. Duda, Performance anomaly of 802.11b, In Proc. IEEE INFOCOM 2003 (San Francisco, USA, 2003).
19. S. Garg and M. Kappes, Experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks, in: *Proc. IEEE Wireless Communications and Networking Conference* (New Orleans, USA, 2003).
20. S. Garg and M. Kappes, Can I add a VoIP call? in: *Proc. IEEE International Conference on Communications 2003* (Anchorage, Alaska, 2003).
21. A. Veres, A.T. Campbell, M. Barry, and L. H. Sun, Supporting service differentiation in wireless packet networks Using Distributed Control, IEEE Journal of Selected Areas in Communications 19(10) (2001).
22. H. Kim and J. C. Hou, Improving protocol capacity with model-based frame scheduling in IEEE 802.11-operated WLANs, in: *Proc. ACM MOBICOM 2003* (San Diego, Calfornia, USA, September 2003).
23. VINT group, UCB/LBNL/VINT Network Simulator – ns (version 2), http://www.isi.edu/nsnam/ns/.
24. R. Punnoose, P. Nikitin and D. Stancil, Efficient simulation of Ricean fading within a packet simulator, in: *Proc. IEEE Vehicular Technology Conference* (Boston, USA, September 2000).
25. Datasheet for ORiNOCO 11 Mbit/s Network Interface Cards (2001). ftp://ftp.orinocowireless.com/pub/docs/ORINOCO/.