Mobile Networks and Applications 9, 297–309, 2004

Efficient Packet Scheduling Using Channel Adaptive Fair Queueing in Distributed Mobile Computing Systems

LI WANG, YU-KWONG KWOK, WING-CHEONG LAU and VINCENT K.N. LAU
Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

Abstract. In a distributed mobile computing system, an efficient packet scheduling policy is a crucial component to achieve a high utilization of the precious bandwidth resources while satisfying users' QoS (quality of service) demands. An important class of scheduling techniques, namely, the wireless fair queueing algorithms, have been extensively studied recently. However, a major drawback in existing approaches is that the channel model is overly simplified – a two-state channel (good or bad) is assumed. While it is relatively easy to analyze the system using such a simple model, the algorithms so designed are of a limited applicability in a practical environment, in which the level of burst errors is time-varying and can be exploited by using channel adaptive coding and modulation techniques. In this paper, we first argue that the existing algorithms cannot cater for a more realistic channel model and the traditional notion of fairness is not suitable. We then propose a new notion of fairness, which bounds the actual throughput normalized by channel capacity of any two data connections. Using the new fairness definition, we propose a new fair queueing algorithm called CAFQ (Channel Adaptive Fair Queueing), which, as indicated in our numerical studies, outperforms other algorithms in terms of overall system throughput and fairness among error prone connections.

Keywords: distributed mobile computing, scheduling, wireless networks, fair queueing, fairness, performance guarantees, quality of service

1. Introduction

In a distributed mobile computing system, the mobile devices compete to gain access to the channel in order to transmit/receive data to/from the base-station. While the uplink (from the mobile devices to the base-station) communication is a multiple access control problem, the downlink (from the base-station to the mobile devices) communication is a data multiplexing and packet scheduling problem [3]. In our study, we focus on the latter problem which concerns about how to fairly schedule packets, flowing into the base-station via multiple parallel connections destined for different mobile devices, to make use of the downlink channel efficiently.

Fair queueing algorithms, which are the major techniques for packet scheduling, have been extensively studied in wireline networks for providing QoS (Quality of Service) guarantees to connections among end hosts. In recent years, as wireless networks proliferate, researchers have also put much efforts in extending the fair queueing techniques for applications in a wireless environment [4,15,23]. However, a major drawback in these wireless fair scheduling techniques is that the channel model is rather unrealistic: the channel is either in a "good" state (or perfect state) in which a session (defined as an active data connection) can transmit using full bandwidth, or in a "bad" state in which a session cannot transmit any data. In reality, using state-of-the-art channel adaptive techniques [14], the transmitter/receiver in a wireless network can exploit the time-varying nature of the channel and accordingly adjust the effective throughput by choosing an appropriate level of FEC (forward error correction). Simply put, in techniques such as ABICM (Adaptive Bit-by-Bit Interleaved Channel Modulation) [14] or MQAM [7], when the channel condition is not good (by checking the pilot symbols in a feedback channel on the uplink), the amount of protection can be re-adjusted by choosing a different channel coding and modulation mode [14]. Thus, even in a so-called "bad" channel state, a mobile device can in fact transmit data and realize a possibly lower effective throughput, instead of being totally unable to transmit.

With such a realistic channel adaptive transmission method (e.g., using a channel adaptive MAC layer [13]), intuitively the overall system throughput will be enhanced. However, a question remains is what a scheduler should do in order to maintain the fairness among the sessions in the system, which, more often than not, are in a "not so good" channel states. In previous algorithms, the answer is simple because those algorithms simply regard a session as "dormant" (cannot transmit) if it is in a "not so good" channel state – only a session with a "perfect" (the best) channel state can transmit.

In view of the fact that existing algorithms cannot cater for the situation where multiple channel quality levels exist, in this paper we propose a new notion of fairness, which then induces our proposed algorithm called CAFQ (Channel Adaptive Fair Queueing). As indicated by our numerical studies, the CAFQ algorithm outperforms other existing state-of-the-art algorithms in that CAFQ produces a higher overall system throughput and maintains fairness even among the sessions without perfect channel conditions.

The balance of the paper is as follows. In the next section, we first provide a discussion on the different fairness notions in wireline and wireless networks, and then demonstrate that a new fairness notion is needed in order to cater for the multi-level channel qualities. We also present a detailed qualitative analysis of several representative wireless fair queueing techniques. The objective of the analysis is to illustrate the deficiencies in the existing approaches. We then describe our new notion of fairness and the CAFQ algorithm in detail in section 3. Simulation results are presented in section 4. Finally, we provide some concluding remarks in section 5.

2. Fairness notions

2.1. Effort fair and outcome fair

In a broad sense, fairness can be defined with respect to two aspects: effort and outcome [5]. Intuitively, a policy is called *effort fair* if the allocation of services to different sessions is fair, without regard to the actual amount of data successfully delivered by the sessions using the allocated services. Informally, "fair" means a session gets the service amount that it deserves to get. On the other hand, a policy is called *outcome fair* if the actual realized data throughput among the sessions is fair.

Effort fair. A scheduler is fair if the bandwidth (e.g., the amount of time slots) the system allocates to different sessions is proportional to the different service shares. Mathematically, that means the difference between the normalized services the system allocates to any two sessions i and j is bounded as follows:

$$\left|\frac{S_i(t_1,t_2)}{r_i} - \frac{S_j(t_1,t_2)}{r_j}\right| < \epsilon, \tag{1}$$

where $S_i(t_1,t_2)$ denotes the allocated service of a certain session i during time interval $(t_1,t_2]$, r_i is the requested service share, and ϵ is a finite constant. Such a fair scheduler can be considered as effort fair [6] in that the scheduler only guarantees the effort expended on the sessions is fair, without regard to the actual throughput achieved by the different sessions.

Outcome fair. A scheduler is fair if the difference between the normalized amount of realized throughput of any two sessions i and j is bounded as follows:

$$\left|\frac{T_i(t_1,t_2)}{r_i} - \frac{T_j(t_1,t_2)}{r_j}\right| < \epsilon, \tag{2}$$

where $T_i(t_1,t_2)$ denotes the actual throughput session i achieves during the time interval $[t_1,t_2]$. Such a fair scheduler can be considered as outcome fair [16] in that the scheduler tries to provide a fair actual performance achieved by the sessions (rather than the "nominal" performance as in the effort fair definition discussed above).

In a TDMA system, "effort" is the number of time-slots allocated, while "outcome" is the actual data throughput using the allocated time-slots. Note that a "variable actual throughput" is manifested by the fact that some data may be lost due to poor channel conditions and thus, inducing retransmissions; or, in adaptive FEC schemes such as ABICM [14] or MQAM [7], the amount of data protection varies according to the channel conditions (detailed in section 3.2).

2.2. Fairness notions for wireline networks

In wireline networks, the classical fairness notion is based on the Generalized Processor Sharing (GPS) concept [20]. In GPS, the notion of fairness is defined in the following manner. Let $G_i(t_1,t_2)$ denote the throughput of session i in a given time interval $(t_1,t_2]$, where $i \in F$ and F is the set of all backlogged sessions (i.e., sessions having data pending to be sent) in the system. A scheduler is fair if and only if, for any $j \in F$ such that both session i and session j are continuously backlogged in $(t_1,t_2]$, we have:

$$\frac{G_i(t_1,t_2)}{G_j(t_1,t_2)} \geqslant \frac{r_i}{r_j}, \tag{3}$$

where r_i and r_j are the allocated rates of the two sessions.

Because of the high computational complexity involved in keeping track of the GPS fairness, there are many other improved variants [28] such as: D-EDD, FFQ, VC, WFQ, WF2Q, and SCFQ (see [28] for a detailed survey). For all these fairness notions, one important point to note is that in a wireline network, effort fair and outcome fair are equivalent because the channel (link) state is constant.

While a fairness notion like GPS (or its variants) works well in a wireline network, it is unsuitable for a wireless environment in which the channel quality of different sessions may vary considerably due to different shadowing and fading effects [21]. Specifically, using such a fairness measure, the scheduler will try to allocate the same throughput levels to different sessions. However, this is not efficient from a resource utilization point of view because those sessions suffering from deep fading (i.e., channel quality is not good) will not be able to utilize the time-slots efficiently (e.g., data loss may occur more frequently). Essentially, in a wireless network, effort is not necessarily equal to outcome. A more intelligent method is to allow the sessions having better channel states to proceed first.

2.3. Fairness notions for wireless networks

2.3.1. Overview

Recently, much research has been done on devising new algorithms for fair queueing in wireless networks. Many algorithms have been proposed [4,6,10,11,15–19,22,23]. For an excellent discussion of wireless fairness notions and a useful unified framework, the reader is referred to the recent papers by Bharghavan, Nandagopal, and Lu [4,16,18]. The general idea of wireless scheduling algorithms is as follows: the scheduler simulates an error-free system running a wireline packet scheduling algorithm when the sessions are in the perfect channel state (at which the effective throughput is maximum). When the session that is scheduled to transmit data encounters a bad channel state, it will give up the transmission opportunity to other error-free sessions (i.e., in the perfect channel state), then these error-free sessions will give their transmission rights back to the error session to compensate when it escapes from a bad channel state. Thus, essentially, the scheduler tries to swap the allocated time-slots between error-free sessions and error-prone sessions when sessions encountering errors. The goal is to hide the short term channel error burst from the end users. The system maintains long term fairness at the expense of instantaneous fairness between sessions.

In our study, we have considered the following existing scheduling algorithms for wireless networks: WPS (Wireless Packet Scheduling) [16], IWFQ (Idealized Wireless Fair Queueing Algorithm) [16], CIF-Q (Channel-Condition Independent Fair Queueing) [19], SBFA (Server Based Fairness Algorithm) [22], CS-WFQ (Channel State Independent Wireless Fair Queueing) [15], ELF (Effort Limited Fairness) [6], Proportional Fairness [9], and WFS (Wireless Fair Service) [18]. A scrutiny of these current scheduling algorithms for wireless networks reveals that in most of these algorithms, there are two common major deficiencies:

1. The channel model is too simple and not realistic. Only a two-state (good or bad) model is used.
2. There is few analysis for sessions which have bad channel states.

On the surface, these previous algorithms work well in that they schedule the error-free sessions to transmit data while leaving the error sessions (in a bad channel state) waiting until their channel states become good again. Thus, to maintain fairness, it suffices to guarantee that the error sessions can catch up (i.e., get back the missing service share) within a bounded period of time. However, usually nothing can be said about the behavior and the time bound of the error period. Furthermore, the key assumption, which, we believe, is the major drawback, is that a session in a bad channel state can transmit nothing. This is undeniably an over-simplification in view of the fact that channel-adaptive and variable rate physical layer protocols are commonly sought to combat the time-varying nature of wireless channels. Algorithms that use such a simplified assumption include: CIF-Q, IWFQ, SBFA, and WFQ.

On the other hand, the more practical algorithms, such as the ELF, CS-WFQ (uses a similar principle as in ELF), proportional fair, and our proposed algorithm, allow sessions to transmit packets even though the sessions are in a non-perfect channel state (hence, effort is very likely not equal to outcome). To illustrate the different design philosophies of these existing algorithms, we describe CIF-Q, ELF, and proportional fair (PF) in detail below.

2.3.2. Channel-condition Independent Fair (CIF)

In [19], Ng, Stoica, and Zhang define the notion of Channel-condition Independent Fair (CIF) in the following manner. To achieve CIF, a packet fair queueing algorithm should provide:

- delay and throughput guarantees for error-free sessions;
- long term fairness for error sessions;
- short term fairness for error-free sessions; and
- graceful degradation for sessions that have received excess service.

The long term fairness for error sessions can be interpreted as: a *lagging* session (a session that cannot realize its requested service rate) which enjoys an error-free channel after some time will be guaranteed to catch up with a certain time bound. The short term fairness for error-free sessions can be interpreted as: between any two error-free sessions that are in the same status (both are leading, or both are satisfied, or both are lagging), the difference of the normalized amount of service they received is bounded for some short time interval [19].

This notion of fairness is adapted to the wireless environment as it takes the non-perfect channel states (i.e., burst errors occur) into consideration. It allows the scheduler to delay the service of a session if it does not have a good channel state, so long as it can catch up in the long run. We can see that using CIF, because the scheduler does not allow a session having "intermediate" (not so good or not so bad) channel states to transmit, effort is equivalent to outcome. However, this scheduling philosophy is unsuitable in a practical environment, where we can expect that a significant portion of sessions will have intermediate channel states. This can be further explicated in the following simple example.

To ease the discussion let us use a simplified channel quality model: suppose there are five possible channel states: A, B, C, D, and E. Accordingly, assume that the effective throughput that can be achieved in different channel states are: in channel state A, 100% of the maximum bandwidth can be realized; in channel state B, 75%; in channel state C, 50%; in channel state D, 25%; and in channel state E, 0%. Suppose there is a mobile device currently in channel state B, what should we do to this session when we apply the CIF-Q scheduler? If we treat only channel state A to be "good" in a two-state model and leave all other states as "bad", the session will receive no service. But this is obviously inefficient and unfair because in fact the session can still transmit some (albeit smaller amount) data. If we treat channel state B as a "good" channel state and let the session transmit as usual, some bandwidth will be wasted because there will be discrepancy between the effort and the outcome.

Indeed, the major drawback in CIF-Q is that in the short term, the system gives priority to the sessions which have a good channel state, while all the other error sessions (with possibly different channel states) are just treated as the same – not allowed to transmit. This is in fact unfair among sessions and may increase the average delay of the sessions. Specifically, the CIF-Q algorithm assumes a rather ideal wireless environment, in which the sessions have a high probability of having a good channel state, in that a lagging session can only be guaranteed to catch up when it has an error-free channel afterwards. However, in reality, the time period during which a session can make full use of the bandwidth (i.e., in channel state A) is usually limited; in other words, a session can spend most of the time in channel states between the perfect (channel state A) and the worst (channel state E). An efficient scheduling algorithm should provide fairness and performance guarantee even when the channel states of the sessions are varying among different quality levels.

Moreover, the CIF notion is still not comprehensive enough in that the fairness among sessions suffering from intermediate channel states is not properly handled. We believe that, from the end user's point of view, the behavior (i.e., fairness and performance) of the sessions suffering from bad channel states is of utmost importance because it can indicate the worst case service quality the session can possibly get in the wireless network. This motivates our proposed new notion of fairness as detailed in section 3.

2.3.3. Effort Limited Fairness (ELF)
Using ELF [6], outcome fair is maintained among sessions unless a session has a channel state poorer than a predefined threshold. Among the sessions with channel states higher than the threshold, the normalized amount of time-slots allocated can be quite different. In order to maintain outcome fair, the scheduler allocates more time-slots (i.e., exerts more effort) on a session with a very poor channel state. Thus, outcome fair is maintained at the expense of the system throughput. However, in order to avoid the pathological case that the poor session wastes too much of the system throughput (i.e., despite the great effort, the outcome is still not enough), a "power factor" is used to control the amount of effort exerted on such extremely unlucky sessions. In order to maintain outcome fair, the ELF approach cannot avoid wasting some bandwidth so as to achieve a fair distribution of realized throughput to the sessions with poor channel conditions.

2.3.4. Proportional Fairness (PF)
Recently, designed for HDR (high data rate) services in CDMA systems, *proportional fair* [2,9] (PF) is considered to be a simple yet effective fairness notion. Specifically, based on the utility based concept (utility is defined as a logarithmic function of the rate allocated to a user; because of the convex nature of the logarithmic function, diminishing return is modeled) introduced by Kelly [12], the HDR (downlink) scheduling is performed in a TDMA manner (i.e., only exactly one user is selected for high data rate transmission in each burst session) and without power water filling. At the scheduling time (time $= t$), suppose a session i has an average realized throughput $H_i(\tau)$ over a past time window of length τ (i.e., from time $= t - \tau$ to t), and the real throughput that can be achieved by session i at time t is $\lambda_i(t)$, which is the aggregate rate of a certain number of supplemental channels (SCH) determined by the base-station according to the interference and power limits. A scheduler is said to achieve proportional fair if it selects for transmission the session with the largest value of

$$\frac{\lambda_i(t)}{H_i(\tau)}. \tag{4}$$

Furthermore, the proportional fairness notion has the nice property that a proportional fair allocation cannot be replaced by any other arbitrary allocation that does not lead to a reduced aggregate fractional rate change.

It should be noted that a proportional fair scheduler heuristically tries to balance the services (in terms of outcome) of the sessions, while implicitly maximizing the system throughput in a greedy manner. Obviously, the proportional fairness notion is a purely outcome fairness metric. Thus, while such a metric is simple to use, proportional fairness does not guarantee fairness in a strict sense. For example, consider the situation where a session has experienced a prolonged period of poor channel states (hence, has a small value of $H_i(\tau)$), it may not get service even though its channel condition improves (e.g., with a moderately large value of $\lambda_i(t)$) if there is a "dominant" session which has a very good channel state (i.e., a very large value $\lambda_i(t)$). Furthermore, the delay experienced by sessions can also be uncontrollably high in a proportional fair system.

3. Channel-adaptive fair queueing

3.1. Overview

Our proposed algorithm is called Channel-Adaptive Fair Queueing (CAFQ) which has the following distinctive features:

- a new notion of fairness is employed;
- contrary to CIF-Q, graceful degradation is not ensured to help the lagging session more efficiently;
- a *punish factor* is used to decide how seriously the scheduler punishes a non-perfect channel state session that transmit packets (thus, the notion of "punishment" is defined with respect to the goal of maximizing overall system throughput); and
- a virtual compensation session is incorporated to help the lagging sessions to catch up.

We believe that, from the user's viewpoint, fairness should be maintained in that so long as a session can transmit some data, it should be provided with some chance to transmit. At the same time, QoS should also be met. However, from the system manager's viewpoint, it is hard to meet these two sometimes conflicting goals with a limited bandwidth and channels that have time-varying quality. Because whenever a session without a perfect channel state is allowed to transmit, there will be part of the bandwidth wasted, and the wasted bandwidth can never be replenished. It should be noted that this is very different from the idea of swapping sessions that are error-free and error-prone, as in existing scheduling algorithms such as CIF-Q. When an error-free session takes the opportunity of an error-prone sessions, it will relinquish the service when the error-prone one is in a good channel state.

Indeed, if abundant bandwidth is available or the channel state is most likely to be perfect, we should maintain the graceful degradation, and prevent the leading sessions from starving. But in a realistic system in which the channel is usually not so good, we cannot expect to achieve perfect allocations, but rather we should meet the sessions QoS first. Thus, in our proposed CAFQ algorithm, graceful degradation is not implemented and the rationale is to compensate the lagging sessions as soon as possible so as to quickly resume a higher throughput and to reduce the delay.

3.2. Channel model

Specifically, the channel condition of a particular mobile device is governed by two components: namely the *fast fading* component and the *long-term shadowing* component. Fast fading is caused by the superposition of multipath components and is therefore fluctuating in a very fast manner (on the order of a few msec). Long-term shadowing is caused by terrain configuration or obstacles and is fluctuating only in a relatively much slower manner (on the order of one to two seconds). To illustrate, a sample of measured fading signal is shown in figure 1.

Let $c(t)$ be the combined channel fading which is given by $c(t) = c_l(t)c_s(t)$, where $c_l(t)$ and $c_s(t)$ are the long-term and short-term fading components, respectively. Both $c_s(t)$ and $c_l(t)$ are random processes with a *coherence time* (time separation between two uncorrelated fading samples) on the order of a few milli-seconds and seconds, respectively.

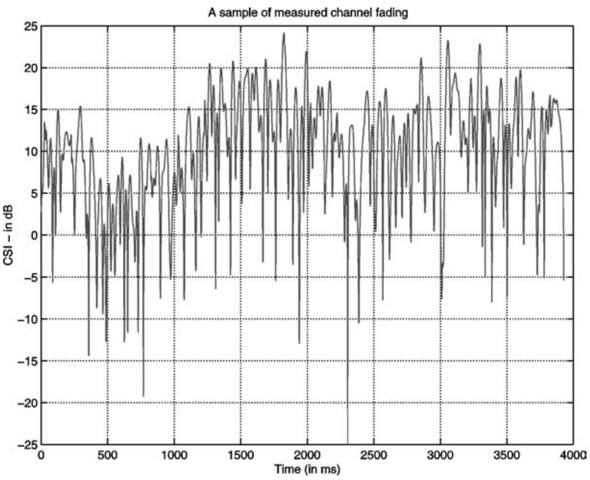


Figure 1. A sample of channel fading with fast fading superimposed on long-term shadowing.

Short-term fading. Without loss of generality, we assume $\mathcal{E}[c_s^2(t)] = 1$ where $\mathcal{E}[\cdot]$ denotes the expected value of a random variable. The probability distribution of $c_s(t)$ follows the Rayleigh distribution which is given by $f_{c_s}(c_s) = c_s \exp(-c_s^2/2)$. In this paper, we assume the mean and maximum speeds of the mobile device are 50 km/hr and 80 km/hr, respectively. Thus, the Doppler spread [21], $f_d \approx 100$ Hz. It follows that the coherence time, denoted by T_c, is approximately given by $T_c \approx 1/f_d$, which is about ten msec.

Long-term fading. The long-term fading component, $c_l(t)$, is also referred to as the *local mean* [21], which, as shown by field test measurement, obeys the *log-normal* distribution, $f_{c_l}(c_l)$. That is, we have:

$$f_{c_l}(c_l) = \frac{4.34}{\sqrt{2\pi}\sigma_l c_l} \exp\left(-\frac{(c_l(\mathrm{dB}) - m_l)^2}{2\sigma_l^2}\right), \tag{5}$$

where m_l, σ_l are respectively the mean (in dB) and the variance of the log-normal distribution, i.e., $c_l(\mathrm{dB}) = 20 \log c_l$. Since $c_l(t)$ is caused by terrain configuration and obstacles, the fluctuation is over a much longer time scale. Again, from field test results, the order of time span for $c_l(t)$ is about one second. Since mobile devices are scattered geographically across the cell and are moving independently of each other, we assume the channel fading experienced by each mobile device is independent of each other.

As usual, redundancy is incorporated to the information packet for error protection. To exploit the time-varying nature of the wireless channel, a variable-throughput channel-adaptive physical layer is employed as illustrated in figure 2. Channel state information (CSI), $c(t)$, which is estimated at the receiver, is fed back to the transmitter via a low-capacity

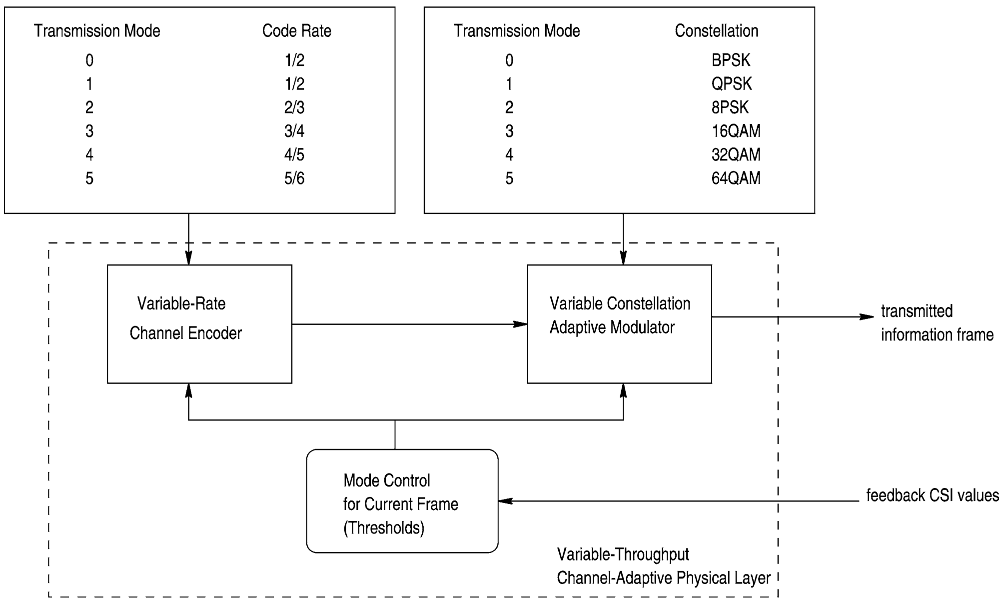


Figure 2. A conceptual block diagram of the variable-throughput channel adaptive physical layer.

feedback channel. Based on the CSI, the level of redundancy and the modulation constellation applied to the information packets are adjusted accordingly by choosing a suitable transmission mode. Thus, the instantaneous throughput is varied according to the instantaneous channel state. In contrast to existing work in wireless packet scheduling that uses simple channel model, in our study [25], a 6-mode variable-throughput adaptive bit-interleaved trellis coded modulation scheme (ABICM) is employed [14]. Transmission modes with *normalized throughput*[1] varying from 1/2 to 5 are available depending on the channel condition.

We assume the coherence time of the short-term fading is around ten msec which is much longer than an information slot duration. Thus, the CSI remains approximately constant within a frame and it follows that the transmission mode for the whole frame is determined only by the current CSI level. Specifically, transmission mode q is chosen if the feedback CSI, $\hat{c}$, falls within the *adaptation thresholds*, (ζ_{q-1}, ζ_q). Here, the operation and the performance of the ABICM scheme is determined by the set of adaptation thresholds $\{\zeta_0, \zeta_1, \ldots\}$. In this paper, we assume that the ABICM scheme is operated in the *constant BER* mode [14]. That is, the adaptation thresholds are set optimally to maintain a target transmission error level over a range of CSI values. When the channel condition is good, a higher mode could be used and the system enjoys a higher throughput. On the other hand, when the channel condition is bad, a lower mode is used to maintain the target error level at the expense of a lower transmission throughput. Note that when the channel state is very bad, the adaptation range of the ABICM scheme can be exceeded such that the throughput (mode-0) becomes very low, making it impossible to maintain the targeted BER level. This adverse situation is illustrated in figure 3(a).

Given the above considerations about the channel state, the instantaneous throughput offered to the access control layer, denoted by ρ, is also variable and is therefore a function of the CSI, $c(t)$, and the target BER, P_b, denoted by $\rho = f_\rho(c(t), P_b)$. Figure 3(b) illustrates the variation of ρ with respect to the CSI.

3.3. Channel-adaptive fairness

We propose a new notion of fairness to be maintained in the short term, called *channel-adaptive fairness* (CAF). Specifically, a scheduler is channel-adaptive fair if in the short term the difference between the normalized throughput (normalized with respect to the channel capacity) of any two backlogged sessions i and j is bounded as follows:

$$\left| \frac{T_i(t_1, t_2)}{r_i f(\Phi_i)} - \frac{T_j(t_1, t_2)}{r_j f(\Phi_j)} \right| < \epsilon, \tag{6}$$

where Φ_i denotes the channel state (e.g., one of the five classes A, B, C, D, and E), and $f(\Phi_i) = M(\Phi_i)^\eta$ in which $M(\Phi_i)$ is the *effective throughput factor* $(0 \leqslant M(\Phi_i) \leqslant 1)$.

[1] Normalized throughput refers to the number of information bits carried per modulation symbol.

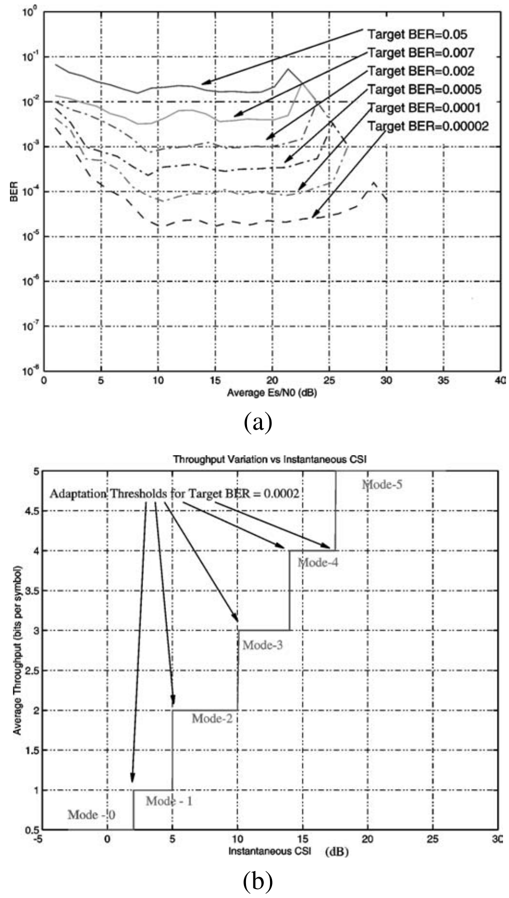


Figure 3. BER and throughput of ABICM scheme. (a) Instantaneous BER and the adaptation range. (b) Instantaneous throughput vs. CSI.

The effective throughput factor is channel state dependent: $M(\Phi_i) = 0.75$ if Φ_i is channel state B, and so on. Here, η is a *punish factor* which is a positive number. Thus, in our definition of fairness, the throughput a session receives will be proportional to its channel quality. And, in the long term, outcome fair is maintained among all sessions.

Our proposed fairness is more reasonable in the wireless environment because it considers explicitly the different channel states. Unlike the CIF-Q algorithm that prevents the sessions without perfect channel state from transmitting and unlike the ELF algorithm that distributes the normalized amount of service inversely proportional to their channel states, a CAF scheduler provides transmission opportunities to all sessions that do not suffer from the worst channel state in the short term, and at the same time, it punishes the sessions without good channel states to different extent. Furthermore, unlike the proportional fair scheduler, using the CAF scheduler does not necessarily schedule the session with the best channel condition to transmit first. With the channel-adaptive fairness, we can formalize a new fair queueing algorithm, which is explicated in detail in the following section.

The punish factor η can help to decide between to make use of the bandwidth more efficiently and to treat every session more fairly. When a larger value of punish factor is used, we punish the non-perfect channel state session that transmits packets more seriously, and prevent them from wasting too

Table 1
Qualitative comparison of fairness notions.

Fairness	Short-term	Long-term
CIF	Short term fairness is maintained only among sessions with perfect channel states; neither outcome fair nor effort fair is considered for sessions with "not so good" channel states	Outcome fair provided that the sessions are under homogeneous error characteristics in the long run
ELF	Outcome fair is maintained among sessions with channel states better than a predefined threshold	Not precisely defined
PF	Short term fairness is not precisely maintained	Not precisely defined
CAF	Short term fairness (normalized by channel states) is maintained among all the sessions unless the session has the worst channel state; a compromise is achieved in attaining outcome fair and in attaining efficient bandwidth usage	Outcome fair provided that the sessions are under homogeneous error characteristics in the long run

much bandwidth. In effect, the bandwidth is used more efficiently, and the average delay of the total system is decreased and the throughput is increased. But if there is a session that is more unlucky than the others and has a higher probability of having a bad channel state, its average delay and throughput may be very bad, because it is punished seriously and prevented from occupying the bandwidth. When a smaller punish factor is used, this kind of unlucky sessions will be punished only moderately, so the average delay and throughput of these sessions are reduced. But as they have more chance to access the bandwidth and hence incur a larger wastage of bandwidth, the total throughput and average delay of the system will be adversely affected. Thus, the punish factor can be used to tune the utilization of system resources.

3.4. Comparison with other fairness notions

Having defined our proposed fairness notion, it is useful to compare it with other existing fairness notions, as shown in table 1.

3.5. Detailed description of CAFQ

As in existing algorithms, we associate the scheduling system with an error-free system to account for the service lost or gained by a session due to errors. A session is classified as leading or non-leading depending on the difference of the service it received between the error-free system and the real one. A session is leading if it has received more service in the real system than in the error-free one, while it is non-leading if it has received less or the same amount.

We simulate SFQ (Start-time Fair Queueing) [8] in the error-free system in our study for the reason of simplicity because it is hard to schedule according to the finish times of the packet in the wireless environment. In the SFQ, when packet k of session i arrives, it is stamped with a virtual start time $S(P_{i_k})$, computed as

$$S(P_{i_k}) \leftarrow \max\bigl\{V\bigl(A(P_{i_k}), F(P_{i_{k-1}})\bigr)\bigr\}, \tag{7}$$

$$F(P_{i_k}) \leftarrow S(P_{i_k}) + \frac{l_{i_k}}{r_i}, \tag{8}$$

where P_{i_k} is the kth packet of session i, $F(P_{i_k})$ is the virtual finish time of packet P_{i_k}, $V(A(P_{i_k}))$ is the virtual clock of the system at the arrival time $A(P_{i_k})$ of the packet, r_i is the pre-allocated service share of session i, and l_{i_k} is the length of the packet. The virtual time of the packets are initialized to zero. In the error-free system, a session i is selected in the increasing order of the sessions virtual starting times among sessions that are backlogged. Since it is possible that the packet of another session instead of session i will be transmitted in the real system, a session's virtual time only keeps track of the normalized service received by the session in the error-free system.

Another parameter, Δ, is used to keep track of the difference of the service a session received in the real system and in the error-free one. The Δ of a session is initialized to zero. A session is non-leading if Δ is greater than or equal to zero, while it is leading if Δ is less than zero.

In CAFQ, fairness is maintained in two aspects: in the short term, CAF is maintained among the leading sessions and non-leading sessions separately unless the sessions have the worst channel state (cannot transmit). In the long term, outcome fair is ensured with the help of a virtual compensation session.

3.5.1. Short term fairness

We introduce two parameters N and L to implement the channel-adaptive fairness in the short term. N_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is non-leading. When a session i becomes both non-leading and not suffering from the worst channel state, N_i will get initialized as follows:

$$\max\Bigl\{N_i, \min_{k\in\Psi}\bigl\{N_k \mid \mathrm{lag}_k \geqslant 0\bigr\}\Bigr\}, \tag{9}$$

where Ψ denotes the set of sessions that are backlogged and for a non-leading session chosen to transmit packets in the real system, the N_i is updated as follows:

$$N_i \leftarrow N_i + \frac{l_i}{r_i f(\Phi_i)} \tag{10}$$

and L_i is defined similarly. Here, L_i keeps track of the normalized amount of services received by session i which is proportional to its channel state function when it is leading. When a session i becomes both leading and not suffering from the worst channel state, L_i will get initialized in a way analogous to (9).

In the real system, selection is made among the non-leading ones first. The session with the minimum N_i will be selected, and the packet at the head of the waiting queue of this session will be transmitted and N_i will be updated accordingly. If there is no such kind of session which is non-leading and backlogged, the system will select from the leading ones in the increasing order of the sessions' L_i, and then L_i will be updated accordingly. If all sessions are not backlogged (a very unlikely situation in a mobile computing system with a reasonable number of active users), dummy packets will be sent. If the session j selected in the real system is not the one chosen in the error-free one and it is i that is selected in the error-free system, the Δ of i and j will both be updated: $\Delta_i \leftarrow \Delta_i + l_i$, $\Delta_j \leftarrow \Delta_j - l_j$; otherwise, the Δ will not be changed.

When a session with a comparatively bad channel state transmits packet, the N_i or L_i will increase more rapidly than a session with a better channel state. As the punish factor changes, we can decide how seriously we should punish a session which does not have a perfect channel and transmits packets. The larger the punish factor is, the more seriously we punish the unlucky sessions.

Let us consider a simple example. Suppose session i has channel state 0.75 (i.e., class B), session j has channel state 0.25 (i.e., class D), and both are non-leading and have the same service rate, and all packets are of the same length l. If the punish factor is 1, then after both of them transmit one packet, N_i increases by $1.33 \cdot l/r_i$, N_j increases by $4 \cdot l/r_j$. The reason why N_j increases much more than N_i is that j has a much poorer channel state than i, while both of them get one packet transmitted. After that, j has less chance to transmit because of the large N_j, so what the system does is to give j the chance to transmit, but punishes it because it wastes the bandwidth. If we change the punish factor to 0.5, then N_i increases by $1.15 \cdot l/r_i$ after i transmits one packet, N_j increases by $2 \cdot l/r_j$ after that. So, as expected, j is punished only moderately as the punish factor decreases.

3.5.2. Long term fairness

Nonetheless, there is still one issue to be considered: although the sessions, which do not have perfect channel states but get packets transmitted, are punished, they are given some chance to transmit, and part of the bandwidth of the system is wasted and can never get compensated. Because the scheduler will not schedule a leading session to transmit if there is a lagging one which is backlogged and is not in the worst channel state (i.e., state E), the scheduler will not save the effort of the system as most of the other scheduling algorithms do. So, we assign a service share to a virtual compensation session to help in the long term. This pre-allocated service share is used to help the lagging ones with perfect channel state, because only when a session has a perfect channel state, can it get compensation most efficiently. When a lagging session exits from non-perfect channel states, its session ID will be queued in the virtual compensation session. Sessions that are queued in the virtual compensation session are in the decreasing order of their Δ. So we give bonus service to the lagging sessions if it has perfect channel state, and the session which lags most will get it first so that it can be helped to catch up, and thus, long term outcome fair can be maintained.

The design of the virtual compensation session is quite different from that in SBFA [22]. In SBFA, only sessions with perfect channel states will be scheduled, while sessions with bad channel states will only get their transmission opportunities back from the pre-allocated bandwidth. Thus, compensation is implemented by using the pre-allocated bandwidth instead of using the excess services as in CIF-Q and IWFQ. In CAFQ, however, compensation is not implemented purely by such swapping because some of the bandwidth wasted by the poor sessions may not be recovered. Thus, the virtual compensation session in CAFQ is for compensating the wasted bandwidth such that outcome fair can be maintained in the long term.

In the error-free system, we select a session i among all the backlogged sessions and the virtual compensation session in the increasing order of the virtual time. If it is the virtual compensation session that is selected and there is session ID waiting in the queue, the session with the ID at the head of the virtual compensation queue will be scheduled to transmit in the real system. The Δ of this session will be decreased as $\Delta_i \leftarrow \Delta_i - l_i$. If it is not the virtual compensation session that is selected or there is no session ID waiting in the queue, the system will select a session to transmit in the real system from the non-leading ones according to N_i, then from the leading ones according to L_i if there is no non-leading one to take the service as we have mentioned above.

Using the proposed CAFQ algorithm described above, the leading sessions may possibly be starved because we always select from the non-leading ones first. The reason why we do not maintain graceful degradation in CAFQ as in CIF-Q and WFS do is that we want to maximize the performance of the non-leading ones first. One of the most important goals of scheduling algorithms is to meet the QoS of the sessions, so CAFQ will not help a session that has achieved the same amount of service as it should have in the error-free system, if there are lagging sessions that have not had their QoS fulfilled.

Due to space limitations, the pseudo-code of the proposed CAFQ algorithm is not listed here but can be found in [25, pp. 57–58].

3.6. Comparison with CIF-Q

The design of using N_i and L_i on the surface is similar to the using of C and F counters in the CIF-Q algorithm in that the counters are used for maintaining short term fairness. However, there are some important differences:

- Firstly, different kinds of short term fairness are maintained in CIF-Q and CAFQ; in the former, fairness is maintained for two error-free sessions provided they are in the same state (both lagging or both leading), while in the latter, fairness is maintained for all sessions (unless a session is suffering from the worst channel state).
- Secondly, different amount of bandwidth is distributed by using the counters; in CIF-Q, "additional" service is distributed using the C and F counters, while in CAFQ, all

the bandwidth is distributed and there is no additional service notion.

4. Numerical results

In this section, we illustrate the functionality of our proposed CAFQ algorithm using several scenarios to generate numerical results, based on which we quantitatively compare the CAFQ algorithm with other algorithms (CIF-Q and CS-WFQ). Due to space limitations, more detailed results are not included in the paper and can be found in [25].

4.1. Parameters

We make the following assumptions in the scenarios:

- all the packets are of the same length of 1000 bits;
- the bandwidth of the system is 4 Mbps;
- the punish factor (η) is set to be 1 if not otherwise stated;
- each mobile device moves in a straight line with a speed of 20 km/hr in random directions within a field of 200 m $\times$ 200 m (a new random direction is generated when the device hits the boundary); the fast fading and shadowing effects are thus computed based on the instantaneous geographical locations;
- the parameter α of CIF-Q [19] is set to be 0.1; and
- in CS-WFQ the thresholds for all sessions is $1/3$ so when a session has a channel state poorer than state C (corresponds to $1/3$ as elaborated below), it will not be scheduled.

To ease understanding the scenarios, we employ a slightly simpler physical layer model. Specifically, we first use a 4-mode ABICM adaptive channel coder and modulator to generate the channel error statistics[2] together with the corresponding throughput values (as governed by the desired modulation and coding modes selected, according to the SINR perceived). Using these simulation results, we abstract the relationship between channel state and error mode, as well as effective throughput using a Markov model [26,27]. Using such a discrete Markov model allows us to trace the operations of the algorithms easily. In our study, we normalize the effective throughput of the four states (corresponding to the four ABICM operating modes; denoted by A, B, C, and D) to the maximum one, i.e., the effective throughput in channel state A is 1; the effective throughput in channel state B is $2/3$; the effective throughput in channel state C is $1/3$; and the effective throughput in channel state D is 0 (data loss is too high). The better the channel quality, the higher the effective throughput. In other words, the session that is in channel state A channel can make use of 100% of the allocated bandwidth; the session under channel state B channel can make use of $2/3$ of the allocated bandwidth; the session under channel state C can make use of $1/3$ of the allocated bandwidth; the session under channel state D cannot transmit any data effectively.

Table 2
Channel states and error modes.

Channel state	Error mode A	B	C	D
1	0.44	0.49	0.054	0.016
2	0.38	0.48	0.12	0.02
3	0.31	0.44	0.19	0.06
4	0.24	0.40	0.27	0.09
5	0.17	0.33	0.34	0.16

We identify 5 kinds of error modes as they differ in the steady-state probability shown in table 2 (we would like to emphasize that such a hypothetical error model is used merely for illustrative purposes). As we can see that error mode 1 has the best overall channel state, while error mode 5 has the worst overall channel state. In each simulation run, the sessions begin with different channel states selected at random.

4.2. Scenario 1

We simulate CIF-Q and CAFQ with the environment as follows. The service shares of the two sessions are both 0.5 in CIF-Q. The virtual compensation session has a service rate of 0.1, and the sessions rates are 0.45. Both of the sessions are continuously backlogged. Session 1 has error-free channel state all along (i.e., $M(\Phi_1)$ is always equal to 1), while the channel state of session 0 changes periodically as follows ($k = 0, 1, 2, 3$):

- $M(\Phi_0) = 1$ when $8k \leqslant t \leqslant 8k + 2$;
- $M(\Phi_0) = 2/3$ when $8k + 2 \leqslant t \leqslant 8k + 4$;
- $M(\Phi_0) = 1/3$ when $8k + 4 \leqslant t \leqslant 8k + 6$; and
- $M(\Phi_0) = 0$ when $8k + 6 \leqslant t \leqslant 8k + 8$.

We keep track of the difference between the expected service and the actual service of these two sessions (denoted it by β) changing in the two algorithms. The result is shown in figure 4.

For CIF-Q, the β of session 0 increases at the rate of 2 Mbps, while the β of session 1 decreases at the same rate during the period when session 0 does not have a perfect channel state. This is because session 1 occupies the bandwidth to transmit when session 0 cannot transmit due to its channel state. After session 0 has perfect channel state, it will be compensated and the β decreases at the rate of 1.8 Mbps. At the same time, the β of session 1 increases at the rate of 1.8 Mbps. Using the simple periodic error pattern, it suffices to show that when the session has a comparatively high probability of not having perfect channel state, it is possible that its β grows very fast. So session 0 is seriously lagged when session 1 is seriously leading. It is because CIF-Q tries to make use of the bandwidth greedily, and will not save session 0 at the expense of wasting bandwidth.

The CAFQ algorithm works better because it always allows session 0 to transmit provided that the session is not

[2] Note that the usage of the ABICM scheme [14] is just for illustration purposes only; other adaptive physical layer schemes, such as MQAM [7] can also be used with our scheduler.

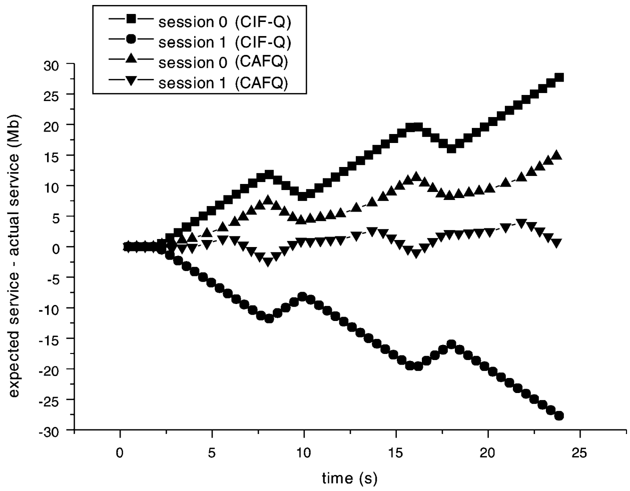


Figure 4. Variations of Δ in CIF-Q and CAFQ.

under the worst channel state (i.e., state D). Meanwhile, session 0 is also punished (as governed by η) for transmitting data under an error-prone channel (hence, induces losses and the wasting of bandwidth). So session 1 will not lead as much as it will do in CIF-Q, neither will session 0 lag that much. The only drawback is that session 1 is sometimes affected as its β exceeds 0 occasionally. But the amount of β that exceeds is not large, and it gets recovered soon. This is because we always try to help the non-leading one first, and the leading one will be ignored until it gets lagged. In summary, we find that:

- the long term fairness of CIF-Q may not be maintained when there is a comparatively high probability for a session to have non-perfect channel state; and
- the CAFQ algorithm helps the non-leading session more efficiently.

4.3. Scenario 2

In this scenario, the simulation time is 500 seconds and we computed the average result over 10 simulation runs. We simulate CS-WFQ and our CAFQ algorithm under 5 kinds of error modes. There are 3 sessions in the system. The preallocated service rates of them are: 0.25, 0.25, 0.5 in CS-WFQ. The virtual compensation session has the rate of 0.2, and the other sessions rates are: 0.2, 0.2, 0.4 in the CAFQ algorithm. The data source of the sessions are Poisson sources with the arrival rates as: 0.8 Mbps, 0.8 Mbps, and 1.6 Mbps. All the sessions in the system have the same kind of error mode in each simulation run.

We calculate the average delay, max delay of all the session and the system throughput both in the CAFQ algorithm and in CS-WFQ when the error mode changes. The results are shown in figure 5.

As can be seen, the average delays and maximum delays increase, and the throughputs decrease as the overall channel state becomes worse both in CS-WFQ and in CAFQ. But the rate of increase (decrease) is slower in the CAFQ algorithm, and the average delays and the maximum delays are always

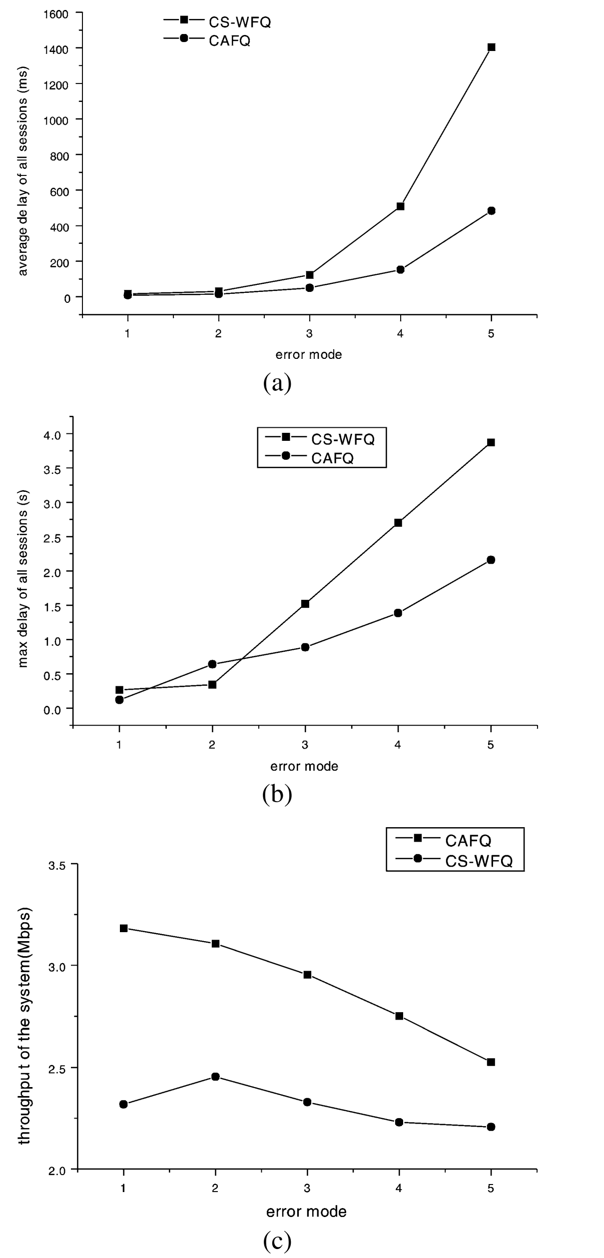


Figure 5. Comparison between CS-WFQ and CAFQ. (a) Average delay. (b) Maximum delay. (c) Throughput.

smaller in the system with the CAFQ algorithm than in the system with CS-WFQ, while the throughputs in the CAFQ algorithm are always higher than in CS-WFQ. It is because CS-WFQ wastes the bandwidth seriously by maintaining outcome fair within the effort limit. Thus, fewer packets can be transmitted in a given time period, and packets have to wait for a longer time before they get transmitted so that the average delays and maximum delays grow. On the contrary, the CAFQ algorithm grants the session without perfect channel state to transmit at the same time of punishing them, so they have chance to transmit, but the chance is less if it has worse channel state.

In the short term, CAF is maintained and thus, the sessions' need for maintaining outcome fairness is handled.

At the same time, efficient utilization of bandwidth is also achieved such that the precious bandwidth is not wasted to desperately maintain outcome fair. In the long run, virtual compensation session helps the session which lags most seriously and has perfect channel state. This helps to reduce the average and maximum delays.

4.4. Scenario 3

Simulation time and bandwidth of the system are the same as in scenario 2. There are three sessions in the system with rate 0.2, 0.2, 0.4, and the sources are Poisson source with arrival rate 1 Mbps, 1 Mbps, and 2 Mbps, respectively. The channel error model is a simple four-state Markov chain. The steady state probability of session 0 and session 2 is error mode 1, and the channel state probability distribution of session 1 follows that of error mode 5 as defined in table 2. We keep track of the throughput and the maximum delay of the sessions. The results are shown in figure 6.

When the punish factor increases, the system punishes the session that experiences a poor channel condition but uses significant bandwidth. Thus, when a session has a poor channel state, it has less chance of getting its packets transmitted. As the punish factor increases, the packets are more likely to get backlogged in the queue and that is the major cause of the increase in maximum delay as shown in figure 6(a). Furthermore, sessions with poor channel conditions have less chance to waste bandwidth (i.e., unable to deliver the desired outcome given the effort allocated), and thus, the system throughput improves as the punish factor increases, as illustrated in figure 6(b).

In figure 6(c), we can see that the throughput of sessions 0 and 2 increase as the punish factor increases. More importantly, we can see that the two curves are very close to each other, demonstrating that the CAFQ treats sessions with similar channel conditions in a fair manner. Note that under CIF-Q, the throughputs of the three sessions are depicted as horizontal lines independent of the punish factor.

4.5. Scenario 4

It is also interesting to examine the performance of the CIF-Q and CS-WFQ algorithms when our proposed new notion of fairness (channel-adaptive fairness) is incorporated in them.

There are three sessions in the system. The pre-allocated service rates of them are: 0.2, 0.2, and 0.4, respectively. The pre-allocated service share for the virtual compensation session is 0.2. The sources are the Poisson sources with arrival rates 0.8 Mbps, 0.8 Mbps, 1.6 Mbps. The sessions channel model is a four-state Markov chain. The steady state probability of session 1 is as error mode 1, and the remaining sessions are as error mode 2. We calculated the throughput and the average delay of the system if the system runs the original CIF-Q or CS-WFQ algorithm. The modified algorithms (with channel adaptive fairness incorporated), called CIF-Q* and CS-WFQ*, are also tested in the same environment. The results are shown in figure 7.

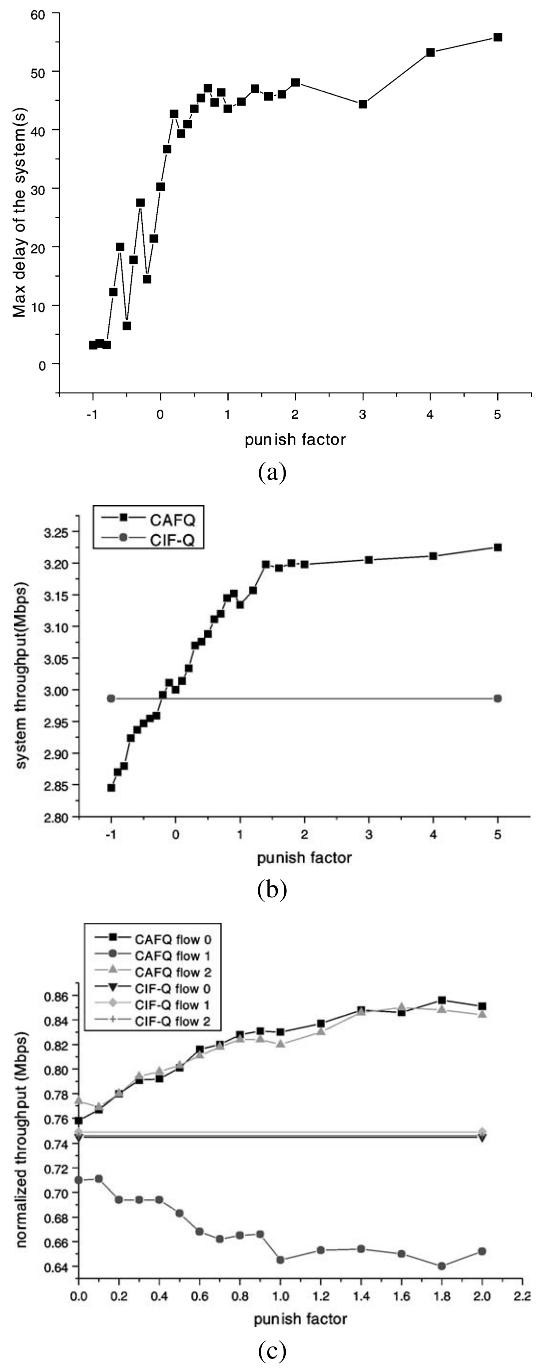
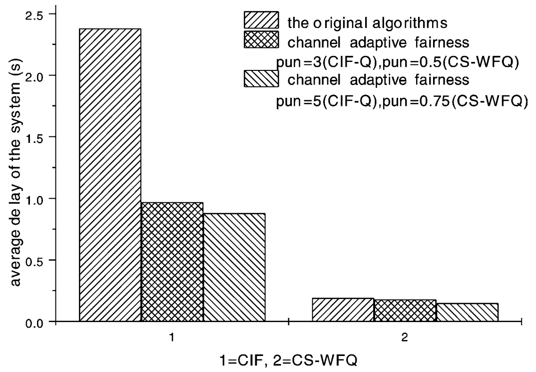


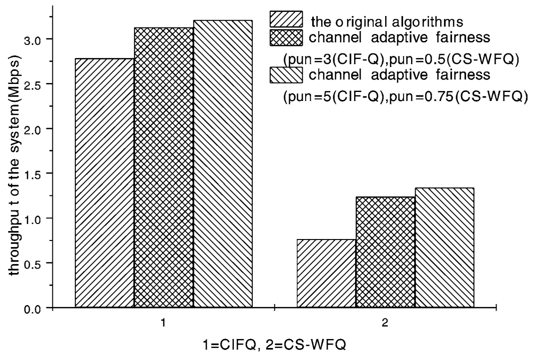
Figure 6. Performance of the CAFQ algorithm with various values of punish factor. (a) Maximum delay. (b) System throughput. (c) Normalized throughput.

We simulate the CIF-Q* twice when the punish factor is 3 and 5 and find that the throughput and the average delay are improved more if the punish factor is bigger. This is because we punish the sessions that access the bandwidth by improving C_i or F_i more if they experience worse channel states. So the bandwidth is protected from being wasted by the unlucky sessions with worse channel states, and thus, the throughput and the average delay of the system are improved.

We find a similar situation in CS-WFQ*. As session 1 always perceives a better channel than session 2, let us con-



(a)



(b)

Figure 7. Performance of CIF-Q and CS-WFQ with CAF incorporated. (a) Average delay. (b) Average throughput.

sider a time instance when session 1 has a perfect channel state, while session 2 can only make use of $1/3$ of the allocated bandwidth. In CS-WFQ, the bandwidth ratio is $1/3$. Although CS-WFQ can ensure the same normalized amount of throughput at the devices, it is unfair to session 1. Session 1 has the same service share as session 2, but it is allocated much less bandwidth than session 2, so in fact, session 1 is punished by the system because session 2 has a poor channel state. In CS-WFQ*, if the punish factor is 0.5, the bandwidth is as follows: $(1/3)^{0.5} = 0.577 > 1/3$. Comparing with the situation in the rate proportional GPS, it improved the throughput at the device of session 2, as it tried to leave more effort on session 2 at the base-station. Although it wastes part of the bandwidth, it helps session 2 to have a better QoS which is nearer to the pre-determined one despite of the bad channel state. Comparing with the situation in the CS-WFQ, it reduces the wastage of bandwidth too drastically at the same time of helping session 2. When the punish factor is 0.75, the bandwidth ratio is now $(1/3)^{0.75} = 0.76 > 0.577 > 1/3$. Comparing with the situation when the punish factor is 0.5, less bandwidth is granted to session 2 to avoid wasting the bandwidth, although it still allocated more bandwidth to help session 2 compared with GPS. So the throughput and average delay of the system is further improved. The price to be paid is that when the punish factor is larger, the session with worse channel state achieves less throughput at the device than its pre-allocated share.

5. Concluding remarks

In this paper, we have presented a qualitative and quantitative analysis of different fair queueing scheduling algorithms in wireless networks. Because of the time-varying nature of the wireless channel in a practical situation, burst errors are the norm rather than an exception and, thus, we believe that a good scheduling algorithm should take into consideration, or even exploit, the variations of channel conditions among the mobile devices. In this regard, we propose a new notion of fairness in which a scheduler is fair with respect to the throughput normalized by the channel capacity. Using this new fairness definition, we propose a new scheduling algorithm called CAFQ (Channel Adaptive Fair Queueing). In our numerical results, we have demonstrated that the CAFQ algorithm can balance the often times conflicting goals of maintaining fair service and maximizing overall system throughput.

There are several possible avenues of further research. The proposed CAFQ algorithm is a centralized approach. It would be more practicable if we could devise a distributed implementation such that both the base-station and the mobile devices contribute in the scheduling process, in a manner similar to the algorithm suggested by Kelly [12]. Furthermore, aided by the information theoretic understanding of the throughput capacity of a multi-access fading channel [24], it would be interesting to study the theoretical behaviors of the CAFQ algorithm for scheduling multimedia data transmission [1] in a CDMA system which involves several more system dimensions (e.g., interference, power control, soft handoff, etc.).

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful and constructive comments. Thanks are also due to Professors Albert Zomaya and Mohan Kumar for their professional handling of our manuscript and kind suggestions. This research is supported by HKU URC seed grants under contract numbers 10203010 and 10203413, and by a RGC research grant under contract number HKU 7024/00E.

References

[1] I.F. Akyildiz, D.A. Levine and I. Joe, A slotted cdma protocol with BER scheduling for wireless multimedia networks, IEEE/ACM Trans. Networking 7(2) (1999) 146–158.

[2] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi, CDMA/HDR: A bandwidth efficient high-speed wireless data service for nomadic users, IEEE Communications Magazine 38(7) (2000) 70–77.

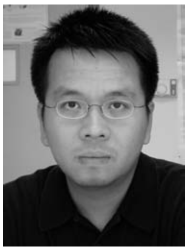
[3] P. Bhagwat, P. Bhattacharya, A. Krishna and S.K. Tripathi, Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs, ACM/Baltzer Wireless Networks 3(1) (1997) 91–102.

[4] V. Bharghavan, S. Lu and T. Nandagopal, Fair queuing in wireless networks: Issues and approaches, IEEE Personal Communications 6(1) (1999) 44–53.

[5] Y. Cao and V.O.K. Li, Scheduling algorithms in broadband wireless networks, Proceedings of the IEEE 89(1) (2001) 76–87.

[6] D.A. Eckhardt and P. Steenkiste, Effort-Limited Fair (ELF) scheduling for wireless networks, in: *Proc. INFOCOM'2000*, Vol. 3 (2000) pp. 1097–1106.
[7] A.J. Goldsmith and S.-G. Chua, Variable-rate variable-power MQAM for fading channels, IEEE Trans. Communications 45(10) (1997) 1218–1230.
[8] P. Goyal, H.M. Vin and H. Chen, Start-time fair queueing: A scheduling algorithm for integrated services, in: *Proc. ACM SIGCOMM'96* (1996) pp. 157–168.
[9] A. Jalali, R. Padovani and R. Pankaj, Data throughput of CDMA-HDR: A high efficiency high data rate personal communication wireless system, in: *Proc. VTC'2000*, Vol. 3 (2000) pp. 1854–1858.
[10] M.R. Jeong, H. Morikawa and T. Aoyama, Fair scheduling algorithm for wireless packet networks, in: *Proc. ICPP'99* (1999) pp. 280–285.
[11] M. Kang and S. Wilbur, A fair guaranteed down-link sharing scheme for cellular packet switched networks, in: *Proc. GLOBECOM'97*, Vol. 2 (1997) pp. 1006–1010.
[12] F. Kelly, Charging and rate control for elastic traffic, European Transactions on Telecommunications 8 (1997) 33–37.
[13] Y.-K. Kwok and V.K.N. Lau, A quantitative comparison of multiple access control protocols for wireless ATM, IEEE Transactions on Vehicular Technology 50(3) (2001) 796–815.
[14] V.K.N. Lau, Performance analysis of variable rate: Symbol-by-symbol adaptive bit interleaved coded modulation for Rayleigh fading channels, IEEE Transactions on Vehicular Technology 51(3) (2002) 537–550.
[15] P. Lin, B. Bensaou, Q.L. Ding and K.C. Chua, CS-WFQ: A wireless fair scheduling algorithm for error-prone wireless channels, in: *Proc. 9th Internat. Conf. Computer Communications and Networks* (2000) pp. 276–281.
[16] S. Lu, V. Bharghavan and R. Srikant, Fair scheduling in wireless packet networks, IEEE/ACM Trans. Networking 7(4) (1999) 473–489.
[17] J.R. Moorman and J.W. Lockwood, Multiclass priority fair queuing for hybrid wired/wireless quality of service support, in: *Proc. ACM WoWMoM'99* (1999) pp. 43–50.
[18] T. Nandagopal, S. Lu and V. Bharghavan, A unified architecture for the design and evaluation of wireless fair queueing algorithms, Wireless Networks 8(2–3) (2002) 231–247.
[19] T.S.E. Ng, I. Stoica and H. Zhang, Packet fair queueing algorithms for wireless networks with location-dependent errors, in: *Proc. INFOCOM'98*, Vol. 3 (1998) pp. 1103–1111.
[20] A.K. Parekh and R.G. Gallager, A generalized processor sharing approach to flow control in integrated services network: The single-node case, IEEE/ACM Trans. Networking 1(3) (1993) 344–357.
[21] J.D. Parsons, *The Mobile Radio Propagation Channel*, 2nd Edition (Wiley, New York, 2000).
[22] P. Ramanathan and P. Agrawal, Adapting packet fair queueing algorithms to wireless networks, in: *Proc. MOBICOM'98* (1998) pp. 1–9.
[23] A. Stamoulis and G.B. Giannakis, Packet fair queueing scheduling based on multirate multipath-transparent CDMA for wireless networks, in: *Proc. INFOCOM'2000*, Vol. 3 (2000) pp. 1067–1076.
[24] D.N.C. Tse and S.V. Hanly, Multiaccess fading channels – Part I: Polymatroid structure, optimal resource allocation and throughput capacities, IEEE Trans. Information Theory 44(7) (1998) 2796–2815.
[25] L. Wang, Channel-adaptive fair queueing in wireless packet data networks, M.Phil. Thesis, The University of Hong Kong (August 2002), available at `http://www.eee.hku.hk/~ykwok/students/thesis-mphil-WangLi.pdf`
[26] H.S. Wang and P.-C. Chang, On verifying the first order Markovian assumption for a Rayleigh fading channel model, IEEE Trans. Vehicular Technology 45(2) (1996) 353–357.
[27] H.S. Wang and N. Moayeri, Finite-state Markov channel – A useful model for radio communication channels, IEEE Trans. Vehicular Technology 44(1) (1995) 163–171.
[28] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, Proceedings of the IEEE 83(10) (1995) 1374–1396.

Li Wang was born in Nanjin, China in 1978. She received a B.Sc. degree in electrical engineering from the Shanghai Jiao Tong University in 2000. She is now finishing the M.Ph. degree in the Department of Electrical and Electronic Engineering, the University of Hong Kong. Her research interests include packet scheduling algorithms for wireless networks, resource management schemes for CDMA systems, and mobile computing.
E-mail: liwang@eee.hku.hk



Yu-Kwong Kwok is an Associate Professor in the Department of Electrical and Electronic Engineering at the University of Hong Kong. Before joining the University of Hong Kong, he was a visiting scholar in the parallel processing laboratory at the School of Electrical and Computer Engineering at Purdue University. He received his B.Sc. degree in computer engineering from the University of Hong Kong in 1991, the M.Ph. and Ph.D. degrees in computer science from the Hong Kong University of Science and Technology (HKUST) in 1994 and 1997, respectively. His research interests include mobile computing, wireless communications, network protocols, and distributed computing algorithms. He is a Senior member of the IEEE. He is also a member of the ACM, the IEEE Computer Society, and the IEEE Communications Society.
E-mail: ykwok@eee.hku.hk

Wing-Cheong Lau received the B.S. (Eng.) degree from the University of Hong Kong and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin. From 1995 to 1997, he was a Member of Technical Staff with Southwestern Bell Technology Resources, Austin, TX, responsible for broadband network architectural design and performance analysis. Since 1997, he has been a Member of Technical Staff with the Performance Analysis Department, Bell Laboratories, Lucent Technologies, Holmdel, New Jersey. Between November 1999 and June 2001, he was on leave from Bell Labs and taught at the University of Hong Kong where he served as the Associate Director for the Master of Science Programme in E-Commerce and Internet Computing. Currently, he remains as a Visiting Instructor for the programme. Dr. Lau is a Senior Member of IEEE and a Member of ACM and Tau Beta Pi.
E-mail: lau@lucent.com

Vincent K.N. Lau (M. IEEE 1998–2001, S.M. IEEE 2001) obtained a B.Eng. (Distinction 1st Hons) in EE in 1992 from the University of Hong Kong. He joined the HK Telecom after graduation for 3 years as system engineer, responsible for transmission systems design. He was awarded the Sir Edward Youde Memorial Fellowship and the Croucher Foundation in 1995 and obtained a Ph.D. degree in 1997 from the University of Cambridge. He joined the Lucent Technologies – Bell Labs in the US as member of technical staff and was engaged in the algorithm design, standardization and prototype development of CDMA2000 systems. He joined the University of Hong Kong in 1999 as Assistant Professor and was appointed the co-director of Information Engineering Programme as well as the co-director of 3G Technology Center. On July 2001, he left the University and returned to the Wireless Advanced Technology Lab of Lucent Technologies. His research interest includes digital transceiver design, adaptive modulation and channel coding, CDMA power control, soft handoff and CREST factor control algorithms, jointly adaptive multiple access protocols as well as short-range wireless adhoc networking. He is currently working on BLAST–MIMO systems, iterative decoding and UMTS call processing protocol stack design.
E-mail: knlau@lucent.com