EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Channel condition self-clocked packet scheduling scheme for wireless networks

Jyh-Ming Chen, Eric Hsiao-Kuang Wu, Hsiang-Wei Lu, Ching-Hsiang Chu and Meng-Feng Tsai*

Abstract

Since mobile hosts suffer from burst and location-dependent channel errors in wireless networks, packet fair queueing algorithm in wireline networks cannot be applied directly to wireless networks. Generally, a fair scheduler in a wireless network retains the virtual time of flow when the flow encounters channel errors. This results to the flow having higher priority when it exits from errors, and the system can compensate the lost service for the flow. This causes the lagging flows to capture the shared channel and affects the queueing delay of flows perceiving a clean channel. In this paper, we present a channel condition self-clocked packet scheduling scheme. This algorithm can address the problem and achieve the following goals: (1) steady delay and jitter for flows perceiving an error-free environment, (2) delay and throughput guarantees in an error-free environment, (3) short-term fairness among flows perceiving an error-free environment, and (4) long-term fairness for error system. Our algorithm is based on start-time fair queueing, and the virtual time of flows is reset to contend for forwarding its packet the next time when errors happen for the flow.

Keywords: Channel condition self-clocked packet scheduling, Wireless scheduler, Fair queue

1. Introduction

Multimedia streaming services are becoming increasingly popular in third-generation (3G) and fourth-generation (4G) mobile networks. One important characteristic of multimedia streaming services is the load asymmetry between downlink and uplink, and the services are mostly used in the downlink. In order to enhance downlink performance, some systems have been proposed. It includes the high data rate system, high-speed downlink packet access (HSDPA), and [3G partnership project technical report (3GPP TR) 25.848; 3GPP TR 25.308] system. These systems introduce many schemes such as adaptive modulation and coding technique (AMC), hybrid automatic repeat request (HARQ) scheme, the high-speed channel shared by multiple users, quality of service (QoS) control for packet services [1,2]. In recent years, worldwide interoperability for microwave access (WiMAX) had been proposed to be a candidate of the 4G network system. WiMAX aims at providing long distance and high data rate in various metropolitan areas of wireless network. Although WiMAX applies many new techniques such as use of orthogonal frequency division multiple access, time and frequency division duplexing (TDD and FDD), multiple QoS classes for a combination of data, and voice and video service, WiMAX leaves the packet scheduling algorithm for uplink and downlink as an open issue [3,4].

The packet scheduler plays a key role in the system especially for the downlink because it can provide the quality of service targets that should be met for applications [5-16]. Scheduler designers need to consider the allocations logically and physically. Logically, the scheduler should calculate the number of slots based on QoS service classes. Physically, the scheduler needs to select which subchannels and time intervals are suitable for each user. The goals of schedulers are basically to meet QoS guarantees for all service classes, to minimize bit error rate, to maximize the system throughput, to maintain the fairness, to have as less a complexity as possible, and finally, to ensure the system scalability.

Some types of QoS parameters are delay, jitter, loss rate, or throughput. For real-time applications (e.g., audio/video streams), in additional to delay and throughput, it requires the guarantee of the jitter.

* Correspondence: mftsai@csie.ncu.edu.tw
Department of Computer Science and Information Engineering, National Central University, No. 300 Jhongda Rd, Jhongli 32001, Taiwan

In many wireless fair queueing, the system gives higher priority to the flow that exits the error state for fairness among flows. This may cause the lagging flows to capture the channel, and the delay of flow in the clean channel becomes longer. Also, the jitter of flow in the clean channel is affected. For real-time application, the quality of service such as delay and jitter must be guaranteed [17,18]. In order to provide quality of service such as delay and jitter for the mobile host in the clean channel among flows' fairness, the channel condition aware scheme must be designed for real-time application.

1.1 Fairness

In wireless networks, the packet transmission may fail during a channel error interval due to fading, interference, and shadowing. In general, schedulers favor the users with better channel quality because the optimal resource allocation schedules the user with the best channel to exploit multiuser diversity and channel fading, or perhaps the scheduler does not allocate any resources for the mobile host with high error rate because the packets would be dropped anyway.

However, the schedulers also need to consider other users' QoS requirements such as the minimum reserved rate and may need to introduce some compensation mechanisms. The schedulers basically use the property of multiuser diversity in order to increase the system throughput.

A flow is said to perceive a clean channel if both the communication end points perceive clean channels and if the handshake can take place. A flow is said to perceive a dirty channel if either end point perceives a channel error [19]. We consider the following case as shown in Figure 1. There are two mobile hosts in the network. The first mobile host is in a clean channel (the status is good), and the second mobile host may suffer from errors during transmission (the status is bad). Each mobile host is assumed to assign 50% of the throughput.

The system avoids transmitting/receiving the packet during errors for performance. In Figure 1, the second mobile host suffers from errors during the second and the fourth slot. The system swaps the second mobile host's slots with the first mobile host's slot. In this case, the first mobile host receives 70% of the service and the second mobile host receives 30% of the service. To solve the problem, there are some approaches. In class base queueing channel-state-dependent packet scheduling (CBQ-CSDPS) [20] (an enhancement version of CBQ), the system calculates the instant throughput and reallocates the mobile host's throughput. In some wireless fair queue, when the flow exits the error state, the system compensates the lost service for the flow.

1.2 Network model

In this paper, we consider a shared-channel packet-cellular wireless network model with a high-speed wired backbone to represent a general third- or fourth-generation network system such as HSDPA or WiMAX. Each cell is served by a base station which performs the centralized scheduling of packet transmissions. All transmissions are assumed to be downlinked (for base station to mobile host). Each mobile host can communicate with the base station, but it may suffer from location-dependent channel errors. To the best of our knowledge, HSDPA designs an uplink high-speed dedicated physical control channel to carry the necessary control information in the uplink such as hybrid automatic repeat request (HARQ) and channel quality indicator channel (CQICH) feedback signaling related to the downlink transmission [2]. WiMAX also applied the feature of HARQ and enhanced CQICH feedback channels to carry ACK/NACK information corresponding to downlink transmissions periodically [4]. Therefore, we assume that the base station can detect the mobile hosts' next channel status perfectly through the mobile hosts' instantaneous acknowledgment of channel conditions and packet queue status of all mobile hosts. If the mobile host perceives a channel error, it cannot receive or transmit data during the interval. The error period is short and occasionally relative to the lifetimes of the flow.

The rest of this paper is organized as follows. In Section 2, we introduce the operation of fluid fair queueing (FFQ) and packet fair queueing (PFQ) in

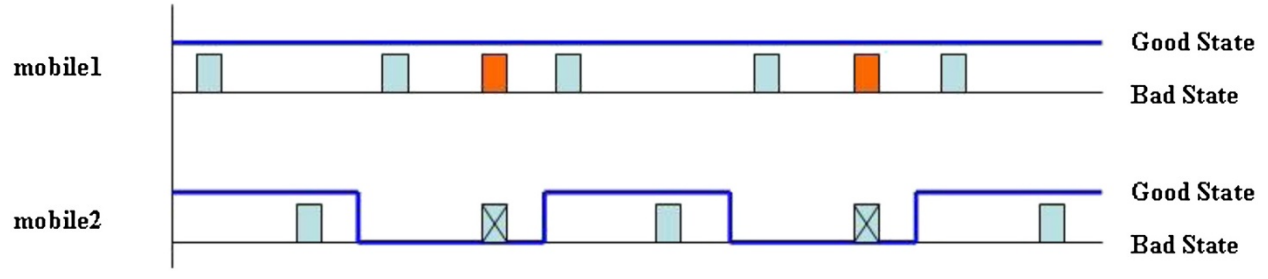

Figure 1 Fairness in the wireless network.

wireline networks and describe the problem that FFQ and PFQ have in wireless networks. Also, we describe some wireless fair queue as follows: idealized wireless fair queueing (IWFQ), wireless packet scheduling (WPS), channel condition-independent fair queueing (CIF-Q), server-based fairness approach (SBFA), wireless general processor sharing (WGPS), proportional fairness scheme (PFS), and modified largest weighted delay first (M-LWDF). In Section 3, we present our basic idea and channel condition self-clocked packet scheduling (CSCPS). In Section 4, the proposed scheme is compared with CIF-Q and SBFA through simulation to show the performance improvement. In Section 5, we conclude our contributions and point out future works.

2. Related work

In wireline networks, many packet fair queueing [21-23] have been proposed which can provide fairness and delay bound. Because there are some characteristics of the wireless network such as location condition-dependent error, channel contention, and burst errors, packet fair queueing in the wireline network cannot be applied directly in the wireless network. Several wireless fair queueing [3,20,24-30] have been developed, and they can provide short-term fairness for an error-free system, long-term fairness, graceful degradation in service for flows perceiving clean channel, and delay bound.

2.1 Fluid fair queueing and packet fair queueing

In wireline networks, the FFQ model is called general processor sharing (GPS) [22]. In the FFQ model, each packet flow is treated as a fluid flow and the flow i is allocated a weight r_i. During any time interval (t_1, t_2), there is a set of backlogged flows $B(t_1, t_2)$, which receives the service in proportion to its weight for each flow. It satisfies the following property:

$$\forall\ i,\ j \in B(t_1,t_2)\quad \left|\frac{W_i(t_1,t_2)}{r_i} - \frac{W_j(t_1,t_2)}{r_j}\right| = 0. \tag{1}$$

PFQ is based on the FFQ model. In PFQ, the system maintains the system virtual time v. Each packet is associated with a start tag and a finish tag, which correspond respectively to the virtual time at which the first bit of the packet and the last bit of the packet are served in fluid fair queuing. The scheduler then serves the packet with the minimum finish tag in the system. The kth packet of flow i that arrives at time a_i^k is allocated a start tag S_i^k and a finish tag F_i^k as follows:

$$\begin{aligned} S_i^k &= \max\left\{ S_i^{k-1} + \frac{L_i^{k-1}}{\phi_i}\ ,\ \ v\left(a_i^k\right) \right\}, \\ F_i^k &= S_i^k + \frac{L_i^k}{\phi_i}, \end{aligned} \tag{2}$$

where $v(t)$, the virtual time at time t, denotes the current round of service in the corresponding fluid fair queueing service. L_i^k is the length of the kth packet of the flow i, a_i^k is the arrive time of the kth packet, and ϕ_i is the weight of flow i.

2.2 Wireless fair queueing

In wireline networks, FFQ and PFQ provide delay bound, minimum throughput, and fairness. Packet losses in wireline networks are very rare. In contrast, there are location-dependent channel errors in wireless networks. This is due to the distance between the transmitter and receiver, and users' move causes fading, interference, and shadowing phenomenon. When the packet is transmitted in error status, the transmission fails and wastes the system's resource. The prediction mechanism can predict the next channel state. If the flow will be fallen into error state, the system swaps the flow with another error-free flow. It happens that the backlogged flow is unable to transmit due to channel error. This causes unfairness problem. Thus, the wireless fair queueing must compensate the lost service for the flow.

2.2.1 Idealized wireless fair queueing

IWFQ [24] algorithm uses WFQ for its error-free service. Each arriving packet is tagged as in weighted fair queueing, and the service tag for a flow is set to the finish tag of its head-of-line (HOL) packet. Among the flows that can be transmitted (i.e., channel is good and flow is backlogged), a HOL packet of the flow with least service tag will be picked and transmitted. For lagging flows with dirty channel, if the aggregated length of packets with finish tags less than the virtual time is greater than a constant bound (B_i) bits, then the first Ni packets in the queue and service tag are the only ones that are retained.

The compensation mechanism favors channel access for lagging flows. When a flow is denied, service due to channel error, its service tag retains. When its channel becomes error-free, a lagging flow has a low service tag and captures the channel. The compensation mechanism makes the lagging flow catch up its lag, but it may starve out, leading flows in the short term.

2.2.2 Idealized wireless packet scheduling

WPS [24] uses the weighted round robin as in WFQ as its error-free service. Consider three backlogged flows f_1, f_2, f_3 and have weights (in terms of packets) 2, 2, 3, respectively. If all flows are backlogged, WPS generates a basic slot allocation identical to WFQ using spreading as follows: $<f_3, f_1, f_2, f_3, f_1, f_2, f_3>$.

At any instant, the one-step prediction algorithm predicts the next channel state. If prediction algorithm predicts that the backlogged flow has channel error, WPS tries to swap the slot with the other slot of flow with clean channel within the same frame. If there is no backlogged flow with clean channel in the frame, the flow credits the slot to the flow with clean channel in the next frame. At the beginning of the frame, the effective weight for the flow with credit is the sum of its default weight and its credits and the effective weight for the flow with debit is given by its default weight minus its debits.

In WPS, intraframe swapping is first attempted to compensate the flow with dirty channel. If intraframe swapping fails, WPS tries to adjust the system of credit/debit. This presents the effective weight of flow at the beginning of the frame. WPS alleviates the problem that the large lagging flow captures the channel in IWFQ.

2.2.3 Channel condition-independent fair queueing

CIF-Q [29] uses start-time fair queueing (SFQ) [21] as the error-free system. SFQ tags the start-time of packet as service tag and makes the HOL packet with the smallest transmit first.

Besides virtual time v_i, each flow i is associated to an additional parameter lag_i that represents the difference between the service that flow i should receive in a reference error-free packet system and the service it has received in the real system. An active flow i is said to be 'lagging' if its lag_i is positive, leading if its lag_i is negative, and sync otherwise.

As in IWFQ, the packet is served in the increasing order of its virtual starting time in CIF-Q; otherwise, the packet retains its starting time. Whenever a flow is selected, the HOL packet is transmitted if the channel is clean or the flow is lagging. When a flow is selected, HOL cannot be transmitted if the channel is dirty or the flow is leading; thus, it has to give back its lead. In CIF-Q, a flow is 'active' if it is backlogged or leading.

The backlogged leading flow uses a parameter α to control the basic rate of the flow. In other words, the backlogged leading flow relinquishes the $(1 - \alpha)$ amount of its service to the lagging flow. If the flow is leading and non-backlogged, it relinquishes the slot to the lagging flows. When leading relinquished slots, lagging flows receive additional service in proportion to the lagging flow's 'rate' instead of giving all of it to the session with the largest normalized lag. When a lagging flow j wants to leave, its positive lag_j is proportionally distributed among all the remaining active flow i such that each lag_i is updated according to the following equation:

$$\text{lag}_i = \text{lag}_i + \text{lag}_j \times \frac{r_i}{\sum_{k \in A} r_k}, \quad (3)$$

where A represents the set of the remaining active flows and r_i is the weight of flow i.

2.2.4 Server-based fairness approach

SBFA [30] provides a framework which can be integrated with wireline network scheduling to the wireless domain. SBFA reserves a portion of the bandwidth for compensating the lost service of flow due to channel error. If the packet is transmitted in the dirty channel, the transmission may fail and the packet is unable to recover at the receiver. If this happens, the packet must be retransmitted for fairness. SBFA provides a special flow called long-term fairness servers (LTFS), which is used to compensate the lost service for fairness.

If the packet of flow cannot transmit due to the dirty channel, it creates a virtual slot and en-queues the slot into the LTFS. The LTFS provides compensation in a first-input first-output (FIFO) manner. Thus, a lagging flow may capture compensation slots till it becomes in sync flow.

Since the LTFS functions to compensate the service for the flow with dirty channel, a leading flow does not relinquish the slot to the lagging flow. The lagging flow cannot receive additional service.

2.2.5 Wireless general processor sharing

The WGPS [26] that deals with burst and location-dependent channel error is the extension of general processor sharing. When the errors happen, WGPS does not provide the service for the flow in an error state. Afterwards the flow's channel becomes good; the flow will receive the lost service that flows in the dirty channel.

The difference of the WGPS between the IWFQ and CIF-Q is that the flow is associated with a time-varying service share in WGPS. In IWFQ and CIF-Q, when the flow suffers from channel error, the flow retains its service tag. When the channel becomes good, the flow has higher priority. In WGPS, when the flow suffers from channel error, it calculates the amount of service required to compensate. When the channel exits the error state, the flow's service share is the sum of the original

service share and compensated service share Δ. At time t, the flow's service share is defined as follows:

$$\begin{array}{ll} \phi_i(t) = \phi_i \times (1+\Delta); & \text{if } i \in S(t) \text{ and } i \in C(t) \\ \phi_i(t) = \phi_i; & \text{if } i \in S(t) \text{ and } i \notin C(t) \\ \phi_i(t) = 0; & \text{otherwise} \end{array} \quad (4)$$

where $S(t)$ is a set of backlogged flows and $C(t)$ is a set of the flows that require compensation.

Figure 2 shows that the flow suffers from error in the interval (t_b, t_r); the flow is in a bad channel state, and its service share is zero. During the compensation period, the flow's service share equal to $\phi_i \times (1 + \Delta)$. At time t_c, the flow receives the lost service completely, and then its service share becomes ϕ_i.

2.2.6 Proportional fairness scheme

The objective of PFS [31,32] is to maximize long-term fairness. PFS uses the ratio of channel capacity (denoted as $W_i(t)$) to the long-term throughput (denoted as $R_i(t)$) in a given time window T_i of queue i as the preference metric instead of the current achievable data rate. $R_i(t)$ can be calculated by exponentially averaging the ith queue throughput in terms of T_i, and then the user with the highest ratio of $W_i(t)/R_i(t)$ receives the transmission from the base station (BS).

PFS tries to balance the capacity-fairness trade-off by serving the users with the best relative channel quality, where the relative channel quality is the user's channel quality condition divided by the user's average throughput. Therefore, the PFS scheme gives more priority to users as their average throughputs decrease in order to prevent users with good channel quality conditions from monopolizing the wireless resources. Generally, the PFS encounters lower system performance and resource utilization [33].

2.2.7 Modified largest weighted delay first

M-LWDF [34] can provide a QoS guarantee by ensuring a minimum throughput guarantee and also maintains delays smaller than a predefined threshold value with a given probability for each user. Also, it is provable that the throughput is optimal for LWDF [35]. The algorithm can achieve the optimal whenever there is a feasible set of minimal rates area. The algorithm explicitly uses both the current channel condition and the state of the queue into account. The scheme serves queue j for which $\rho_j W_j(t) r_j(t)$ is maximal, where $W_j(t)$ is the HOL packet delay for queue j, $r_j(t)$ is the channel capacity with respect to flow, and ρ_j is a constant which could be different for different service flows (the difficulty is how to find the optimal value of ρ_j).

In [34], M-LWDF is proposed to accommodate real-time traffic. M-LWDF uses the relative channel quality condition to compute the user's priority similar to that in PFS. However, to accommodate real-time traffic with delay requirements, M-LWDF multiplies the user's relative channel quality condition by a term representing the user's packet delay. This term ranges from 0 to 1, where it approaches 1 as the user's head-of-queue packet delay approaches its threshold. It has been shown in [36] that M-LWDF may result in an unfair distribution of wireless resources since if two users have the same head-of-queue packet delay, they will be assigned different priorities if their supportable data rates are different. A comparison of the scheduling disciplines [3,37] is presented in Table 1.

3. Proposed scheme: CSCPS

Many well-known wireless fairness queueing function such that the bandwidth resource is fairly shared by mobile hosts and delay bound for real-time application. The wireless fairness queueing is the extension of fairness queueing in the wireline network, and it suffers from problems such as when lagging flows capture the share channel or unsteady jitter. In this section, we address two problems and design the efficient algorithm in wireless network for real-time application.

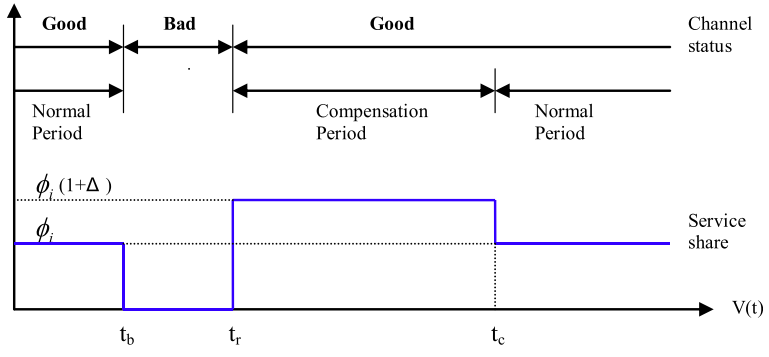

Figure 2 Compensation of error channel in WGPS.

Table 1 Comparison of scheduling mechanisms

Scheduling algorithms	Pros	Cons
IWFQ	With proper and dynamic weight, guarantees throughput and delay, fairness	In IWFQ, the main difficulty is in adjusting the values of the bounds since it exhibits a trade-off between fairness and delay/throughput guarantees
CIF-Q	Guarantees throughput and delay, fairness	CIF-Q is a high algorithmic complexity since it needs to keep a record of leading and lagging counters
SBFA	Achieve long-term fairness	A flow with a good channel state may receive much more service than its promised share
PFS	Achieve long-term fairness	Cannot guarantee the delay constraint, lower system performance, and resource utilization
M-LWDF	Meets throughput and delay guarantee with threshold probability	The difficulty is how to find the optimal value of ρ_j
CSCPS	Steady jitter, guarantees throughput and delay, fairness	CSCPS is a high algorithmic complexity since it needs to keep a record of leading and lagging counters

3.1 Problem description

3.1.1 Capture channel problem

In a work-conserving system such as IWFQ [24] and CIF-Q [29], when the mobile host receives the service, the mobile host's virtual time increases. Because there are location-dependent channel errors in the wireless environment, this might result in the difference of the virtual time between error-free and error systems. We introduce a circumstance as follows; if the mobile host suffers from channel errors, the mobile host is allowed to retain its virtual time. In the meantime, the mobile hosts in the clean channel receive the additional service to transmit data. Afterward, the mobile host exits from the error state and will have the highest priority to transmit data due to the fact that the mobile host's virtual time is the smallest among mobile hosts. Other mobile hosts in the clean channel may wait for the mobile host that exited from the error state to catch up on the amount of lost service. If many mobile hosts suffer from burst errors, mobile hosts perceiving a clean channel may not be able to transmit data for a long time.

3.1.2 Unsteady jitter problem

We consider the unsteady jitter case. There are two mobile hosts in the system. One mobile host is always in the clean channel, and the other mobile host may suffer from channel errors.

Figure 3 shows a result of the mobile host transmitting data in the clean channel. Figure 3a represents the mobile host suffering from an error in the interval (t_b, t_r) and receives no service. When the channel becomes good at t_r, the mobile host is compensated with the amount of lost service. During the interval (t_r, t_c), the mobile host catches up on the amount of lost service. After t_c, the mobile host shares bandwidth by its pre-allocated weight.

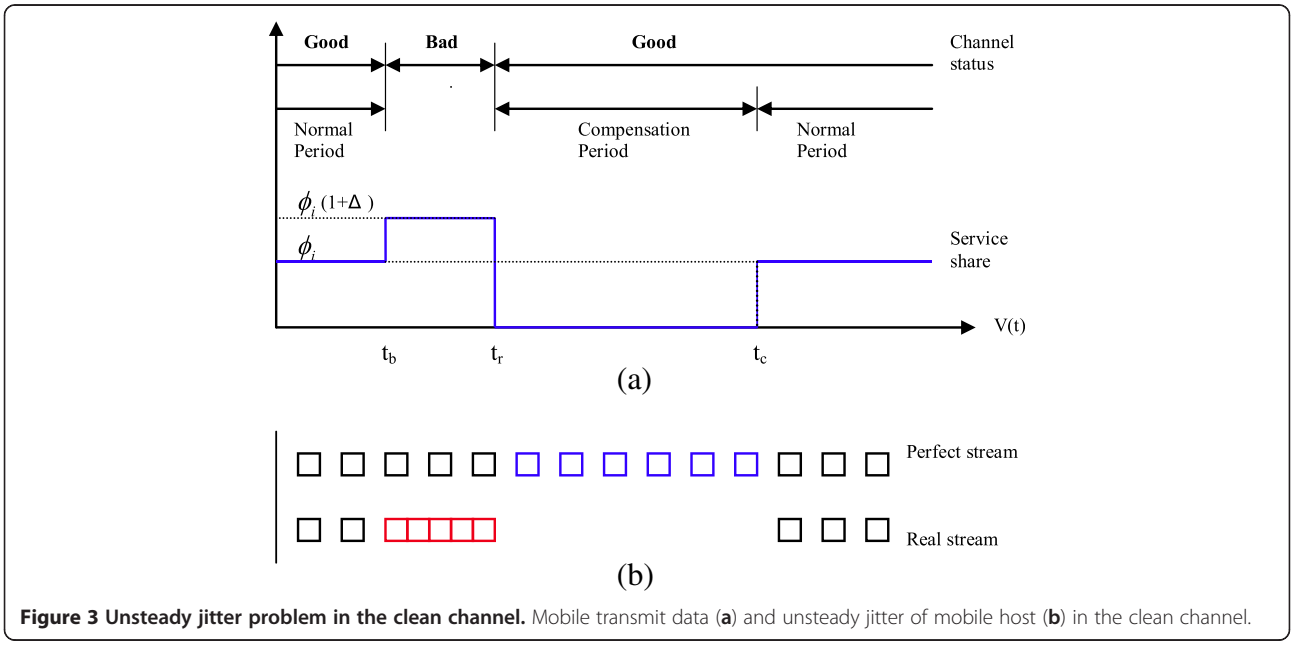

Figure 3 Unsteady jitter problem in the clean channel. Mobile transmit data (**a**) and unsteady jitter of mobile host (**b**) in the clean channel.

Figure 3b represents the unsteady jitter of mobile host in the clean channel. The mobile host with clean channel receives the sum of its service and the service of the other mobile host that suffers from errors during the interval (t_b, t_r), and its queue delay becomes shorter. When the mobile host exits the error state at t_r, its virtual time retains at t_b, and the mobile host has the highest priority. In the meantime, the mobile host in the clean channel may receive no service, and its queue delay becomes longer. This causes the unsteady jitter of mobile host in the clean channel.

3.2 Channel condition self-clocked packet scheduling

3.2.1 Basic idea and start-time fair queueing

In SFQ, the packet has a start tag and a finish tag. The packet is transmitted in the increasing order of the start tag. The algorithm is defined as follows:

$$\begin{aligned} S_i^k &= \max\left\{F_i^{k-1}\ ,\quad v\left(a_i^k\right)\right\}, \\ F_i^k &= \ S_i^k + \frac{L_i^k}{\phi_i}, \end{aligned} \tag{5}$$

where S_i^k is the start time of the kth packet of the flow i, F_i^k is the finish time of the kth packet of the flow i, $v(t)$ is the virtual time at time t, a_i^k is the arrive time of the packet, L_i^k is the length of the kth packet of the flow i, and ϕ_i is the weight of flow i.

In this paper, we modify the start-time fair queueing algorithm in a system with location-dependent channel errors. There is the reason for this choice. Start-time fair queueing has the property of having a low average as well as maximum delay, which means that the jitter of flow is steady in the error-free system.

Problems such as capture channel and unsteady jitter are caused by the backlogged flow retaining its virtual time in channel errors. In order to address these problems, we propose CSCPS which decides the next transmission time of packet for the flow suffering from errors. First of all, we assume that the real-time traffic is constant bitrate traffic. By the property of SFQ, in steady jitter, we can make good use of the period between packets to decide the start time of the next packet. If errors happen, the start time and finish time of packet in the flow will be calculated as follows:

$$\begin{aligned} S_i^k &= \max\left\{F_i^{k-1} + \text{during\_error},\quad v\left(a_i^k\right)\right\}, \\ F_i^k &= \ S_i^k \ + \frac{L_i^k}{\phi_i}, \end{aligned} \tag{6}$$

where during_error is the period when flow i is in channel error. The scheduler chooses the smallest start time of packet and forwards the packet for flow.

3.2.2 Detailed algorithm

We have described the basic idea of algorithm earlier. Besides the start tag and finish tag, each flow i is associated to an additional parameter lag_i which represents the difference between the services that flow i should receive in a reference error-free packet system and the service it has received in the real system. The start tag and finish tag of the packet are updated as the following:

$$\begin{aligned} S_i^k &= \max\left\{F_i^{k-1} + \alpha I_i^{k-1},\quad v\left(a_i^k\right)\right\}, \\ F_i^k &= \ S_i^k + \frac{L_i^k}{\phi_i}, \\ I_i^k &= \ S_i^k - S_i^{k-1}, \end{aligned} \tag{7}$$

where S_i^k is the start time of the kth packet of the flow i, F_i^k is the finish time of the kth packet of the flow i, I_i^k is the interval between packets, and α is either 1 in error or 0 in error-free.

The interval of flow represents that the next packet must wait for the period when the packet departs from the system. Since the system reserves the bandwidth for the flow, this means that the interval of flow is decided by its weight. In other words, the larger the weight of flow, the shorter is the flow interval. If errors happen for the backlogged flow, it must relinquish the slot to another flow with clean channel and updates its start tag and finish tag.

A flow is classified as leading flow, lagging flow, and sync flow. A flow is leading if and only if the flow receives additional service in the interval. When a backlogged flow suffers from channel error, its packet cannot be forwarded, that is, the received services are less than expected. A flow is leading, lagging, or sync as shown in Figure 4.

A flow is lagging if its parameter lag is positive and is leading if its parameter lag is negative; otherwise, it is a sync flow. For work-conserving system, the algorithm maintains the following equation:

$$\sum_{i\in B} \text{lag}_i = 0, \tag{8}$$

where B is the set of backlogged flow.

In CIF-Q and IWFQ, the flow retains its virtual time when it suffers from errors. Consequently, when flow exits from errors, it has the highest priority and the system compensates the lost service for the flow. We consider the worst case. In the worst case, there is only one leading flow and the others are lagging flows. When the leading flow forwards all of its packets, its new packet arrives at an empty queue. In the meantime, the lagging flows exit from the channel errors; delay of the leading flow will be longer due to the system compensating the service for lagging flows.

In order to solve the condition, we propose the different ideas of compensation. Figure 5 shows a compensation

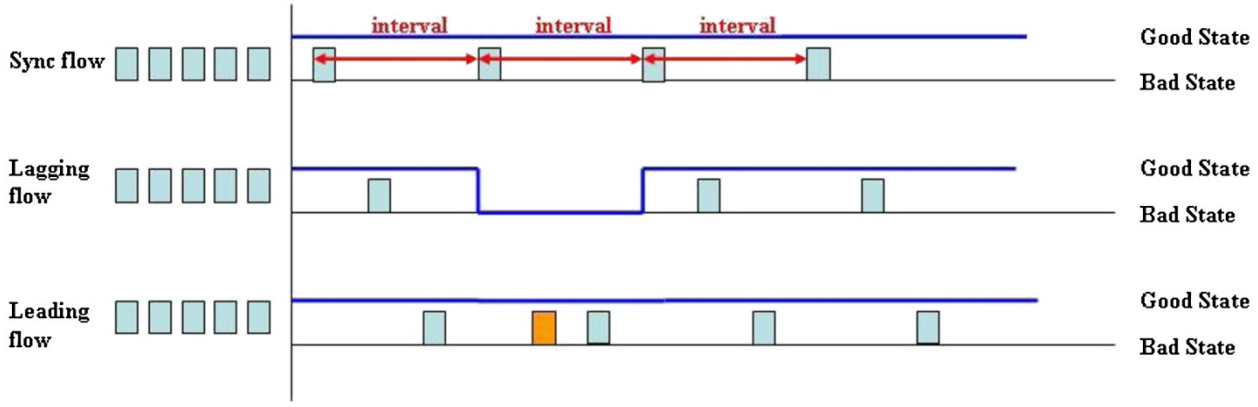

Figure 4 Lagging, leading, and sync flow model.

approach. When the second mobile host suffers from errors, the third mobile host can receive additional service. When the second mobile host exits the error state, the system does not compensate the lost service for the mobile host right away. In Figure 5, the error happens for the third mobile host in the third slot, and the lagging flow (the second mobile host in the figure) is compensated the lost service (red tag in figure), that is, the lagging flow is compensated by the system when another active flow suffers from errors. However, in special cases such as when some mobile hosts are always in the clean channel, they do not release the resource from the mobile hosts perceiving channel error. Therefore, the flow with clean channel is selected, and the system can compensate the lagging flows. It should be noted that the virtual time of flow be not changed when it is compensated or when it gets more service than other flows that suffer from error to relinquish. To avoid starvation of the leading flow for compensating the lagging flow, the leading flow is associated with a parameter-compensated window. The compensated window denotes that leading flows can spend the period compensating lagging flows. This period is the maximum time of starvation for leading flows. When the period is over, leading flows can contend to forward their packets regardless of the existence of lagging flows. Thus, once the leading flows are served by system, its compensated window can start to compensate the lagging flows.

In order to maintain long-term fairness, we must consider the following case: what happens if a flow wants to leave the system? We adopt the approach of CIF-Q. In CIF-Q, there are two situations. One is that the leading flow wants to leave the system, and the other is that the lagging flow wants to leave the system. In order to maintain Equation 8, we must modify the lag of other flows. When a lagging flow j wants to leave, its positive lag_j is proportionally distributed among all the remaining active flow i such that each lag_i is updated according to the following equation:

$$\mathrm{lag}_i = \mathrm{lag}_i + \mathrm{lag}_j \times \frac{r_i}{\sum_{k \in A} r_k}, \tag{9}$$

where flow i is the remaining flow, flow j wants to leave, r_i is the weight of flow i.

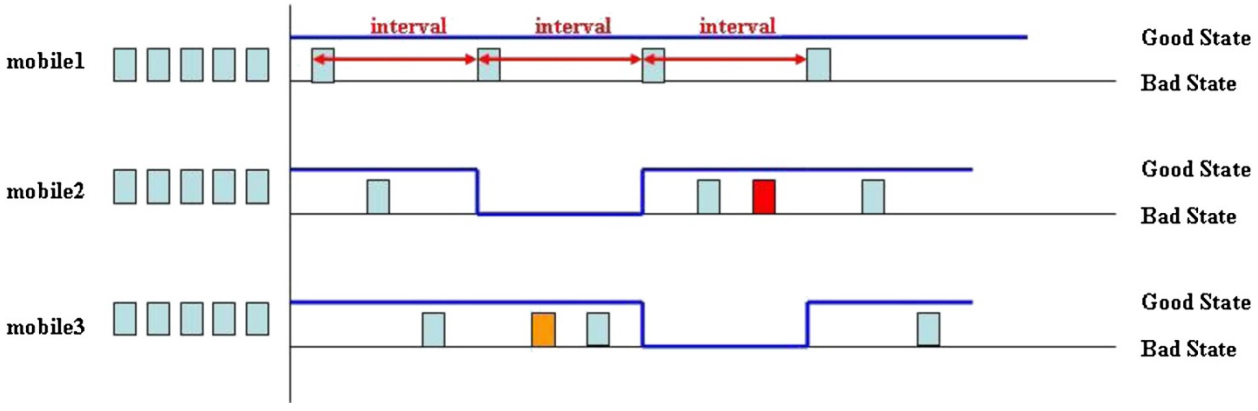

Figure 5 Compensation mechanism of CSCPS.

In contrast, the system does not allow the leading flow to leave until the leading flow relinquishes its lead.

3.2.3 Delay guarantees

In this section, we provide analytical results for delay guarantees. We assume that flows suffer from short error bursts.

Theorem 1 *Given the kth packet of flow i in the error system, its queue delay is bounded as follows:*

$$d_i^k \le e_i^k + (n-1)\frac{L_{\max}}{R} + \frac{l_i^k}{R} + \frac{l_i^k}{r_i}, \tag{10}$$

where R is the channel capacity, r_i is the weighted rate of the flow, n represents the total number of active flows, d_i^k is the kth packet of the departure time of flow i, e_i^k is the expected arrival time of the kth packet of session i, l_i^k is the kth packet of the length of flow i, and L_{max} represents the maximum size of a packet.

Proof The packet arrives at the HOL, and its queue delay is due to its weight and possible errors in the channel. If an error happens, the flow's queue delay increases one interval. In the reference error-free system, we have the following:

$$d_i^r = d_i + I_i^{\text{error}}, \tag{11}$$

where d_i^r represents the queue delay in the error system, d_i represents the queue delay in the error-free system, and I_i^{error} represents the period when the flow is in the error channel.

We consider the problem of a two-state channel model. Figure 6 shows the state diagram of the two-state channel model. In Figure 6, state G is the good state channel and state B is the bad state channel. P_{GG} is the probability that the good state maintains to be good in the next state, P_{BB} is the probability that the bad state maintains to be bad in the next state, P_{GB} is the probability that the good state transforms into a bad state in the next state, and P_{BG} is the probability that the bad state transforms into a good state in the next state.

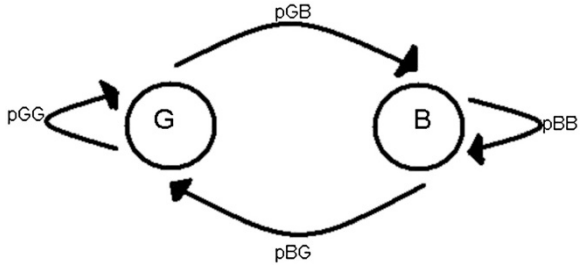

Figure 6 Two-state channel model.

The average probability of being in state G and B can be calculated as $\pi_G = \frac{P_{BG}}{P_{BG}+P_{BG}}$ and $\pi_B = \frac{P_{GB}}{P_{BG}+P_{GB}}$, respectively, and the average packet loss rate can be calculated as $p = p_G\pi_G + p_B\pi_B$. Thus, we have the following:

$$\begin{aligned} I_i^{\text{error}} &= P^1 \times 1 \times \frac{l_i^k}{r_i} + P^2 \times 2 \times \frac{l_i^k}{r_i} + P^3 \times 3 \times \frac{l_i^k}{r_i} + \ldots. \\ &= \sum_{n=1}^{\infty} P^n \times n \times \frac{l_i^k}{r_i} \\ &= \frac{p}{(1-p)^2} \times \frac{l_i^k}{r_i}, \end{aligned} \tag{12}$$

where p represents the error rate of the channel.

We assume that the error rate of channel is acceptable. Thus, we have the following:

$$I_i^{\text{error}} \le \frac{l_i^k}{r_i}. \tag{13}$$

Consequently, according to Equation 11, the departure time of the packet of flow i in the system is bounded by the following:

$$d_i^r \le d_i + \frac{l_i^k}{r_i}. \tag{14}$$

The departure time of the kth packet of flow i in the error-free system with SFQ is denoted as follows:

$$d_i^k = e_i^k + (n-1)\frac{L_{\max}}{R} + \frac{l_i^k}{R}. \tag{15}$$

Finally, we can gain the result of delay bound for the error system:

$$d_i^k \le e_i^k + (n-1)\frac{L_{\max}}{R} + \frac{l_i^k}{R} + \frac{l_i^k}{r_i}.$$

4. Simulation results

In order to evaluate the properties of fairness, delay bound guarantees and performance improvement of steady queue delay and jitter for the proposed packet scheduling scheme (CSCPS) are presented. In this section, several experimental simulations are presented by using the network simulator tool (NS 2.35) [38].

Figure 7 shows the basic topology used for following all simulations. Si denotes the transmitter, and the Ui denotes the receiver. There is a router between nodes Si and BS with wired links. BS denotes the base station which connects to the router with a wired link and to receivers with a wireless link. The wireless channel is applied either as an error-free or a two-state channel error model which is depicted in Figure 6. For the two-state channel error model, the value of the following are

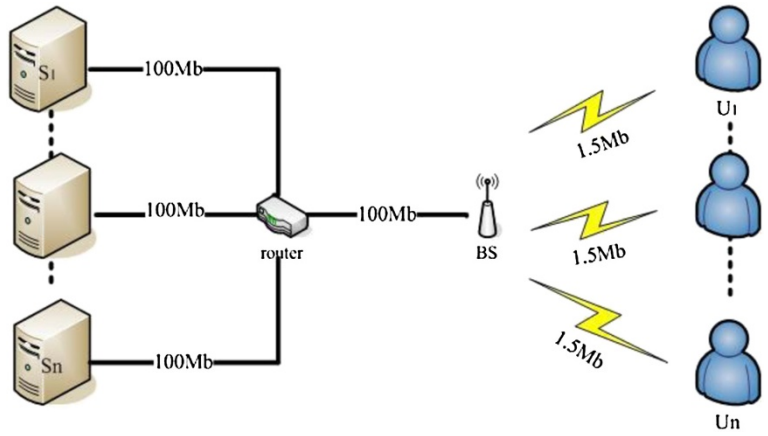

Figure 7 Basic simulation topology where location-dependent channel errors in a wireless environment are present.

presented: $P_{GG} = 0.94$, $P_{BB} = 0.96$, $P_{GB} = 0.4$, and $P_{BG} = 0.01$. Under this error model, channels tend to stay at the same state and the flow easily experiences burst error while transmission behaves to more realistic situations. Bandwidth of the link between Si, router, and BS is 100 Mbps, and the propagation delay is 10 ms. The wireless channel value between the base station and the receiver is 1.5 Mbps, and the propagation is 10 ms. User datagram protocol/constant bitrate (UDP/CBR) traffic is applied to generate constant bitrate real-time traffic flows from Si to Ui through router and BS. The packet size is set to 1,000 bytes to simplify the observation and analysis.

Obviously, the bottleneck for this topology is located at wireless channels. Therefore, the packet scheduling algorithm on BS plays a key role for the network's performance. The measurement metrics include instantaneous and average queue delay, throughput, and jitter on BS since it directly reflects the properties and performance of fairness, delay bound, throughput guarantees, and steady queue delay and jitter.

4.1 Delay bound and throughput guarantees for flows in error-free channel

Delay bound and throughput guarantees in the error-free channel are the common objectives for packet fair scheduling algorithms. We first present the queue delay of CSCPS and compare the simulation result to the analysis. Based on Equation 10, the delay of a packet in the flow is affected by the expected arrival time (i.e., time difference between the start and finish tags), the number of flows in the system, bandwidth of bottleneck link, packet size, and the weighted rate of the flow since the packet size and bandwidth of links are fixed in the simulation for ease analysis. Therefore, only the expected arrival time, number of flows and weighted rate of the flow will affect the analytic delay bound. According to our definition and algorithm, the expected arrival time is the most significant variable among the three variables since it takes the channel condition into consideration.

In this section, we introduce one UDP/CBR flow in the error-free channel and two flows in the two-state error channel as cross traffic. The weights of flows are identical as 2 at the beginning. Only the weight of the flow in the error-free channel can be changed every 10 s to observe the effect on analysis and simulation result. The weight of the flow is changed to 5 at 10 s, to 1 at 20 s, and to 3 at 30 s. Figure 8 shows the queue delay of the flow in the error-free channel. It reveals that the simulation delay time is bounded by the analytic delay time, and the analytic result is dominated by the expected arrival time rather than the weighted rate of the flow.

For example, the compensation action most frequently takes place during (0 and 10 s) and (20 and 30 s) due to the weight of flow in the error-free channel which becomes smaller than other flows. This leads to a bigger range of queue delay in Figure 8. Although the weight of the flow is 5 at 10 and 20 s, which is the largest in the simulation, the expected arrival time is still kept lower than other time intervals since CSCPS is aware of the channel conditions and dynamically increases or decreases the expected arrival time. Hence, CSCPS provides the delay bound guarantee for the flow in the error-free channel. On the other side, queue delay is the inverse of throughput. Lower queue delay leads to higher throughput. Therefore, Figure 8 also implies that CSCPS guarantees the throughput for the flow in the error-free channel.

4.2 Short-term fairness of flows in error-free channel

In this section, the experiment is built to illustrate the short-term fairness of CSCPS for the flows in the error-free channel. There are three 1.5 Mbps UDP/CBR traffic flows with different weights that last for 35 s. The weights

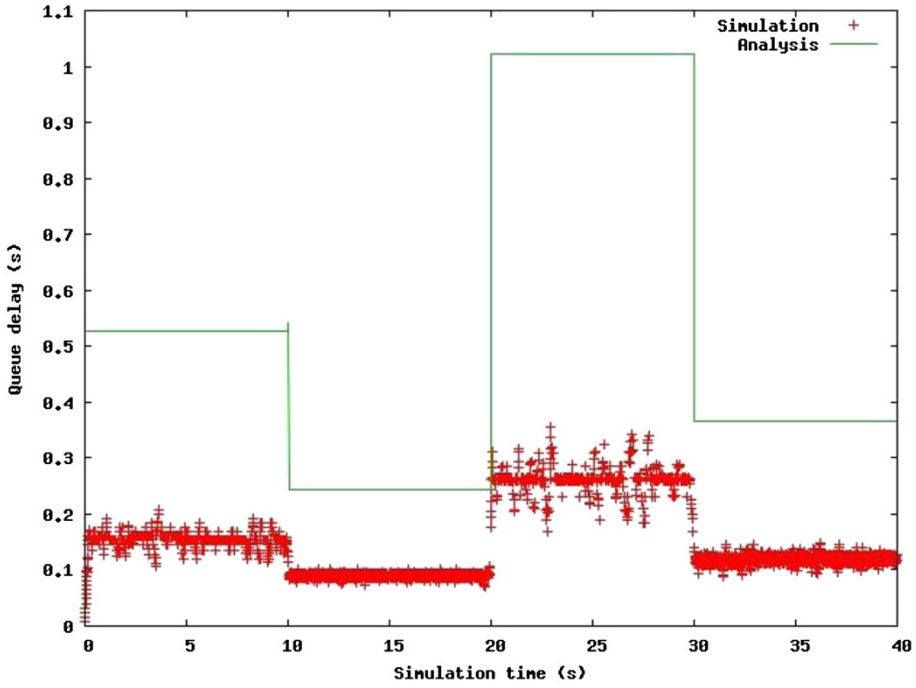

Figure 8 Comparison between queue delay and analysis with simulation.

of flows are reassigned every 5 s to observe its effect on queue delay and fairness. Table 2 shows the weights of flows used in this simulation.

Theoretically, the queue delay of flow should be inversely proportional to its weight, that is, when the weight of flows becomes larger, its queue delay becomes shorter. In the simulation result, the weights of flows are 4, 3, and 3 respectively from 5 to 10 s. The average queue delays of flows are 0.13, 0.17, and 0.17 s, respectively, and their ratio is about 3:4:4. Similar results can be observed in other intervals. Notice that in Figure 9, the distributions of queue delay for flows virtually overlap each other when they have identical weights. This observation reveals that the average queue delay of a flow would not be affected by other traffic flows but only by its weight. Therefore, this implies that our algorithm achieved the short-term fairness of flows for the error-free channel.

Table 2 Weights of flows that are reassigned every 5 s

	CBR1	CBR2	CBR3
5 ~ 10 s	4	3	3
10 ~ 15 s	2	4	4
15 ~ 20 s	4	3	3
20 ~ 25 s	6	2	2
25 ~ 30 s	5	3	2
30 ~ 35 s	4	3	3

4.3 Long-term fairness of flows in two-state error channel

In this simulation, we present the long-term fairness of CSCPS algorithm by showing the average queue delay of flows which is only affected by its weight even in the error channel. Three 1.5 Mbps UDP/CBR traffic flows are generated, and they last for 100 s. The weight of CBR1, CBR2, and CBR3 are 5, 3, and 2, respectively. A two-state channel model is applied. If a flow predicts that the slot suffers from errors, it will relinquish its slot to another flow with clean channel to transmit packets. Under this circumstance, the packet in queue of the flow with dirty channel will receive a longer queue delay. Therefore, the flow with clean channel receives additional service and its queue delay becomes shorter. Consequently, errors may cause unsteady queue delay for flows, but they still satisfy the delay guarantees for the flows in the error system.

The queue delay statistics for the flows is shown at Table 3. The average queue delays of CBR1, CBR2, and CBR3 are 0.1, 0.17, and 0.26 s, respectively. The ratio of average queue delay of flows is almost inversely proportional to their weight. In order to address the fairness issue, the received service between flows should be proportional to their weight. In Figure 10, clearly, the mean values of instantaneous throughputs of CBR1, CBR2, and CBR3 are about 750, 450, and 300, respectively. According to the results of delay and throughput of flows, it reveals that CSCPS algorithm achieves long-term fairness for flows in the two-state error channel model.

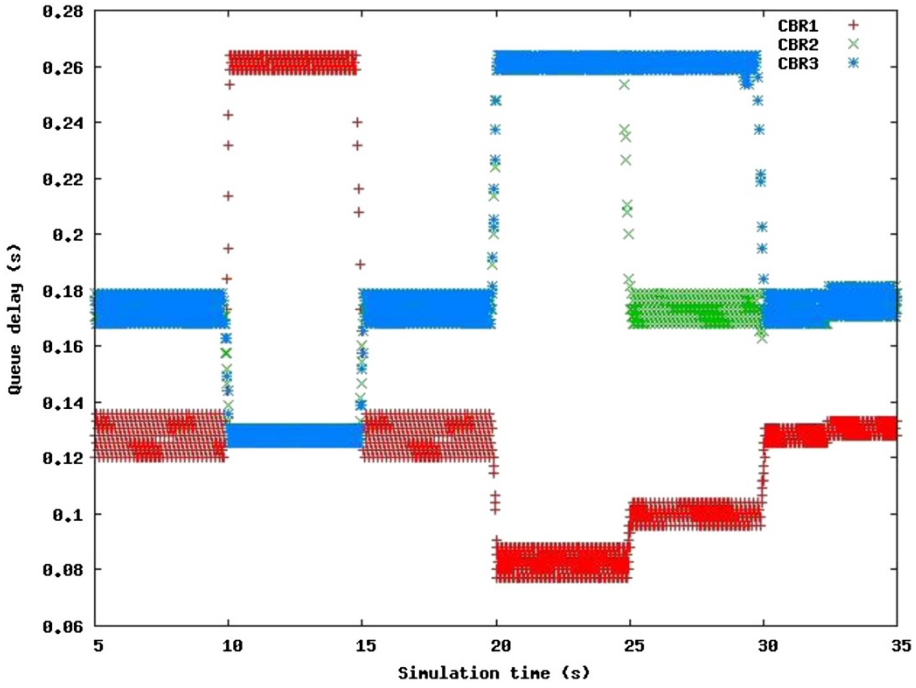

Figure 9 Queue delay of flows with different weights of CSCPS algorithm for error-free channel. The weight of the flow dominates the queue delay.

Through these three simulations, we can obtain the following contributions of CSCPS. CSCPS can achieve (1) delay and throughput guarantees in the error-free channel, (2) short-term fairness for the error-free channel, and (3) long-term fairness in the two-state channel error model.

4.4 Comparison of queue delay of flow in error channel using two-state channel model

In this simulation, we compare the queue delay performance of the proposed algorithm with other two packet scheduling algorithms, namely CIFQ and SBFA, in the wireless environment. There are five 1.5 Mbps UDP/CBR traffic flows in the system with a two-state channel model. The weights of flows are identical as 2.

Figure 11 shows the result of simulation. Since the SBFA algorithm reserves the portion of bandwidth in order to compensate for lagging flows if need be, obviously, some flows will share a less reserved bandwidth. Therefore, SBFA makes the flows to experience a longer queue delay than in other algorithms. On the other hand, if the channel error happens, the CIFQ algorithm makes the flow to retain its virtual time. This action will let the queue delay of flow become longer due to it suffering from errors. When flow exits from errors, the system compensates the service for the flow and its delay becomes shorter. There are two reasons that the delay of flow becomes longer: one is that the flow suffers from errors, and the other is that the flow is the leading type and the system compensates the service for lagging flows. In other words, the queue delay of flow may be affected by other flows. For our proposed CSCPS algorithm, the queue delay of flow is decided by its errors. For lagging flows, it can receive the compensated service from two approaches: one is that leading flows release their resource, and the other that is some flows suffer from channel errors. If the flow suffers from errors, its virtual time is updated to contend for the next forwarded chance. For the flow perceiving a clean channel, it is unaffected by other flows and can forward its packets. In this simulation, we get the result that CSCPS has a better performance than CIF-Q and SBFA algorithms for queue delay. The reason is that the queue delay of flow is unaffected by the lagging flow or its weight is stricken to become smaller.

Table 3 Queue delay statistics for the flows

	Maximum	Minimum	Mean
CBR1	0.147	0.008	0.1
CBR2	0.232	0.005	0.17
CBR3	0.394	0.01	0.26

4.5 Comparison of queue delay of flow in error channel using AMC technique

In this simulation, we compare the queue delay performance of the proposed algorithm with other two packet

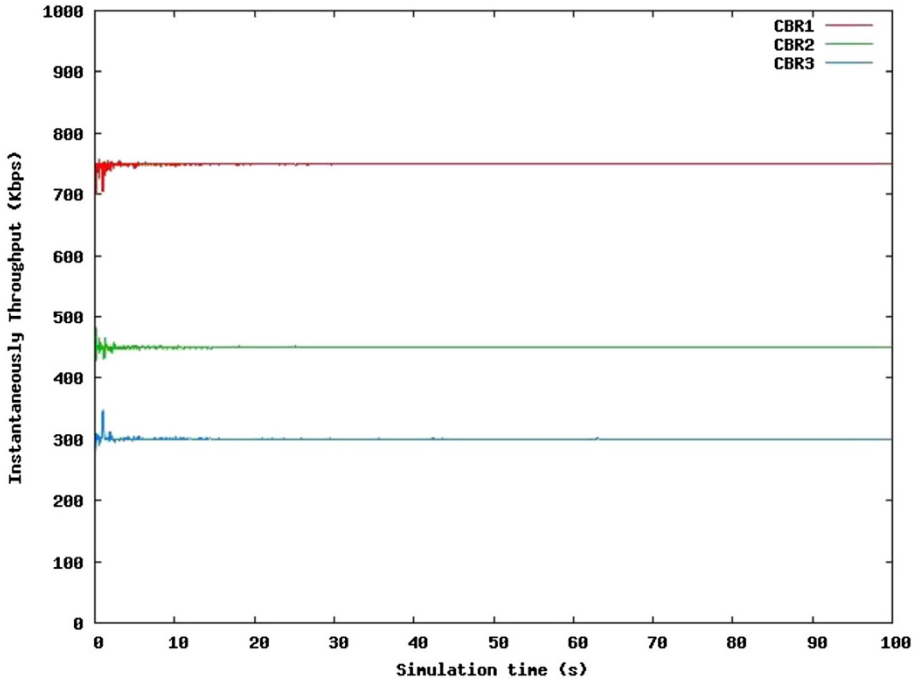

Figure 10 Instantaneous throughputs of three flows in two-state channel model. A steady throughput implies the long-term fairness property of CSCPS.

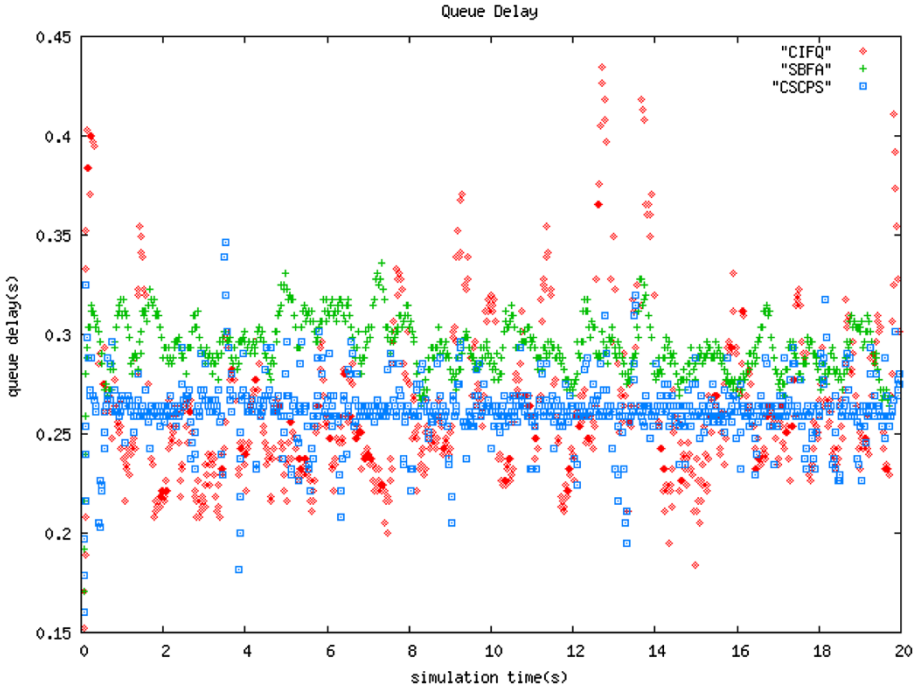

Figure 11 Comparison of queue delay in error channel using two-state channel model.

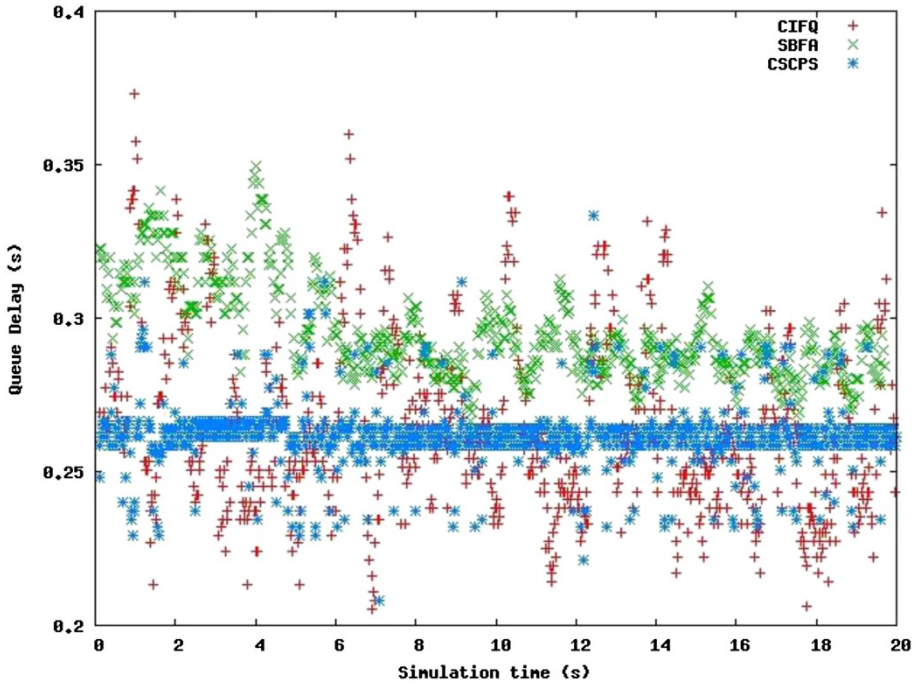

Figure 12 Comparison of queue delay among three algorithms for the flow in error channel using AMC. CSCPS provides a steadier queue delay.

scheduling algorithms, namely CIFQ and SBFA, using AMC. AMC has offered an alternative link adaptation method that promises to raise the overall system capacity by matching the modulation-coding scheme (MCS) to be flexible with the current received signal quality or average channel conditions. There are five 1.5 Mbps UDP/CBR traffic flows in the system with an AMC scheme using four MCS levels. The error model is the uniform distribution.

Figure 12 shows the result of simulation. The experimental result is kept the same as that of Subsection 4.4. To show the queue delay property precisely, Table 4 presents the queue delay statistics in terms of maximum, minimum, mean, variance, and standard deviation of the flow among three algorithms. The mean of queue delay of CSCPS is only 0.262 s, which is lower than other algorithms. Furthermore, the variance of queue delay of CSCPS is 0.163 ms, which is the lowest among the three algorithms. Lower mean, variance, and standard deviation of queue delay represent a steadier queue delay for the algorithm. In this simulation, we get the result that CSCPS has a better performance than CIF-Q and SBFA algorithms for queue delay.

4.6 Steady queue delay and jitter for the flow in error-free channel using two-state channel model

For multimedia streaming applications, steady queue delay and jitter are as important as short queue delay. Steady queue delay and jitter let real-time data to be transmitted smoothly [17,18]. Since the flow in the error-free channel have to compensate other flows if need be, this increases the queue delay. When there is no need of compensation, the queue delay of the flow will decrease immediately. Hence, whenever the state of flows switch between compensation and normal state, the queue delay varies dramatically in this period of time. This phenomena result in unsteady queue delay and jitter for both flows in the error-free and error channels. It is unavoidable for the flows in the error channel since those flows have to get additional slots to transmit

Table 4 Comparison of queue delay statistics for the flow in error channel using AMC

	Maximum (s)	Minimum (s)	Mean (s)	Variance	Standard deviation (ms)
CIFQ	0.373	0.179	0.263	7.11×10^{-4}	26.5
SBFA	0.349	0.264	0.289	2.36×10^{-4}	15.3
CSCPS	0.333	0.208	0.262	1.63×10^{-4}	12.7

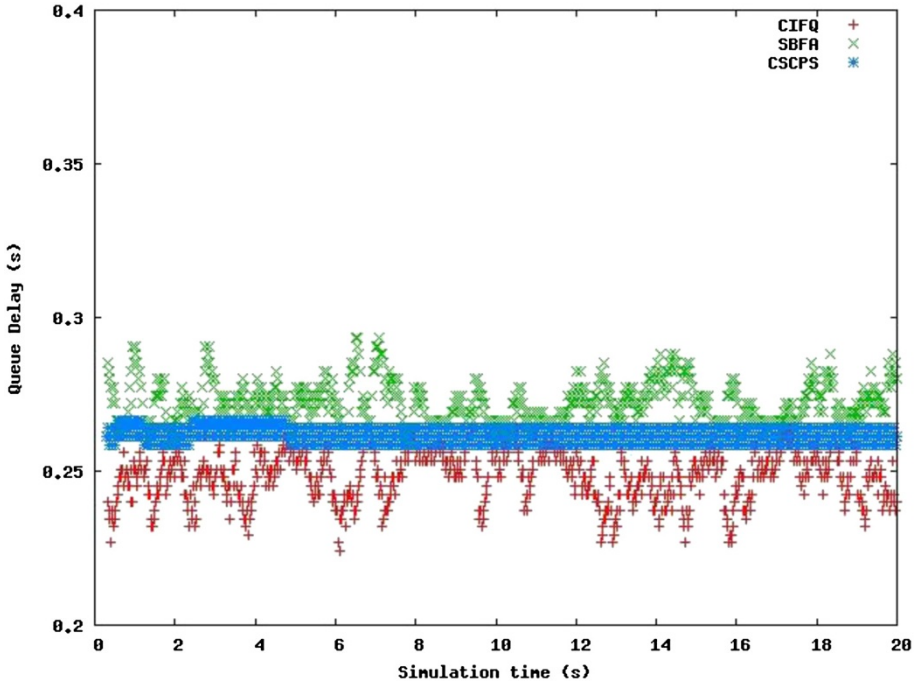

Figure 13 Comparison of queue delay among three algorithms for the flow in error-free channel using two-state channel model. CSCPS provides steadier queue delay.

data immediately to reduce the average delay time and maintain the fairness, but it should be avoided for the flows in the error-free channel to improve the quality of service for multimedia streaming applications.

In this simulation, we demonstrate the steady queue delay and jitter of the flows in the error-free channel by comparing the variety of queue delay and jitter statistics between CSCPS and other algorithms. The simulation lasts for 100 s, and there are five 1.5 Mbps UDP/CBR traffic flows, two flows in the error-free channel and three flows in the two-state channel error model. One flow in the error-free channel is selected to observe the distribution of queue delay in the first 20 s as shown in Figure 13. The percentage of the difference in queue delay between the instantaneous queue delay and the mean reflects the steady property of an algorithm. We first compare the steady queue delay property between CSCPS, SBFA, and CIFQ.

The mean queue delay of CSCPS is 0.262 s. There is a 99% difference between the instantaneous and mean queue delays within 0.01 s which is the highest percentage among three algorithms. This shows that CSCPS controls the queue delay in a tiny region without a dramatic variation since CSCPS takes the channel condition into account of the compensation scheme. If the flow suffers from errors or a leading flow is selected to forward its packets, the lagging flow with maximum service tag is compensated first. If there are no backlogged lagging flows in the system, the system serves the flow with minimum service tag. In other words, leading flows can release their resource to lagging flows quickly if lagging flows exist. Therefore, flows preserving error-free channel have a constant rate to forward their packets to achieve steady queue delay and jitter.

The mean queue delay of SBFA is 0.272 s which is the highest among the three algorithms. There is an 81% difference between the instantaneous and mean queue delays within 0.01 s. Because SBFA algorithm reserves the portion of shared bandwidth and uses a fractional bandwidth for compensation, the weights of flows perceiving error-free channel are beaten and the flows experience a longer queue delay than those in other algorithms when the system compensates lagging flows. The major reason for unsteady queue delay and jitter of SBFA is that lagging flows have higher priority than leading and sync flows to let the system serve for lagging flows first. The mean queue delay of CIFQ is 0.25 s which is the lowest

Table 5 Comparison of queue jitter statistics for the flow in error-free channel using two-state channel model

	Mean (ms)	Variance
CIFQ	1.42	1.52×10^{-6}
SBFA	1.23	5.73×10^{-7}
CSCPS	1.05	2.92×10^{-7}

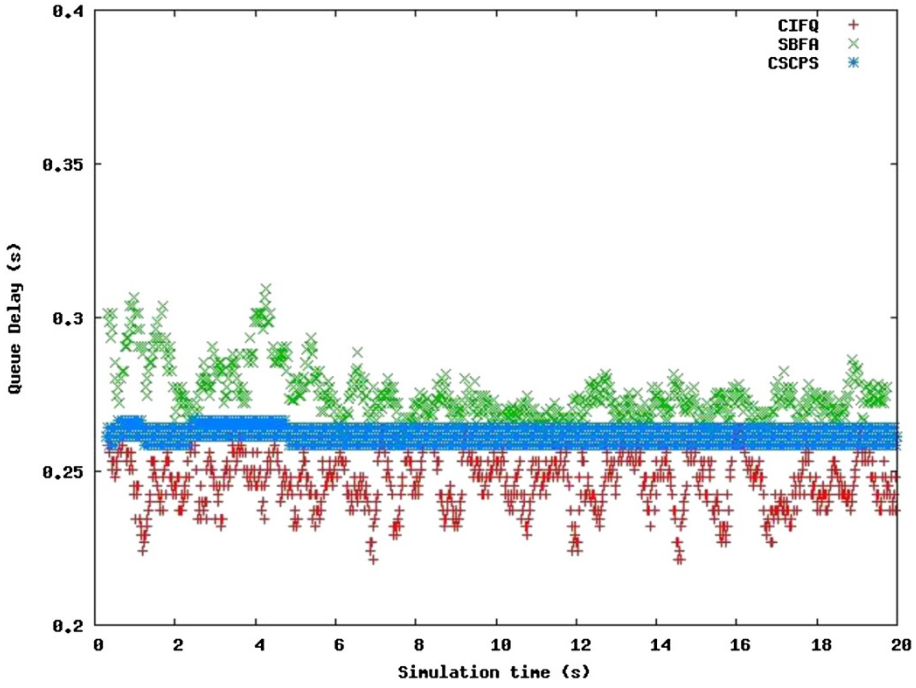

Figure 14 Comparison of queue delay among three algorithms for the flow in error-free channel using AMC technique. CSCPS provides a steadier queue delay.

among the three algorithms, but there is only a 77% difference between the instantaneous and mean of queue delays within 0.01 s. CIFQ provides shorter queue delay while it compensates the lagging flow, but it provides longer delay when the flow experiences burst in the error channel to achieve a graceful degradation for the flows in the error-free channel. Flows preserving error-free channel can get additional slots from lagging flows to forward its packets. This feature allows the average queue delay of the CIFQ algorithm to be the shortest among the three algorithms for the flow in the error-free channel.

To show the steady jitter property precisely, Table 5 presents the queue jitter statistics in term of mean and variance of the flow among the three algorithms. Queue jitter might be negative because it is a variation of delay between two consecutive packets in the queue. In Table 5, all jitter values turn to be unsigned values in calculating the mean and variance. The mean of jitter of CSCPS is only 1.05 ms, which is much lower than in other algorithms. Lower mean and variance of jitter represent a steadier queue jitter for the algorithm. Therefore, CSCPS has the steadiest queue jitter property than other algorithms for the flows in the error-free channel.

4.7 Steady queue delay and jitter for the flow in error-free channel using AMC technique

In this simulation, we demonstrate the steady queue delay and jitter of the flows in the error-free channel by comparing the variety of queue delay and jitter statistics between CSCPS and other algorithms. The simulation uses five 1.5 Mbps UDP/CBR traffic flows in the system with an AMC scheme using four MCS levels. There are two flows in the error-free channel and three flows in the uniform distribution error model. One flow in the error-free channel is selected to observe the distribution of queue delay in the first 20 s as shown in Figure 14. The experimental result is the same with that in Subsection 4.6. Table 6 shows the queue jitter statistics in term of mean and variance of the flow among the three algorithms. The mean of jitter of CSCPS is only 1.108 ms, which is much lower than that in other algorithms. Lower mean and variance of jitter represent a steadier queue jitter for the algorithm. Therefore, CSCPS has the steadiest queue jitter property than other algorithms for the flows in the error-free channel using AMC technique.

Table 6 Comparison of queue jitter statistics for the flow in error-free channel using AMC

	Mean (ms)	Variance
CIFQ	1.545	2.43×10^{-6}
SBFA	1.292	2.46×10^{-6}
CSCPS	1.108	1.29×10^{-6}

5. Conclusions

Packet fair queueing has long been popular and provides delay bound and fairness in wireline networks. While these algorithms cannot be applied directly to wireless networks because there are some characteristics of location-dependent and burst channel errors in wireless networks, this may result unfairness among flows and makes it difficult to provide delay guarantees.

In this paper, we provided a general and practical packet scheduling algorithm which can take the wireless channel condition into account of packet scheduling in a general wireless transmission. CSCPS provides delay guarantees and short-term fairness for the error-free system and long-term fairness for flow perceiving channel errors. Aside from the mentioned properties, CSCPS improves the flows in the clean channel, which are affected by flows perceiving channel errors, and makes the queue delay and jitter of flows in the clean channel steady. Furthermore, the queue delay of flow, which is the worst case delay bound for packets, is unaffected by lagging flows.

The further work of CSCPS is to determine the interval time of flow precisely. For traffic, such as hypertext transfer protocol (http) traffic and file transfer protocol (FTP) traffic, the rate of packet arrival in the system is not good as that in UDP/CBR traffic with a constant bitrate. When errors happen, it is difficult to set the virtual time of packet for http and FTP traffic and for other traffic types. This may result in later packet arrival in the system which has higher priority than the packet of flow that suffered from error previously.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

This work was sponsored in part by National Science Council Taiwan under project NSC 101-2221-E-008-015-MY3.

Received: 30 October 2011 Accepted: 1 March 2013
Published: 17 May 2013

References

1. TE Kolding, F Frederiksen, PE Mogensen, Performance aspects of WCDMA system with high speed downlink packet access (HSDPA). Proc. 56th IEEE Vehicular Technol. Conf. (VTC) **1**, 477–481 (2002)
2. SH Wu, YL Chung, Z Tsai, *A study of dynamic network selection for HSPA dual-network users, International Conference on Information Networking (ICOIN)* (IEEE, New York, 2011), pp. 329–334
3. C So-In, R Jain, A-K Al-Tamimi, Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey. IEEE J. Selected Areas Commun. **27**, 156–171 (2009)
4. JM Westall, JJ Martin, Performance characteristics of an operational WiMAX network. IEEE Trans. Mobile Comput. **10**, 941–953 (2011)
5. S Abedi, Efficient radio resource management for wireless multimedia communications: a multidimensional QoS-Based packet scheduler. IEEE Trans. Wirel. Commun. **4**, 2811–2822 (2005)
6. B Ai-Manthari, H Hassanein, N Ali, N Nasser, Fair class-based downlink scheduling with revenue considerations in next generation broadband wireless access systems. IEEE Trans. Mob. Comput. **8**, 721–734 (2009)
7. P Ameiqeiras, J Wigard, P Mogensen, Performance of the M-LWDF scheduling algorithm for streaming services in HSDPA. Proc IEEE Vehicular Technol. Conf. VTC2004-Fall **2**, 999–1003 (2004)
8. RO Garroppo, S Giordano, D Iacono, L Tavanti, Tavanti, Game theory and time utility functions for a radio aware scheduling algorithm for WiMAX networks. Wireless Netw **17**, 1441–1459 (2011)
9. R Heidary, M Mehrjoo, *Delay and rate based multichannel scheduling for heterogeneous traffic* (International Symposium on Computer Networks and Distributed Systems, IEEE, New York, 2011), pp. 187–192
10. WS Jeon, DG Jeong, B Kim, Packet scheduler for mobile internet services using high speed downlink packet access. IEEE Trans. Wirel. Commun. **3**, 1789–1801 (2004)
11. O Jo, JW Son, DH Cho, An enhanced packet scheduling algorithm combined with HARQ for HSDPA system. IEEE Commun. Lett. **12**, 247–249 (2008)
12. M Lundevall, B Olin, J Olsson, J Eriksson, F Eng, Streaming applications over HSDPA in mixed service scenarios. Proc. 60th IEEE Vehicular Techno. Conf. (VTC) **2**, 841–845 (2004)
13. S Manistis, E Nikolouzou, I Vemieris, QoS issues in the converged 3G wireless and wired networks. IEEE Commun. Mag. **40**, 44–53 (2002)
14. H Shao, C Shen, D Gu, J Zhang, P Orlik, *Dynamic resource control for high-speed downlink packet access wireless channel. 23th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (IEEE, New York, 2003), pp. 838–843 9. 838–849
15. C So-In, R Jain, A-K Al-Tamimi, A Scheduler for unsolicited grant service (UGS) in IEEE 802.16e mobile WiMAX networks. IEEE Syst. J **4**, 487–494 (2010)
16. L Xu, X Shen, JW Mark, Dynamic bandwidth allocation with fair scheduling for WCDMA system. IEEE Trans. Wireless Commun. **9**, 26–32 (2002)
17. L Tao, F Yu, Delay-Jitter aware slot assignment for real-time applications in wireless. Comput. Commun. **35**, 1967–1982 (2012)
18. L Zhang, L Zheng, KS Ngee, Effect of delay and delay jitter on voice/video over IP. Comput. Commun. **25**, 863–873 (2002)
19. V Bharghavan, S Lu, T Nandagopal, Fair queuing in wireless networks: issues and approaches. IEEE Pers. Commun. **6**, 44–53 (1999)
20. M Srivastava, C Fragouli, V Sivaraman, Controlled multimedia wireless link sharing via enhanced class-based queueing with channel-state-dependent packet scheduling. Proc. IEEE INFOCOM **2**, 572–580 (1998)
21. P Goyal, HM Vin, H Cheng, Start-time fair queueing: a scheduling algorithm for integrated service packet switching networks. IEEE/ACM Trans. Netw. **5**, 690–704 (1997)
22. A Parekh, R Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Trans. on Networking **1**, 344–357 (1993)
23. EHK Wu, HT Lai, MF Tsai, CF Chou, Low latency and efficient packet scheduling for streaming applications. Proc. IEEE Int. Conf. Commun. **4**, 1963–1967 (2004)
24. S Lu, V Bharghavan, R Srikant, Fair scheduling in wireless packet networks. IEEE/ACM Trans. on Networking **7**, 473–489 (1999)
25. NH Vaidya, P Bahl, S Gupta, Distributed fair scheduling in a wireless LAN. IEEE Trans. Mobile Comput. **4**, 616–629 (2005)
26. MR Jeong, H Morikawa, T Aoyama, *Fair scheduling algorithm for wireless packet networks. 1999 International Workshops on Parallel Processing* (IEEE, New York, 1999), pp. 280–285
27. AKF Khattab, AKF Khattab, KMF Elsayed, *Channel-quality dependent earliest deadline due fair scheduling schemes for wireless multimedia networks. MSWiM '04 Proceedings of the 7th ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems* ((ACM, New York, 2004), pp. 31–38
28. P Lin, B Benssou, QL Ding, CS-WFQ KC Chua, *a wireless fair scheduling algorithm for error-prone wireless channels. Ninth International Conference on Computer Communications and Networks* (IEEE, New York, 2000), pp. 276–281
29. TS Ng, I Stoica, H Zhang, Packet fair queueing algorithms for wireless networks with location-dependent errors. Proc. IEEE INFOCOM **3**, 1103–1111 (1998)
30. P Ramanathan, P Agrawal, *Adapting packet fair queueing algorithms to wireless networks. MobiCom '98 Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking* (ACM, New York, 1998), pp. 1–9
31. P Bender, P Black, M Grob, R Padovani, N Sindhushayana, A Viterbi, CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users. IEEE Commun. Mag. **38**, 70–77 (2000)

32. H Kim, Y Han, A proportional fair scheduling for multicarrier transmission systems. IEEE Commun. Lett. **9**, 210–212 (2005)
33. B Ai-Manthari, N Nasser, H Hassanein, Downlink scheduling with economic considerations for future wireless networks. IEEE Trans. Veh. Technol. **58**, 824–835 (2009)
34. M Andrews, K Kumaran, K Ramanan, A Stolyar, P Whiting, R Vijayakumar, Providing quality of service over a shared wireless link. IEEE Commun. Mag. **39**, 150–154 (2001)
35. AL Stolyar, K Ramanan, Largest weighted delay first scheduling: large deviations and optimality. Annals of Applied Probability **11**, 1–48 (2001)
36. P Jose, *Packet scheduling and quality of service in HSDPA* (Aalborg University, PhD dissertation, 2003)
37. KMF Elsayed, AKF Khattab, Channel-aware earliest deadline due fair scheduling for wireless multimedia network. Wirel. Pers. Commun. **38**, 233–252 (2006)
38. Nsnam, *The network simulator ns-2*, 2007. http://nsnam.isi.edu/nsnam/index.php/Main_Page

doi:10.1186/1687-1499-2013-131
Cite this article as: Chen *et al.*: **Channel condition self-clocked packet scheduling scheme for wireless networks.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:131.