Packet Scheduling in 3.5G High-Speed Downlink Packet Access Networks: Breadth and Depth

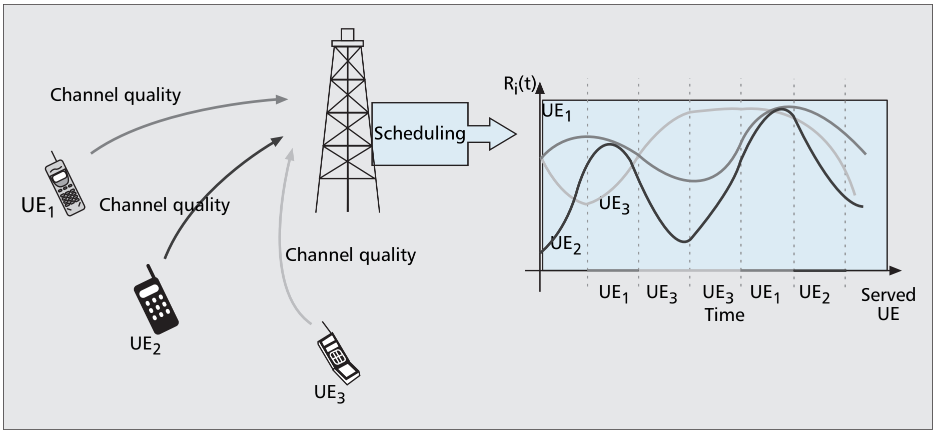
Bader Al-Manthari and Hossam Hassanein, Queen's University
Nidal Nasser, University of Guelph

Abstract

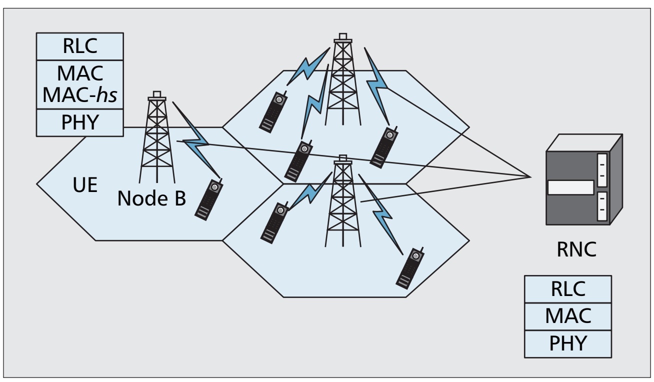
Forecasts for emerging mobile device markets anticipate that bandwidth will be squeezed by demanding applications like multimedia on demand. This will spur the need for data rates beyond what the upcoming 3G wireless cellular systems such as UMTS can offer. To boost the support for such high data rates, HSDPA, labeled as a 3.5G wireless system, has been introduced in Release 5 of UMTS technical specifications. HSDPA is a definite step toward meeting the "anywhere, anytime, and in any form" 4G communication concept. HSDPA promises a peak data rate of up to 10 Mb/s, five times larger than the data rate offered by 3G systems. In order to support such high data rates, HSDPA relies on many new technologies, among which is packet scheduling. In this article we provide breadth and depth related issues of packet scheduling in HSDPA, discuss state-of-the-art HSDPA scheduling algorithms in terms of their objectives, advantages, and limitations, and suggest further research issues that need to be addressed. In addition, we propose a packet scheduling algorithm for data traffic in HSDPA. Simulation results demonstrate the effectiveness of the proposed algorithm.

The evolution of today's wireless technology began in the early 1980s with the introduction of the first mobile phones. These systems utilized analog interface technology and supported voice-only capabilities. This technology is still used in some parts of the world; however, it is limited in bandwidth and is low in quality. With the high demand for cell phones, and increased need for enhanced quality and more features, the second generation (2G) was introduced. 2G is primarily voice only, but it does provide higher bandwidth, better voice quality, and limited data services that use packet data technology. Extension of the 2G system is introduced in 2.5G systems such as General Packet Radio Service (GPRS). The main feature of 2.5G systems is the data packet service enhancement. Furthermore, the continuous success of mobile communication systems as well as its consequences in terms of the need for better quality of service (QoS), more efficient systems and more services have led to the development of the third-generation (3G) mobile telecommunications system: Universal Mobile Telecommunications System (UMTS). UMTS is the standard version of 3G mobile systems in Europe [1]. It promises a transmission rate of up to 2 Mb/s, which makes it possible to provide a wide range of multimedia services including video telephony, paging, messaging, Internet access and broadband data. However, it is expected that there will be a strong demand for multimedia applications which require higher data rates above 2 Mb/s in cellular systems, especially in the downlink, where mobile users will enjoy high-speed Internet access and broadcast services. In order to offer such broadband packet data transmission services, High Speed Downlink Packet Access (HSDPA), labeled a 3.5G wireless system, has been introduced in Release 5 of UMTS [2]. HSDPA is expected to achieve higher performance with a peak data rate of about 10 Mb/s. In HSDPA, high-speed packet transmission is possible by time-sharing a commonly used data channel among access users, called the high-speed downlink shared channel (HS-DSCH). UMTS already includes a downlink shared channel, but HSDPA extends this concept to provide significantly higher throughput and hence more efficient use of the radio spectrum. The high transmission rate of HSDPA will also allow UMTS to support new high-data-rate services and improve the QoS of already existing ones. HSDPA relies on new technologies that make it possible to achieve such a high data rate. These new technologies include adaptive modulation and coding, hybrid automatic repeat request, fast cell selection, and fast packet scheduling. Packet scheduling is one of the key design features of HSDPA. A packet scheduler controls the allocation of channels to users within the coverage area of the system by deciding which user should transmit during a given time interval. Therefore, to a large extent, it determines the overall behavior of the system. One important factor that has been added to the scheduling problem in HSDPA is the channel conditions of mobile users. Mobile users experience varying channel conditions due to mobility, interference caused by other users in the system, distance from the base station, which is known as the Node B in 3G and 3.5G systems, and other factors. Varying channel conditions affect the supportable data rates of users (i.e., the amount of data rate they can obtain from the Node B) from time to time. The packet scheduler in HSDPA should track the instantaneous channel conditions of the users and select for transmission those who are experiencing good channel conditions in order

 0890-8044/07/$20.00



■ Figure 1. *Exploiting the user channel quality for scheduling decisions.*



■ Figure 2. *The MAC-hs at the Node B in HSDPA.*

to maximize the system throughput. However, this raises the issue of fairness, as users who have bad channel conditions may not get served, and thus end up having very low average throughputs.

This article is organized as follows. We first discuss packet scheduling in HSDPA, its design and architecture. Then we provide an overview of different packet scheduling algorithms in HSDPA. Next we propose a packet scheduling algorithm for HSDPA and evaluate its performance.

Packet Scheduling in HSDPA

One of the most important features of HSDPA is packet scheduling. The main goal of packet scheduling is to maximize system throughput while satisfying the QoS requirements of users. The packet scheduler determines to which user the shared channel transmission should be assigned at a given time. In HSDPA the packet scheduler can exploit short-term variations in the radio conditions of different users by selecting those with favorable instantaneous channel conditions for transmission, which is illustrated in Fig. 1. This idea is based on the fact that good channel conditions allow for higher data rates (R) by using a higher-order modulation and coding schemes [2] and thus result in increased system throughput.

In order to quickly obtain up-to-date information on the channel conditions of different users, the functionality of the packet scheduler has been moved from the radio network controller (RNC) in UMTS to the medium access control high-speed (MAC-hs) sub ayer at the Node B [2], as shown in Fig. 2. The MAC-hs is a new sublayer added to the MAC layer at the Node B in HSDPA in order to execute the packet scheduling algorithm. In addition, the minimum time transmission interval (TTI) (i.e., the time between two consecutive transmissions) has been reduced from 10 ms in UMTS Release 99 to 2 ms in Release 5, which includes HSDPA. This is because it allows the packet scheduler to better exploit the varying channel conditions of different users in its scheduling decisions and increase the granularity of the scheduling process. It should be noted that favoring users with good channel conditions may prevent those with bad channel conditions from being served, and may therefore result in starvation. A good design of a scheduling algorithm should take into account not only maximization of the system throughput, but also being fair to users who use the same service and pay the same amount of money. That is, scheduling algorithms should balance the trade-off between maximizing throughput and fairness.

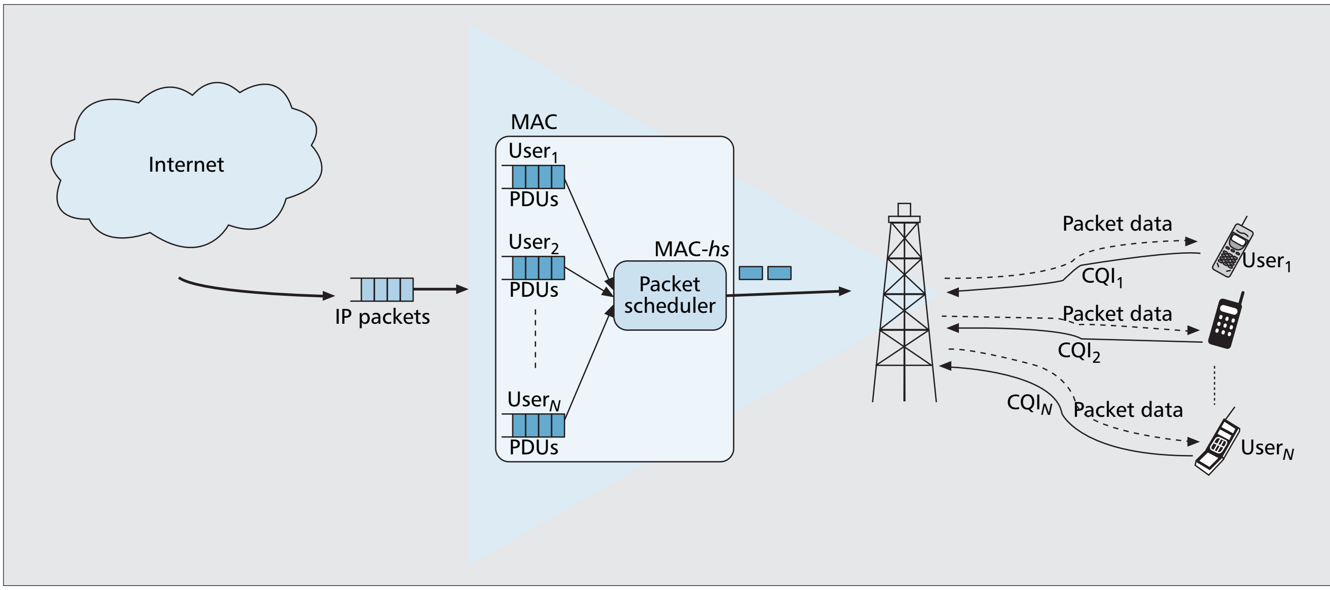
HSDPA Packet Scheduler Model and Process

In this section we briefly describe the packet scheduler model and how it works in HSDPA. As mentioned above, the packet scheduler for HSDPA is implemented at the MAC-hs layer of Node B. Node B can serve N users simultaneously, $N \geq 1$, and selects one transmission user in a slot of fixed time duration. Also, and without loss of generality, it is assumed that each user has one connection request. Thus, a Node B maintains one queue for every user, as shown in Fig. 3. Upon call arrival, the radio link controller (RLC) layer receives traffic in the form of IP packets from higher layers, which are segmented into fixed-size protocol data units (PDUs). These PDUs are stored in the transmission queue of the corresponding user in a first-in first-out fashion. Subsequently, the PDUs are transmitted to the appropriate mobile user according to the adopted scheduling discipline.

The packet scheduler works as follows. Every TTI, each user regularly informs the Node B of its channel quality condition by sending a report known as a channel quality indicator (CQI) in the uplink to the Node B. The CQI contains information about the instantaneous channel quality of the user. This information includes the size of the transport block the Node B should send to the user, the number of simultaneous channel codes, and the type of modulation and coding schemes the user can support. Node B then selects the appropriate mobile user according to the adopted scheduling discipline and send data to the selected user at the specified rates. The user is able to measure its current channel conditions by measuring the power of the received signal from the Node B and then using a set of models described in [3] to determine its current supportable data rates (i.e., the rates at which it can receive data from the Node B given its current channel condition). Therefore, users with good channel conditions will enjoy potentially higher supportable data rates by using higher modulation and coding rates, whereas users with bad channel conditions will experience lower data rates instead of adjusting their transmission power.

Scheduling Algorithms in HSDPA

HSDPA is designed to support non-real-time applications (e.g., interactive and background) and also to some extent real-time applications (e.g., streaming). Since real-time applications have different QoS constraints than non-real-time applications, the design of scheduling algorithms for real-time



■ Figure 3. *The packet scheduler model in HSDPA.*

applications should be different from that for non-real-time applications. Therefore, scheduling algorithms can be classified into two groups: real-time (RT) and non-real-time (NRT) scheduling algorithms. In addition, scheduling algorithms within each group can be characterized by three factors [4]:

- **Scheduling frequency**: The rate at which users are scheduled. Scheduling algorithms that make use of the channel conditions of users need to make decisions every TTI to better exploit fast variation of channel conditions and are therefore called fast scheduling algorithms. Other scheduling algorithms that do not make a decision every TTI are called slow scheduling algorithms.
- **Service order**: The order in which users are served. For example, some scheduling algorithms schedule users based on their channel conditions, whereas others schedule them randomly.
- **Allocation method**: The method of allocating resources. For instance, some scheduling algorithms provide the same data amount for all users per allocation interval, while others give all users the same (time, code, or power) per allocation interval.

In the following sections different NRT and RT scheduling algorithms for HSDPA are discussed in terms of their objectives, advantages, and limitations.

Non-Real-Time Scheduling Algorithms for Data Applications in HSDPA

The time-shared nature of the HS-DSCH in HSDPA makes it very well suited for data traffic (i.e., interactive and background), since NRT applications do not require QoS guarantees as do RT services. In this section we discuss several scheduling algorithms proposed for NRT applications. These algorithms have been organized based on their scheduling frequency, fast or slow.

Fast Scheduling Algorithms

Fast scheduling algorithms exploit the instantaneous channel quality of users and try to increase system throughput by favoring those with good channel conditions. Their name comes from the fact that they need to get the CQI of users every TTI (i.e., 2 ms) to schedule them to better exploit channel quality conditions. In this section several fast scheduling algorithms for HSDPA are surveyed.

One of the most well-known scheduling algorithms in HSDPA is Maximum Carrier-to-Interface Ratio (Max CIR) [5]. This algorithm tends to maximize the system throughput by serving, in every TTI, the user with the best channel quality (i.e., maximum current supportable data rate). It can be seen that this algorithm provides high system throughput since only those with high current supportable data rates get served. However, this algorithm has an obvious drawback in that it ignores those users with bad channel conditions, which may lead to starvation. The unfairness issue in this algorithm has led to many proposals for scheduling algorithms that try to distribute resources evenly among users while still using the channel conditions for user selection for transmission. One such proposal is Proportional Fairness (PF) [6]. The PF algorithm tries to increase the degree of fairness among users by selecting those with the largest relative channel quality. Relative channel quality is the instantaneous channel quality condition of the user divided by its current average throughput. Therefore, this algorithm considers not only those users with good channel conditions but also those with low average throughputs by giving them higher priority.

However, according to Bonald [7], the PF scheduler is fair (in terms of the distribution of the users' average throughputs) only in ideal cases where users experience similar channel conditions. However, PF becomes unfair and unable to exploit multiuser diversity in more realistic situations where users usually experience different channel conditions. Therefore, Bonald proposed the Score-Based (SB) algorithm in order to overcome this problem. The SB algorithm computes the rank of each user's current channel condition among the past channel conditions observed over a window of size W. Then it selects for transmission the user with the lowest rank. Therefore, this algorithm selects the user whose current channel condition is high relative to its own rate statistics instead of selecting the one whose channel condition is high relative to its average throughput, as in the PF algorithm. Simulation results show that the SB algorithm solves the problem of the PF algorithm. However, SB is slightly more complex in terms of implementation than the PF algorithm. In addition, choosing the size of W might be another problem. Small values of W might not be appropriate to track the distribution of user channel conditions, while large values of W increase the time it takes to find the rank.

Another proposal is Fast Fair Throughput (FFT) [8]. The purpose of this algorithm is to provide fair throughput distribution while using information about instantaneous channel quality conditions for the selection criterion. This algorithm uses relative channel quality just like PF, but it multiplies it by an equalizer term that ensures a fair long-run throughput distribution among users.

Slow Scheduling Algorithms

Slow scheduling algorithms usually tend to increase fairness among users and hence do not need to compute the CQI of users every TTI. Therefore, these algorithms are known to be slow. Most of these algorithms were proposed for UMTS Release 99 prior to HSDPA, and a few were proposed for other systems.

One of the simplest scheduling algorithms is Round-Robin (RR). Users in this algorithm are served in a round-robin (i.e., cyclic order) manner. This algorithm does not make use of information about the channel quality of users and therefore may offer lower system throughput than fast scheduling algorithms. In fact, simulation results in [9] show that RR results in lower system throughput than the Max CIR and PF algorithms. However, it is a very simple algorithm to implement and use. In addition, it is fair in that it ensures that all users in the system get equal opportunity for transmission regardless of their channel quality conditions.

Another slow scheduling algorithm is Fair Throughput (FT). The goal of this algorithm is to ensure that an equal number of bits is received by each user in the system regardless of their channel quality conditions. A possible implementation of FT would be to select at each TTI the user with the lowest average throughput [9]. As in RR, the CQI of each user is not used in the scheduling decision. This algorithm is fair in terms of the distribution of user throughput since each user gets the same amount of throughput regardless of its channel condition. Similar to RR, it suffers from lower system throughout than fast scheduling algorithms.

Real-Time Scheduling Algorithms for Streaming Applications in HSDPA

Streaming applications impose strict constraints on the network in order to satisfy their QoS requirements. UMTS uses dedicated channels to support streaming applications as well as conversational services. However, since HSDPA offers a high transmission rate, as well as high spectral efficiency over dedicated channels, there has been some research on using the shared channel (HS-DSCH) of HSDPA to support streaming services. Nonetheless, supporting these kinds of services in HSDPA presents many challenges. One of these is that any packet scheduling algorithm must be able to guarantee QoS requirements for streaming users as well as exploiting information about their instantaneous channel conditions in its scheduling decisions. Guaranteeing the QoS requirements of streaming users is a challenging task, especially when the traffic load in the cell is high. Therefore, a call admission control mechanism is needed to determine the level of acceptable traffic load such that the QoS of streaming users is guaranteed [9]. In addition, the QoS constraints must be maintained not only in each cell but also during the handoff process. In this section we discuss some scheduling algorithms for streaming services. All the scheduling algorithms discussed in this section are fast algorithms.

A variation of the well-known PF algorithm is proposed to accommodate real-time packets [8]. The new algorithm is called Modified Proportional Fairness (MPF). In this algorithm, if the delay for user i is below a certain threshold, user i is served according to the PF algorithm; otherwise, user i is served using FFT. MPF gives higher priority to users with packets close to their deadlines in order to prevent them from being discarded.

In [10] a packet scheduling algorithm suitable for real-time traffic in HSDPA is proposed. The algorithm first calculates two priorities for each user: the delay and throughput priorities. The delay priority gives more weight to users whose packet delays are near a predefined threshold. On the other hand, the throughput priority gives more weight to users whose throughputs are below a certain threshold. These two priorities are then combined to calculate the total priorities of the users. The users' packets are then ordered based on their total priorities and sent accordingly to a buffer of a certain size. The packets in the buffer are assigned wireless resources such as codes for transmission based on the users' channel quality conditions according to formulas explained in [10].

Another proposal for real-time traffic is Max CIR with Early Delay Notification (EDN) [11]. This algorithm tries to maximize the system's throughput by scheduling users using Max CIR as long as their packets' delays are below a certain threshold. If the packets delays of one or more users exceed a certain threshold, then the packets that have been queued the longest time are served first. Simulation results show that this algorithm achieves high system throughput and better packet dropping in most cases.

A comparison of the scheduling algorithms discussed based on the aforementioned factors is provided in Table 1.

A Fast Packet Scheduling Algorithm

In this section we propose a fast packet scheduling aimed at providing a minimum throughput guarantee for HSDPA users by considering their instantaneous channel conditions as well as their current average throughputs and hence giving higher priority to those with average throughputs below a certain threshold. We propose the Channel Quality-Based Minimum Throughput Assurance (CQ-BMTA) scheduling algorithm. We evaluate its performance by simulating the MAC-hs packet scheduler of HSDPA. A performance comparison between CQ-BMTA and the well-known Max CIR and PF algorithms is provided.

The CQ-BMTA Algorithm

CQ-BMTA aims to enhance the average throughput for each user by giving more priority for those with low average throughputs. In the proposed algorithm, the user with the highest priority is selected for transmission where the priority for user i at time t is calculated as follows:

$$P_i(t) = \begin{cases} CQI & \text{if } S_i(t) \geq C \\ CQI \cdot W & \text{Otherwise} \end{cases}$$

where CQI is the CQI for a user that represents the current channel condition for this user, $S_i(t)$ is the average throughput for user i up to time t, C is a predefined minimum throughput, and $W = C/S_i(t)$.

The proposed algorithm prioritizes users based on their radio channel quality. That is, users are served based on their channel quality condition represented by their CQIs as long as they are achieving high average throughputs. However, if users' average throughputs start dropping below C, their CQIs are multiplied by a weight (W) that is inversely proportional to their throughputs (i.e., W increases as a user's average throughput decreases). Therefore, by introducing this weight, higher priorities are given to users with low average throughput, which increases the degree of fairness in the system.

A user's average throughput at time t is calculated by using an exponentially smoothed filter as follows:

Scheduling method	Services supported	Scheduling frequency	Service order	Allocation method
Max CIR	NRT	Fast	Highest channel quality	Same resources (time, code, or power)
Proportional Fairness (PF)	NRT	Fast	Highest relative channel quality	Same resources (time, code, or power)
Fast Fair Throughput (FFT)	NRT	Fast	Highest relative channel quality and average bit rate	Same resources (time, code, or power)
Score Based (SB)	NRT	Fast	Highest rank of the current data rate among the past data rates	Same resources (time, code, or power)
Round Robin (RR)	NRT	Slow	Round-robin	Same resources (time, code, or power)
Fair Throughout (FT)	NRT	Slow	Lowest average throughput	Same data amount
Modified Proportional Fairness (MPF)	RT	Fast	Highest relative channel quality	Same resources (time, code, or power)
Delay and Throughput Priorities	RT	Fast	Channel quality condition, throughput, and packet delay	Resources according to channel quality condition
Max CIR with EDN	RT	Fast	Channel quality and packet delay	Same resources (time, code, or power)

■ Table 1. *Comparison of scheduling algorithms for HSDPA.*

$$S_i(t) = \begin{cases} \left(1 - \frac{1}{t_c}\right) S_i(t-1) + \frac{R_i(t)}{t_c} & \text{if } i \text{ is served in slot } t \\ \left(1 - \frac{1}{t_c}\right) S_i(t-1) & \text{Otherwise} \end{cases}$$

where t_c is the time constant of the filter and is set equal to 1000 time slots [12], and $R_i(t)$ is the current data rate a user can support at time t giving its current channel condition represented by its CQI.

Performance Evaluation

We evaluate and compare the performance of our proposed scheduling algorithm by means of dynamic simulation with the help of Network Simulator (ns-2) and its Enhanced UMTS Radio Access Network Extensions (EURANE) [13]. Figure 4 shows the simulation model. We simulated a one-cell case and, for simplicity, did not consider handoff. The cell radius is 1 km. The Node B is located at the center of the cell. Therefore, only one Node B is involved in allocating the radio resources. Users are connected to the Node B on the downlink by an HS-PDSCH, which is the actual physical channel for HSDPA, and on the uplink by a high-speed physical dedicated control channel (HS-PDCCH), which is used to send the users' current estimates of their channel conditions to the Node B. The Node B is connected to an RNC by a duplex link of 622 Mb/s bandwidth and 15 ms delay. The RNC is connected to the serving GPRS support node (SGSN) by a duplex link with 622 Mb/s and 15 ms delay. The SGSN is connected to the gateway GPRS support node (GGSN) by a duplex link of 622 Mb/s bandwidth and 10 ms delay (the SGSN and GGSN are part of the core network and are used to support packet-switched services). The core network is connected to the Internet by a duplex link of 100 Mb/s bandwidth and 10 ms delay. On the Internet, there is an FTP server connected to it by a duplex link of 100 Mb/s and 35 ms delay. All of these values can be found in [14]. Each user sends a request for one FTP file; the user's connection terminates after the download is complete. The size of each FTP file is 0.5 Mbyte.

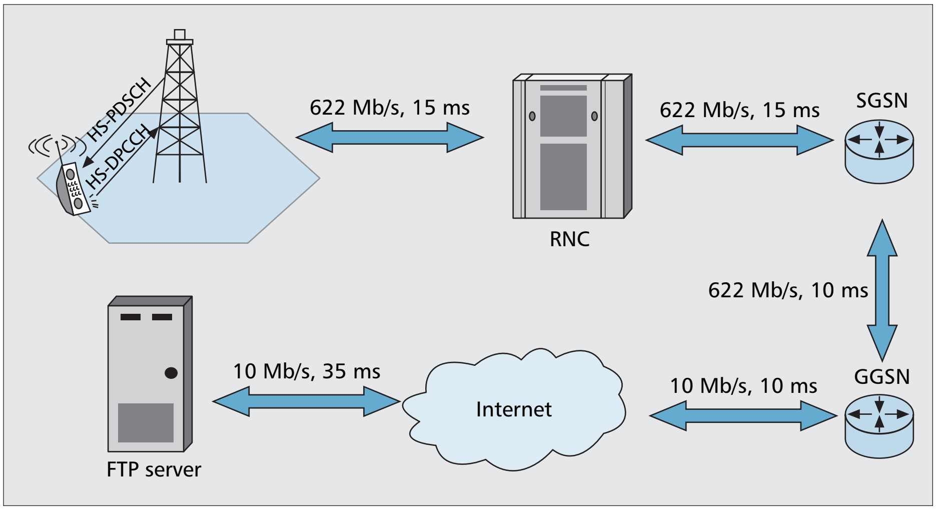
At initialization, N users are uniformly distributed in the cell. Every mobile user moves inside the cell with a constant speed of 3 km/h, which is the recommended value for the pedestrian A environment by the Third Generation Partnership Project (3GPP) [14]. In our simulation the channel model consists of five parts: distance loss, shadowing, multipath fading, intracell interference, and intercell interference. Depending on the values of these parts, the signal-to-noise ratio (SNR) (the signal strength relative to background noise) is calculated to determine how strong the signal is from the Node B to the user. The formulas for the propagation model can be found in [14].The simulation time step is one time frame, which is 2 ms, and the simulation time is 100 s.

Figure 5 depicts the cumulative distribution function (CDF) of the average throughputs of 25 users. The steeper the CDF curve, the fairer the algorithm, because this means users' average throughputs are distributed over a small interval (i.e., users get relatively equal average throughputs). CQ-BMTA has the steepest slope since it prevents users from achieving low average throughputs by increasing their priority by W. Max CIR has the flattest slope since it is an unfair algorithm; only those with the best channel conditions are served at the expense of ignoring the rest. This results in unfair distribution of the users' average throughputs.

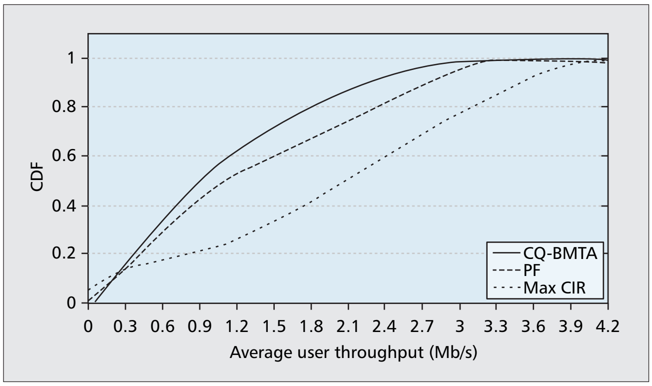
User satisfaction is shown in Fig. 6. In this experiment a user is satisfied if its average throughput is greater than or equal to 128 kb/s. As we can see, the CQ-BMTA algorithm outperforms Max CIR and PF because it increases the chance of those with low throughput getting served by multiplying their priorities by a weight that increases as the difference between the minimum throughput and their throughput increases. The figure also shows that Max CIR achieves the worst performance in terms of user satisfaction because it only serves those users with the best channel conditions while ignoring the rest.

Conclusions

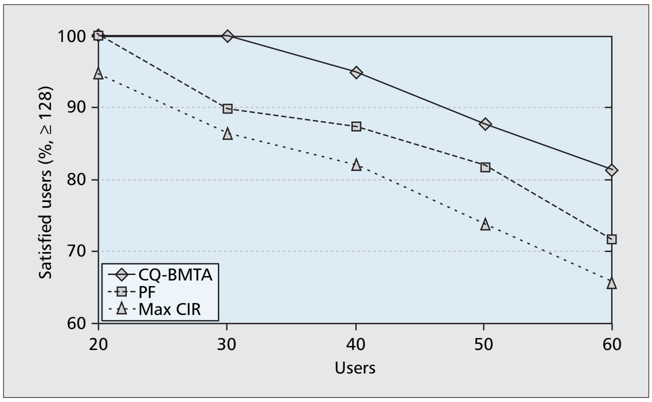
High Speed Downlink Packet Access has been introduced in order to boost support for high data rates beyond those 3G/UMTS can offer. It promises a data rate of up to 10 Mb/s, which allows it to support new multimedia applications and



■ Figure 4. *The simulation model.*



■ Figure 5. *The distribution of users' average throughputs.*



■ Figure 6. *User satisfaction.*

improve the QoS of already existing ones. HSDPA relies on new technologies to help achieve the high data rates it offers, among which is packet scheduling. The functionality of packet scheduling is crucial to the operation of HSDPA since it controls the distribution of scarce radio resources among mobile users. In this article we present the architecture design of a packet scheduling scheme in HSDPA and discuss several scheduling algorithms introduced in the literature. The algorithms are classified based on the type of applications HSDPA can support. We find that the common objective of all the scheduling algorithms is to balance the trade-off between throughput and fairness. Following the same objective, we propose a novel packet scheduling algorithm we call Channel-Quality-Based Minimum Throughput Assurance. By simulation we demonstrate the viability and strength of our proposed algorithm. CQ-BMTA outperforms the Max CIR and PF schemes in terms of providing minimum throughput guarantees, and therefore has a greater degree of fairness.

References

[1] H. Kaaranene *et al.*, *UMTS Networks, Architecture, Mobility, and Services*, 1st ed., Wiley, 2001.

[2] 3GPP TS 25.308, "High Speed Downlink Packet Access (HSDPA); Overall Description," Rel. 5, Mar. 2003.

[3] 3GPP TS25.214, "Physical Layer Procedures," Rel. 5, v. 5.5.0, June 2003.

[4] T. Kolding *et al.*, "High Speed Downlink Packet Access: WCDMA Evolution," *IEEE Vehic. Tech. Soc. News*, vol. 50, Feb. 2003, pp. 4–10.

[5] S. Borst, "User-Level Performance of Channel-Aware Scheduling Algorithms in Wireless Data Networks," *Proc. IEEE INFOCOM*, vol. 1, Mar. 2003, pp. 321–31.

[6] A. Jalali, R. Padovani, and R. Pankaj, "Data Throughput of CDMA-HDR a High Efficiency-High Date Rate Personal Communication Wireless System," *Proc. IEEE VTC*, May 2000, pp. 1854–58.

[7] T. Bonald, "A Score-Based Opportunistic Scheduler for Fading Radio Channels," *Proc. Euro. Wireless*, Sept. 2002, pp. 2244–48.

[8] G. Barriac and J. Holtzman, "Introducing Delay Sensitivity into the Proportional Fair Algorithm for CDMA Downlink Scheduling," *Proc. IEEE 7th Int'l. Symp. Spread Spectrum Techniques and Apps.*, vol. 3, 2002, pp. 652–56.

[9] P. Jose, *Packet Scheduling and Quality of Service in HSDPA*, Ph.D. dissertation, Aalborg Univ., Oct. 2003.

[10] H. Zeng *et al.*, "Packet Scheduling Algorithm Considering Both the Delay Constraint and User Throughput in HSDPA," *Proc. Int'l. Conf. Commun., Circuits and Sys.*, vol.1, May 2005, pp. 387–82.

[11] A. Golaup, O. Holland, and A. Aghvami, "A Packet Scheduling Algorithm Supporting Multimedia Traffic over the HSDPA Link Based on Early Delay Notification," *Proc. 1st Int'l. Conf. Multimedia Services Access Networks*, June 2005, pp. 78–82.

[12] Y. Ofuji, S. Abeta, and M. Sawahashi, " Comparison of Packet Scheduling Algorithms Focusing on User Throughput in High Speed Downlink Packet Access," *IEICE Trans. Commun.*, vol. E86-B, no. 1, Jan 2003, pp. 132–39.

[13] Enhanced UMTS Radio Access Network Extensions for NS2, http://www.tiwmc.nl/eurane/, July 2005.

[14] Deliverable D3. 2v2, "End-to-end Network Model for Enhanced UMTS," http://www.ti-wmc.nl/eurane/, Oct. 2006.

Biographies

BADER AL-MANTHARI (manthari@cs.queensu.ca) is a Ph.D. candidate and a member of the Telecommunication Research Laboratory at the School of Computing at Queen's University. His research interests include radio resource management in next-generation wireless cellular networks, wireless ad hoc and sensor networks, and performance evaluation of communication protocols and algorithms. He received his B.Sc. (Honors) in computing with minor concentration in mathematics and his M.Sc. from Queen's University in 2004 and 2005, respectively.

NIDAL NASSER (nasser@cis.uoguelph.ca) is an assistant professor in the Department of Computing and Information Science at the University of Guelph, Ontario, Canada. He has authored several journal publications, refereed conference publications, and four book chapters. He has been a member of the technical program and organizing committees of several international IEEE conferences and workshops. His current research interests include multimedia wireless networks, mobile sensor networks and heterogeneous wireless networks, with special emphasis on radio resource management techniques, performance modeling, and provisioning QoS.

HOSSAM HASSANEIN [SM] (hossam@cs.queensu.ca) is a professor in the School of Computing at Queen's University in the areas of broadband and wireless network architecture, protocols, control, and performance evaluation. He is the founder and director of the Telecommunication Research (TR) Laboratory (http://www.cs.queensu.ca/~trl) at Queen's. He has more than 250 publications in reputable journals and conferences. He serves as Secretary of the IEEE Communication Society Technical Committee on Ad Hoc and Sensor Networks. He serves on the organizing and program committees of numerous international conferences.