# Issues in Scheduling Multimedia Traffic over the High Speed Downlink Packet Access (HSDPA) link of UMTS

A. Golaup, O. Holland, and A. H. Aghvami

Centre for Telecommunications Research
King's College London
26-29 Drury Lane, London, WC2B 5RL, UK
{assen.golaup, oliver.holland, hamid.aghvami}@kcl.ac.uk

*Abstract*—**In this paper, some issues in scheduling multimedia traffic over the HSDPA link of UMTS are addressed. Unlike previous contributions, the focus is on the case where packets having different QoS attributes need to be delivered to a specific user as part of a multimedia application packet delivery process. The following packet scheduling algorithms were considered: Max C/I, Proportional Fair (PF) and a new algorithm called Largest Average Weighted Delay First (L-AWDF). These algorithms were modified to provide *static* or *dynamic* priority to traffic streams belonging to the same user. It was found that the L-AWDF *static* and L-AWDF *dynamic* algorithms were more successful in reducing the percentage of packets dropped by the transaction and streaming components of the multimedia application than PF *static* and Max-CI *static*. L-AWDF *dynamic* gave lower average delay for the block traffic component than L-AWDF *static.* However, L-AWDF *static* gave the best performance for the transaction traffic component.**

*Index Terms*—**HSDPA, Packet Scheduling, Multimedia, QoS**

## I. Introduction

Third Generation (3G) mobile communication systems have the ability to support high bit rate services. This has been made possible by the introduction of higher throughput transport channels in new standards like Enhanced Data Rate for Global Evolution (EDGE) and Universal Mobile Telecommunications System (UMTS). For example EDGE supports a peak data rate of 237 Kbps on the downlink using four time slots [1]. The WCDMA interface of UMTS can do even better by providing a peak data rate of 2 Mbps [1]. Such capability enables the provision of multimedia services to mobile users which results in an enhanced user experience. In release 5 of UMTS, the High Speed Downlink Packet Access (HSDPA) feature has been introduced to further enhance the achievable data throughput [2].

HSDPA uses adaptive modulation and coding (AMC) to adapt to the channel conditions, instead of using a variable spreading factor and fast power control which are typical of CDMA (Code Division Multiple Access) systems. Moreover, HSDPA uses fast physical layer retransmission to help achieve a high throughput. The use of a small Transmission Time Interval (TTI) of 2ms is vital in facilitating the fast physical layer retransmission process due to a small round trip time [3]. When these features are used, a peak data rate of up to 14.4 Mbps can be achieved on the High Speed Downlink Shared Channel (HS-DSCH) [4]. Moreover, for a fixed amount of power, a dynamic range of 20 dB for the throughput can be achieved [5]. This range can be extended through the use of multicodes.

The ability to support peak data rates of up to 14.4 Mbps will enable application developers to create even more content rich and complex multimedia applications. Applications that are truly 'multimedia' in nature will emerge over wireless networks. Such applications will typically consist of different media types that will all need to be downloaded to the same user. In this paper, the focus is on the support of such applications over the HSDPA link. In Section II, issues that need to be addressed when scheduling multimedia traffic are discussed. In Section III, a simulation study is described to evaluate the performance of the following packet scheduling algorithms in supporting packets of different classes destined to the same user: Max C/I with *static* priority (Max-CI *static*), Proportional Fair with static priority (PF *static*) and Largest Average Weighted Delay First ( L-AWDF) with both *static* and *dynamic* priority. In Section IV, numerical results are provided. Finally, conclusions are provided in Section V.

## II. Issues in Scheduling Multimedia Traffic

A typical multimedia application consists of a number of media components that are integrated into a multimedia presentation. A user downloading such a presentation will have packets from each component queued at the Node-B buffer. These packets are likely to have different QoS attributes, even though they belong to the same user. In this case we have to consider two levels of priority: priority among users and priority among packets of different media components belonging to the same user. For example, the Max C/I or Proportional Fair algorithm deals with the issue of user priority. However, it will be still be necessary to prioritise among the packets of the different media components. Priority will depend on the QoS attributes of these media components. This priority can be *static* or *dynamic*. Static priority is defined as priority that is predetermined based on the QoS attributes. Dynamic priority, on the other hand, is based on the urgency with which packets of each component need to be delivered to the user. This urgency can be based on a certain deadline that must be met by packets of different components. Dynamic priority is more complicated as the scheduling algorithm has to compute the priority for each traffic component every time a slot is allocated to the user. The Modified Largest Weighted Delay First (M-LWDF) algorithm computes the priority among users by considering the delay of the head-of-line packets in addition to the relative channel quality [6]. However, this algorithm assumes that all packets destined to a given user have the same QoS criteria for a given user. It only allows QoS differentiation among different users having different application demands in terms of error rate. In our scenario, a scheduling algorithm is required that does QoS differentiation among packets belonging to the same user, in addition to QoS differentiation among users. For instance, the user priority can be a function of the delay of head-of-line packets of more than one component that make up the multimedia presentation. In addition to determining the user priority, the algorithm still has to work out the packet priority among the different media components. Packet priority is important as the algorithm might need to schedule packets of different components in the same slot in order to make the best use of the available throughput. In the following section, a simulation study is outlined that investigates the performance of several scheduling algorithms, modified to support multimedia traffic.

## III. SIMULATION STUDY

In order to investigate the issue of multimedia scheduling, a multimedia traffic model was required to generate the traffic for each user. This model is outlined in Section *A*. The packet scheduling algorithms used in this study are outlined in Section B followed by the simulation methodology adopted in Section C. In Section D, the performance measures used in this evaluation study are described.

### *A. Multimedia Traffic Model*

The model adopted here was based on the generalized multimedia traffic model presented in [7]. In [7], a multimedia application was described as consisting of up to three media components (block, transaction and streaming) as shown in Fig. 1 [7]. Each component has its own QoS requirements.

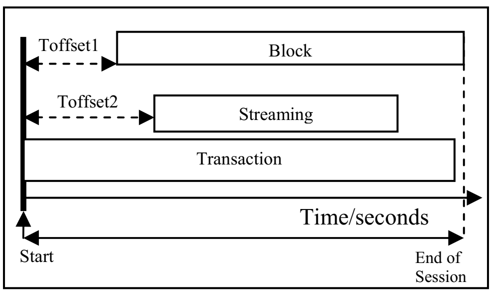


Fig. 1: Description of a multimedia session.

In this investigation, an *online shopping* application was modeled. The parameters of the online shopping session are detailed in Table II in the Appendix. The main QoS parameter used for each component was a discard timer for packets of each component. The discard timers used for the block, transaction and streaming components were respectively 15000, 250 and 1000 TTIs.

### *B. Packet Scheduling Algorithms*

The following packet scheduling algorithms were investigated:

- The Max C/I algorithm [6] and Proportional Fair (PF) [6] algorithms with static packet priority. The *static* priority mechanism was implemented as follows: The highest priority was given to transaction type packets followed by streaming and block type packets.
- The Largest Average Weighted Delay First (L-AWDF) algorithm is a modification of the M-LWDF algorithm. The user priority for the L-AWDF algorithm was defined as follows:

$$P_i = \frac{R_i(t)}{\lambda_i(t)} \sum_{j=1}^{J} w_j \frac{D_{i,j}(t)}{T_{i,j}}, \tag{1}$$

where $R_i(t)$ is the achievable user throughput of user $i$ at time $t$, $\lambda_i(t)$ is the average user throughput at time $t$, $D_{i,j}(t)$ indicates the delay of the head-of-line packet of the $j^{th}$ component of user $i$ and $T_{i,j}$ is the discard timer of the $j^{th}$ component of user $i$. The weighting factor $w_j$ of the $j^{th}$ component determines the relative importance of the $j^{th}$ component in computing the user priority. This algorithm was considered with both *static* (L-AWDF *static*) and *dynamic* priority (L-AWDF *dynamic*). Static priority was determined in the same manner as for the Max C/I and PF algorithms. Dynamic priority for the L-AWDF algorithm was determined according to the ratio $D_{i,j}(t)/T_{i,j}$ of the $j^{th}$ component i.e. the packets having the greatest value of $D_{i,j}(t)/T_{i,j}$ were given transmission priority in the next slot. Thus, the packet priority of a component could dynamically change every time the user was assigned a slot to transmit. For this study the weights for transaction, streaming and block traffic were set to 0.8, 0.15 and 0.05 respectively. These weights can naturally be varied to give higher priority to users having particular traffic types.

### *C. Simulation Methodology*

In this study, a quasi-dynamic network level simulator was used. One approach to simulate the performance of the various modulation and coding schemes of HSDPA is to use Actual Value Interface (AVI) tables [8]. These tables map the received symbol to noise energy ($E_s/N_o$) to an equivalent Block Error

Rate (BLER). Five different modulation and coding schemes were assumed to be available for use over the HSDPA link: QPSK ¼, QPSK ½ , QPSK ¾, 16QAM ½ and 16 QAM ¾ .

AVI tables were built for each of these schemes using results from [9]. The parameters used to obtain these link performance results are given in Table I. The results in [9] map the BLER for different modulation and coding schemes to the received own cell/other cell interference ratio ($I_{or}/I_{oc}$) (also known as the $G$ factor [10]). It is straightforward to map the received $I_{or}/I_{oc}$ into an equivalent $E_s/N_o$ [3, 11].

TABLE I
PARAMETERS USED TO OBTAIN LINK PERFORMANCE RESULTS [9]

| PARAMETER | SETTING |
|---|---|
| TTI | 3 slots (2 ms) |
| HSDPA Power | 80% of Node B power |
| Channel estimation | Perfect timing. Amplitude and phase estimated from Common Pilot Channel (CPICH) |
| Channel | ITU Pedestrian A channel |
| Carrier frequency | 2 GHz |
| Chip rate | 3.84 Mcps |

In order to model the network characteristics in terms of shadowing, propagation loss, antenna characteristics and cell size, the cumulative distribution function (CDF) of $I_{or}/I_{oc}$ (microcell case) from [10] was used. More details can be obtained from [12]. The $E_s/N_o$ variation per TTI, caused by the fast fading phenomenon, was modelled using a Rayleigh fading model [13]. Another important aspect of the HSDPA link that needed to be modelled was the hybrid ARQ retransmission strategy. The empirical approach proposed in [11] was adopted. Terminals were assumed to support up to 15 multicodes. Perfect estimation of the $E_s/N_o$ per TTI was assumed. Users were assumed to arrive in the cell according to a Poisson process. The multimedia users were assumed to be involved in *on-line shopping* sessions that generated traffic of the block type, transaction type and streaming type.

Each user having access to the HS-DSCH must have an Associated Dedicated Channel (ADCH), mostly for signaling purposes. The maximum number of ADCHs that can be supported at present is limited to 32 [4]. In this study, however, no call admission control was considered. The main objective was to evaluate the packet scheduling algorithms under heavy traffic loading conditions. Moreover, even though a maximum of four users can be code multiplexed in the same slot, it was assumed that only one user had access to the HS-DSCH in a slot.

### D. Performance Measures

The following performance measures were defined to characterize the performance of the algorithms over the HSDPA link when multimedia traffic is supported:

- *Average Delay*: The average delay was defined as follows:

$$D_{avg} = \frac{\sum_{j=1}^{J} D_{avg,j}}{J}, \tag{2}$$

where $D_{avg,j}$ is the average time to download the $j^{th}$ data unit and $J$ is the total number of data units downloaded during the simulation.

- *Maximum delay per data unit*: The maximum delay was taken to be the difference between the time at which the data unit arrived and the time at which the last packet belonging to that data unit was delivered.

- *Percentage of Dropped packets*: Packets were dropped when their absolute delay exceeded the specified maximum delay (QoS criterion) for each traffic type. The percentage of dropped packets was defined as:

$$P_{drop} = \frac{N_{drop}}{N_{del} + N_{drop}} \times 100, \tag{3}$$

where $N_{drop}$ is the number of packets dropped due to their delay exceeding a certain deadline for that class and $N_{del}$ is total number of packets delivered.

## IV. NUMERICAL RESULTS

Fig. 2 shows the variation of the average delay of the block traffic component and percentage of dropped block traffic packets against the average number of multimedia users in the system. The Max CI *static* algorithm gave the lowest average delay and lowest percentage of dropped block packets compared to the other algorithms. The L-AWDF *static* algorithm gave slightly higher average delay that the L-AWDF *dynamic* algorithm. However, the latter algorithm gave slightly better performance in terms of the percentage of packets dropped than L-AWDF *static*. The PF *static* algorithm performed quite poorly in avoiding block packets to be dropped.

Fig. 3 illustrates the results obtained for the transaction traffic component. The L-AWDF *static* algorithm gave lower average delay than the L-AWDF *dynamic* algorithm. This was expected as L-AWDF *static* gives absolute priority to transaction traffic packets, whereas L-AWDF *dynamic* computes this priority on a slot basis. The bias of L-AWDF *static* towards transaction traffic packets was more apparent when observing the graph of percentage of transaction packets dropped against average number of users in Fig. 3. L-AWDF *static* gave the lowest percentage of dropped packets followed by the L-AWDF *dynamic*. The PF *static* algorithm gave the highest percentage of dropped packets.

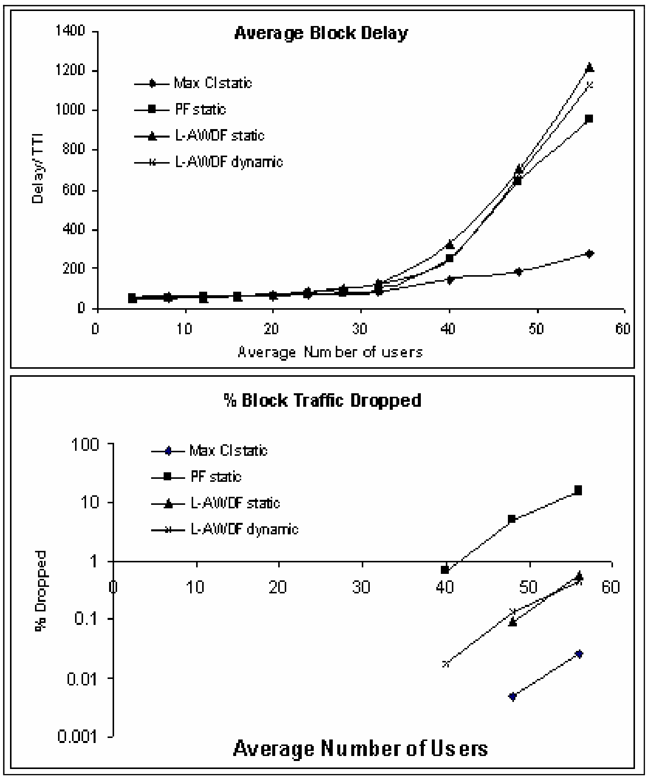


Fig. 2: Average block delay and percentage of block traffic packets dropped against the average number of users in system

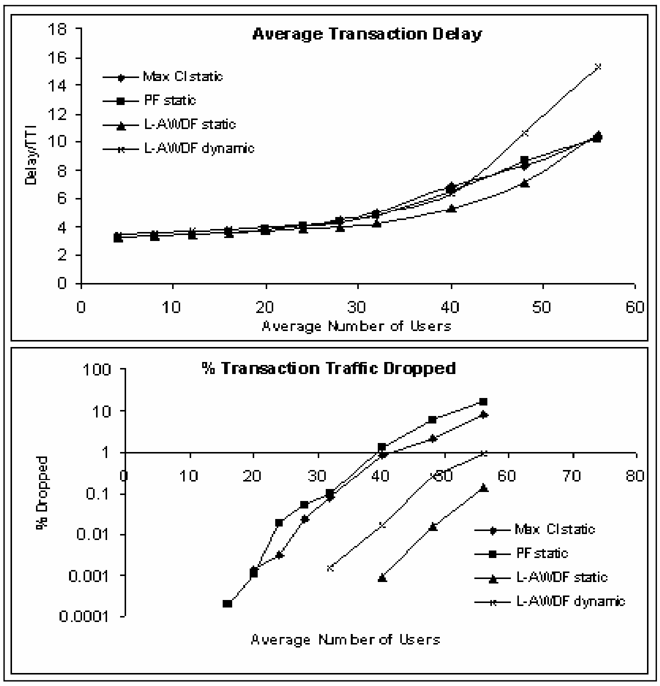


Fig. 3: Average transaction traffic delay and percentage of transaction traffic dropped against average number of users in system.

In Fig. 4 the results for the streaming traffic component are illustrated. Both L-AWDF *static* and L-AWDF *dynamic* gave higher average delay than PF *static* and Max-CI *static*. However, in terms of percentage of packets dropped, L-AWDF *static* and L-AWDF *dynamic* again did better that PF *static* and Max-CI *static*. It appears that the L-AWDF *static* and L-AWDF *dynamic* algorithms reduce the percentage of transaction and streaming packets dropped when compared to the PF *static* and Max-CI static *algorithm*. However, this came at the expense of a higher percentage of dropped block traffic packets when compared to Max-CI *static*. Even so, The L-AWDF algorithms (*static* and *dynamic*) performed much better than the PF *static* algorithm. The L-AWDF *dynamic* algorithm reduces the average delay of the block traffic component when compared to the L-AWDF *static* algorithm. However, this came at the expense of a higher average delay for the transaction and streaming components, as expected. In L-AWDF *static*, transaction traffic has absolute priority over streaming traffic which itself has absolute priority over block traffic. Thus the weights $w_j$ in Eq. (1) can be adjusted to favor the block traffic component to a greater extent if necessary.

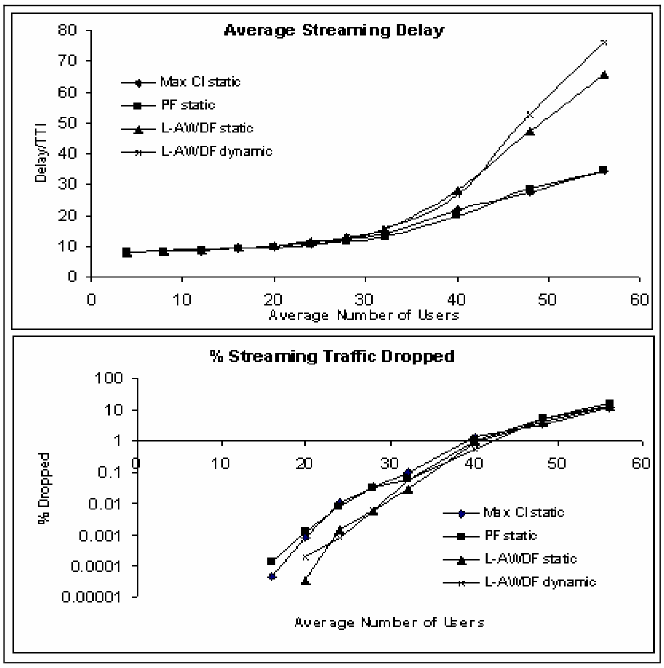


Fig. 4: Average streaming traffic delay and percentage of streaming traffic dropped against the average number of users in the system.

## V. Conclusions and Scope for Future Work

In this paper, the issue of scheduling multimedia traffic to mobile users over the HSDPA link was addressed. More specifically, the case where several traffic streams (with different QoS criteria) had to be delivered to the same user was considered. An *online shopping* application made up of three components (block, transaction and streaming traffic) was used as an example. It was found that the L-AWDF *static* and L-AWDF *dynamic* algorithms were more successful at reducing the percentage of packets dropped by the transaction and streaming components than PF *static* and Max-CI *static*. However, L-AWDF (*static* and *dynamic*) resulted in a higher average delay for the streaming and block traffic components. Moreover, the L-AWDF algorithms (both *static* and *dynamic*) had a higher percentage of block traffic packets dropped when compared to Max-CI *static*. The main advantage of L-AWDF *dynamic* over L-AWDF *static* was that it reduces the average delay of block traffic packets. However, L-AWDF *dynamic* started dropping packets at a lower traffic loading than L-AWDF *static*.

Future work will consider the issue where several users can be code multiplexed in the same slot and the scheduler has to deliver several streams of traffic (having different QoS criteria) to each one of the users.

## Acknowledgements

The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) for funding this work under grant: GR/R90529/01.

## References

[1] T. Halonen, J. Romero and J. Melero, *GSM, GPRS and EDGE Performance (Evolution Towards 3G/UMTS)*, Second Edition, Wiley 2003

[2] 3GPP TS25.855, High Speed Downlink Packet Access: Overall UTRAN Description, Version 5.0.0

[3] M. Dottling, J. Michel and B. Raaf, Hybrid ARQ and Adaptive Modulationand Coding Schemes for High Speed Downlink PacketAccess," *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2002),* Lisbon, Portugal, 2002

[4] H. Holma and A. Toskala, *WCDMA for UMTS*, Wiley, 2002

[5] T.E. Kolding, K.I. Pedersen, J. Wigard, F. Frederiksen and P.E. Mogensen, High Speed Downlink Packet Access: WCDMA Evolution, *IEEE Vehicular Technology Society News*, 2003

[6] P.J.A. Gutierrez, *Packet Scheduling and QoS in HSDPA*, PhD thesis, Aalborg University, Denmark, 2003

[7] A.Golaup and A.H. Aghvami, "A Framework for Modelling Emerging Multimedia Applications," *Consumer Communications and Network Conference (CCNC'04)*, Las-Vegas, USA, 2004

[8] S. Hamalainen, P. Slanina, M. Hartman, A. Lappetelainen, H. Holma and O. Salanaho, "A Novel Interface between Link and System Level Simulations," *ACTS Summit 1997*, Aalborg, Denmark, 1997

[9] T.E. Kolding, F. Frederiksen, P.E. Mogensen, "Performance Evaluation of Modulation and Coding Schemes Proposed for HSDPA in 3.5G UMTS Networks," *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Aalborg, Denmark, 2001

[10] J.R. Fonollosa, M. Heikkila, X. Mestre, A. Pages, A. Pollard, L. Schumacher, L.T. Beger, A. Wiesel and J. Ylitalo, "Adaptive Modulation Schemes for MIMO HSDPA," *IST Mobile & Wireless Telecommunications Summit 2002*, Thessaloniki, Greece, 2002

[11] F. Frederiksen and T.E. Kolding, "Performance and Modelling of WCDMA/HSDPA Transmission/H-ARQ Schemes," *IEEE 56th Vehicular Technology Conference (VTC)*, 2002

[12] E. Tiirola et al., METRA Deliverable D3.2, "Review and Selection of Relevant Algorithms." Available at: www.ist-metra.org

[13] W.C. Jakes, *Microwave Mobile Communications*, IEEE Press, 1974

## Appendix

TABLE II
PARAMETERS FOR 'ON-LINE SHOPPING' SESSION

| | |
|---|---|
| **BLOCK TRAFFIC PARAMETERS** | |
| Distribution of number of Data Units | Geometric |
| Mean number of data units per session | 3 |
| Maximum blocks per session | 5 |
| *Parameters for Block size* | |
| Distribution | Lognormal |
| Mean block size in bytes | 37500 |
| Variance of block size (bytes) | 4e10 |
| *Parameters for inter-arrival time* | |
| Distribution | Lognormal |
| Mean interarrival time of blocks | 39.45 |
| Variance of interarrival time of blocks | 8596 |
| **TRANSACTION TRAFFIC PARAMETERS** | |
| *Session time parameters* | |
| Distribution | Exponential |
| mean session time of packet train session | 300 |
| *ON time parameters* | |
| Distribution | Exponential |
| Mean ON time | 8 |
| *Inter-arrival time during ON period Parameters* | |
| Distribution | Exponential |
| Mean inter-arrival time during ON period | 0.5 |
| *OFF time parameters* | |
| Distribution | Lognormal |
| Mean OFF time | 39.45 |
| Variance | 3.4e10 |
| *Packet size parameters* | |
| Distribution | Lognormal |
| Mean packet size in bytes | 300 |
| Variance of packet size (bytes) | 40000 |
| **STREAMING (MPEG4) TRAFFIC PARAMETERS** | |
| Mean GoP size/bytes | 6616 |
| Variance of GoP size | 9.03e5 |
| Minimum GoP size | 5511 |
| Maximum GoP size | 14111 |
| Frame rate/seconds | 25 |
| *Session time parameters of streaming traffic* | |
| Distribution | Exponential |
| Mean session time of streaming traffic | 240 |
| **MULTIMEDIA TRAFFIC PARAMETERS** | |
| Mean multimedia traffic session time in secs | 300 |
| Mean offset time for block traffic in secs | 10 |
| Mean offset time for packet train traffic in secs | 0 |
| Mean offset time for streaming traffic in secs | 30 |