MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Dynamic Resource Control for High-Speed Downlink Packet Access Wireless Channel

Huai-Rong Shao, Chia Shen, Daqing Gu, Jinyun Zhang, Philip Orlik

TR2003-60 May 2003

Abstract

It is a challenging task to provide Quality of Service (QoS) control for a shared high-speed downlink packet access (HSDPA) wireless channel. In this paper, we first propose a new dynamic resource control framework integrated with adaptive modulation and coding (AMC) and hybrid automatic repeat request (H-ARQ) to support class-based multimedia applications over HSDPA wireless channels. Then we present a new scheduling algorithm, Delay-sensitive Dynamic Fair Queueing (DSDFQ), to meet delay requirements of multimedia applications as well as maintain high network efficiency. The proposed approach can easily adapt to load fluctuations of different traffic classes and varying wireless channel conditions caused by user mobility, fading and shadowing. Performance evaluation shows the advantage of our proposed approach.

23rd International Conference on Distributed Computing Systems, International Workshop on Mobile and Wireless Networks

Dynamic Resource Control for High-Speed Downlink Packet Access Wireless Channel

Huai-Rong Shao, Chia Shen, Daqing Gu, Jinyun Zhang, Philip Orlik
Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA, 02139
{shao, shen, dgu, zhang, porlik}@merl.com

ABSTRACT

It is a challenging task to provide Quality of Service (QoS) control for a shared high-speed downlink packet access (HSDPA) wireless channel. In this paper, we first propose a new dynamic resource control framework integrated with adaptive modulation and coding (AMC) and hybrid automatic repeat request (H-ARQ) to support class-based multimedia applications over HSDPA wireless channels. Then we present a new scheduling algorithm, Delay-sensitive Dynamic Fair Queueing (DSDFQ), to meet delay requirements of multimedia applications as well as maintain high network efficiency. The proposed approach can easily adapt to load fluctuations of different traffic classes and varying wireless channel conditions caused by user mobility, fading and shadowing. Performance evaluation shows the advantage of our proposed approach.

1. INTRODUCTION

Future wireless cellular networks will support integrated multimedia applications with various quality of service (QoS) requirements. Providing QoS differentiation in high-speed wireless data networks is considered as a promising solution for the increasing multimedia demands from wireless end users. Due to the difference between traffic characteristics of packet data and traditional circuit-switched voice, dedicated channels are allocated for data services in many wireless communication systems and standards such as High Data Rate (HDR) systems (a.k.a. 1xEV-DO) [1], 1xTREME (a.k.a. 1xEV-DV) of 3rd Generation Partnership Project 2 (3GPP2) [2], and High Speed Downlink Packet Access (HSDPA) of 3rd Generation Partnership Project (3GPP) [3]. In HSDPA, a high-speed downlink data channel is shared by multiple users within the same cell. Many technologies are currently under consideration for HSDPA system to enhance the system performance. They include adaptive modulation and coding (AMC), hybrid automatic repeat request (H-ARQ), fast cell selection (FCS) and multiple-input-multiple-output (MIMO) technologies.

AMC offers a link adaptation method that can dynamically adapt modulation-coding scheme to current channel conditions for each user (known as UE (user equipment)). In a system with AMC, users close to the base station (BTS) (known as node B in HSDPA) usually have good radio link and are typically assigned higher order modulations and higher code rates (e.g. 64 QAM with R=3/4 turbo codes). The modulation-order and/or code rate will decrease as the distance of a user from BTS increases. H-ARQ provides a retransmission mechanism for the lost or erroneous information. There are many schemes for implementing H-ARQ - Chase combining, Rate compatible Punctured Turbo codes and Incremental Redundancy.

An important research issue is how to perform resource control and management and integrate radio resource control schemes with these new technologies. As a simple instance, data transmission capacity at a base station will vary accordingly with AMC schemes used by users. Given the same amount of spreading code (code space) and time slot (time space) resources, an mobile user with higher order modulation scheme and higher coding rate usually can obtain higher data rate than those with lower order modulation schemes and lower coding rate. This paper focuses on the resource allocation with QoS control in HSDPA system with AMC and H-ARQ mechanisms.

Packet scheduling is one of the most important QoS control approaches for wireless multimedia communications. Many proposals on packet scheduling in wireless environments have been proposed, for example, CSDPS (Channel state dependent packet scheduling)[5], IWFQ (Idealized Wireless Fair Queueing Algorithm) [6], CIF-Q (Channel-Condition Independent Fair Queueing) [7], SBFA (Server Based Fairness Algorithm) [8], I-CSDPS (Improved channel state dependent packet scheduling) [9], CAFQ (Channel adaptive fair queueing) [10], M-LWDF (Modified Largest Weighted Delay First) [11], and CDGPS (Code-division generalized processor sharing) [12]. Most approaches [5-9] assume a simple two-state Markov model: the scheduler simulates an error-free system running a wire-line packet scheduling algorithm when the sessions have good (or perfect) channel states (effective throughput is maximum). When the session that is scheduled to transmit data encounters a bad channel state, it will give the transmission opportunity to other error-free sessions (i.e., with a good channel state). Then these error-free sessions will give their transmission rights back to the error session once the error session escapes from a bad channel state. These algorithms mainly aim to provide fairness and also soft QoS guarantees. In [10], a new definition of fairness and a scheduling algorithm adapting to several channel conditions is proposed. However, it doesn't provide explicit QoS guarantee. In [11], a user scheduling scheme based on the tradeoff between delay and throughput is presented. This approach assumes that each user can only support one QoS traffic class at a time. GPS (Generalized processor sharing) scheme is dynamically applied to spreading codes rather than time slots for different mobile users in [12].

In this paper, we propose a resource scheduling mechanism that is closely integrated with some HSDPA technologies such as AMC and H-ARQ. Because AMC scheme for a user is changed dynamically according to its channel condition, the scheduler can obtain the channel condition information of each user through its AMC scheme. Therefore, the proposed scheduling mechanism is not based on the simple "On-Off" two state Markov model. H-ARQ introduces extra traffic load into wireless networks. Previous work didn't consider this seriously. In the proposed scheme, we differentiate the original packets and the re-transmitted packets by placing them at different queues. In the future, one mobile device should be able to support multiple streams with different QoS requirements simultaneously. For instance, one user can get streaming video from a video server and get text file from a FTP server at the same time. Thus the scheduler at base station needs to handle both QoS traffic classes and different mobile users sharing the same HSDPA downlink channel. Previous work usually considers how to schedule resources to different mobile users and assign each user a separate queue. In the proposed scheme, queues are assigned to different QoS classes rather than different users. Queue parameters such as queue length are specified according to delay and packet loss requirements. We propose DSDFQ (Delay-sensitive Dynamic Fair Queueing) scheduling scheme that dynamically adjust the queue weights with traffic dynamics and queue status. Therefore, explicit QoS can be provided with the proposed scheme. Each class is then sub-divided into sub-classes according to each user's AMC scheme. It is that, in each class, mobile users with the same AMC are grouped into the same sub-class. Users with better channel condition (indicated by AMC schemes) are given higher scheduling priority than those with worse channel condition. It is achieved by setting different RED (random early Detection) for different AMC schemes. Because of this kind of sub-class mechanism, a good system throughput can be maintained by constraining the mobile users with bad channel condition.

The rest of the paper is organized as follows. Section 2 presents our new dynamic resource control framework. Section 3 describes the proposed packet scheduling algorithm and then gives a theoretical analysis. Section 4 shows the simulation results. Finally, section 5 concludes the paper.

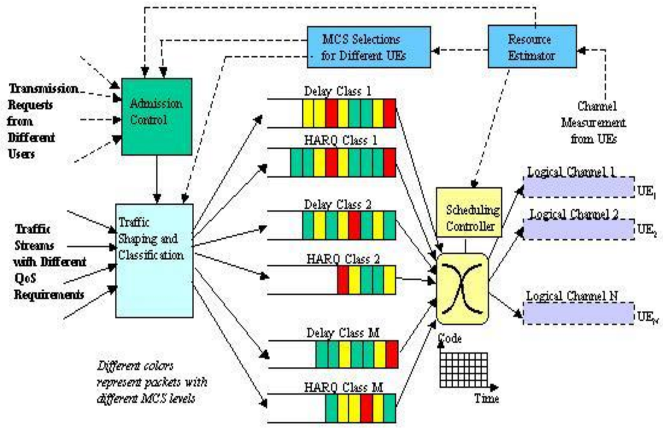

Fig.1: Resource control framework for HSDPA

2. RESOURCE CONTROL FRAMEWORK

Fig. 1 illustrates the proposed dynamic resource control framework for HSDPA.

We mainly consider how to perform resource control at node B in HSDPA systems. It is assumed that multiple mobile or fixed UEs exist in one cell, and each UE can support multiple streams or traffic classes. Four traffic classes are specified in 3GPP or HSDPA: Conventional class – voice traffic; Streaming class – audio and video traffic; Interactive class – web browsing, database read types of traffic; and Background class – best effort traffic. Compared with other's work, our approach not only considers how to schedule packets to different UEs, but also schedule different traffic classes in a multiple UE environment.

Within the framework, transmission requests are sent to the Admission Control module. The Admission Control module makes the decision whether or not to admit new transmission streams by computing the available resource from physical-layer resource measurement and existing traffics. Once admitted, traffic streams enter the Traffic Shaping and Classification module, and then are passed to different queues according to traffic classes to which traffic streams belong. Traffic streams are classified according to delay and packet loss requirements. These QoS parameters decide the queue length, the weight of a queue and RED (Random Early Detection) configuration. Because H-ARQ retransmission may introduce large amount of extra traffic load when wireless channels are in bad condition, we assign two queues to each traffic class: original transmission queue and re-transmission queue. For each class, sub-classes (different colors) are specified according to MCS (Modulation and coding scheme). Both spreading codes and time frames are scheduled to UEs.

3GPP HSDPA specifies that time frame length (or TTI: Transmission Time Interval) is fixed and equal to 2ms, and spreading codes are orthogonal codes. SF (Spreading Factor) is fixed and equal to 16, so that the total spreading code number is 16. We assume that one UE can be assigned with multiple spreading codes in the same TTI. WFQ weights are dynamically adjusted according to the queue status of different classes. Detailed scheduling will be presented in Section 3.

As illustrated in Fig. 1, each UE monitors the channel status and feedback carrier-to-interference (C/I) information to node B periodically. Then node B estimates the available resource and decides modulation and coding scheme for data transmission to the UE. We assume there are N MCS levels available, and let $R_n(1 \le n \le N)$ be data rate with MCS level n, $FER(\gamma, n)$ denote frame error rate (FER) for a given signal-to-noise ratio of channel γ and MCS level n. Then, the effective data transmission rate is equal to $R_n \times (1 - FER(\gamma, n))$. A UE selects MCS level so as to maximize the effective data rate for the measured γ. That is, $j = \max_n \{R_n \times (1 - FER(\gamma, n))\}$ where j is the selected MCS level.

The protocol data unit (PDU) at a radio interface is the data traffic carried during a time frame (or TTI). Since data rate varies with MCS level, PDU size also varies. An IP packet arrived at node B is segmented according to the allowable PDU sizes. A UE acknowledges the reception of a PDU. If node B doesn't receive an acknowledgement within a time period, it retransmits the PDU. If the maximum number of retransmission trials for a PDU is M, a PDU can be transmitted up to M+1 times including the original transmission. During a retransmission sequence, MCS level is kept the same as that for the original transmission. To improve link utilization and adapt to our scheduling mechanism, we propose different maximum number of retransmission (M) for PDUs with different MCS levels. The higher MCS level is, the larger M value is given.

3. DSDFQ Scheduling

We propose a new scheduling scheme called Delay-Sensitive Dynamic Fair Queueing (DSDFQ) for HSDPA systems. The scheduler is located at MAC layer to schedule PDUs carried at different TTIs and with different spreading codes. For convenience, we still call these kinds of PDUs as packets. From Section 2, it can be seen the proposed scheduler is closely integrated with AMC and H-ARQ. The key point of the scheduling scheme is dynamic adjustment of the weight of each queue according to the current delay. We use colored tokens to distinguish packets with different MCS levels within one queue. Moreover, we use Weighted RED as the packet drop scheme.

3.1. Scheduling Algorithm

The purpose of Delay-sensitive Dynamic Fair Queueing is to maintain a dynamic fairness according to the delay status of each queue. Unlike most fair queueing schemes, this scheme is not an approximate approach to GPS (Generalized Processor Sharing), though some ideas are from the common Fair Queueing schemes.

In [13], a sorted priority queue mechanism that is commonly used by Virtual Clock, WFQ and WF2Q was discussed. The framework of Delay-sensitive Dynamic Fair Queueing is also based on this mechanism. In this mechanism, there is a state variable associated with each connection to monitor and enforce its traffic. Here, we use the same state variable as WFQ that is called the Virtual Finish Time.

The virtual finish time of a packet is defined as:

$$F_i^k = \max\{F_i^{k-1}, V(t)\} + \frac{L_i^k}{f_i},$$

where F_i^k is the virtual finish time of the kth packet of class i, $V(t)$ is the virtual time when the kth packet arrives, f_i is the weight of class i and L_i^k is the packet size of the kth packet measured in byte.

In WFQ, weight f_i is fixed and does not reflect the current situation of the queue. In the proposed Delay-sensitive Dynamic Fair Queueing, the weight of each queue, f_i, is a variable. When an event occurs, the delay of this packet $(delay_i^k)$ is changed and f_i is recalculated as follows:

$$f_i = f(delay_i^k) = \min(f_i^0 + delay_i^k \times k_i, f_i^{\max})$$

where f_i^0 is the basic weight of class i, $f_i^{\max}$ is the maximum weight of class i, $delay_i^k$ is queueing delay of the kth packet in class i, and k_i is the adjustment parameter. Let S_i^k denote the virtual time when packet k in class i starts to be served, and F_i^k denote the virtual time when packet k in class i finishes serving.

When a packet is entering a queue, the processing steps are:

1) Get the maximal delay of the queue,

2) Update virtual time, i.e., get S_i^k, then F_i^k, and determine the dequeue order.

Mathematically, Steps 1) and 2) can be respectively represented as

$$delay_i^k = delay_i^{k-1} + L_i^{k-1} / (f_i^{k-1} \times Bw) \times \sum_j f_j$$

and

$F_i^k = \max(F_i^{k-1}, V(t_{last}) + (t - t_{last}) / \sum_i f_i) + L_i^k / (f_i^k \times Bw)$,

where L_i^{k-1} is the length of the kth packet in class i, Bw is the bandwidth of the output link, and t_{last} is the real time when the virtual time last updates (the weight's latest change).

According to the order of finish time, insert this packet to the outgoing queue.

3) Next, change the weight.

According to the new delay value, calculate the current weight as follows:

$$f_i^k = f(delay_i^k) = f_i^0 + g(delay_i^k),$$

where, f_i^0 is the basic weight of class i, and $g(delay_i^k)$ can be treated as a curve called "Weight Curve".

4) Determine the real time at which the next packet will depart the queue. Then schedule the next departure of the queue. (This can also be processed when dequeueing a packet if the processing speed is fast enough).

Denote the real time as Next(t). Then at time Next(t), the objective is to change the weight and delay of each class as follows.

$$delay_i = delay_i - L_i^k / (f_i^k \times Bw) \times \sum_j f_j ,$$

At the output link, the system simply forwards the packets. Thus, when a packet P_i^k enters the queue, its delay is first calculated, and then the weight f_i is increased if the delay is larger than before. So the guaranteed rate of class i is temporarily increased. Since the packets of each class arrive independently and the weight changes each time when a packet arrives, the dynamic balance is maintained and the bandwidth allocation could be much more fair than other static scheduling schemes. We call such fairness "Dynamic Fairness".

3.2. Theoretical Analysis

We analyze the delay bound of Delay-sensitive Dynamic Fair Queueing when $g_i(delay)$ is not zero. In the following analysis, we assume that the traffic has the constraint similar to those by leaky bucket shaping. Thus, the traffic entering the network is shaped as follows:

$$A_i(t,t) \le s_i + r_i(t - t), \forall 0 \le t \le t,$$

where $A_i(t,t)$ is the amount of traffic for class i that enters the network during time interval $[t,t]$. Tokens are generated at a fixed rate r_i, and packets can be released into the network only after acquiring the required number of tokens from the leaky bucket. The leaky bucket contains at most s_i bits tokens.

In our scheme, the calculation of delay of packet k in class i is a key problem. As mentioned earlier, we calculate the delay when a packet enqueues and update it when a packet dequeues the system. Thus the delay that we get would be smaller than the real delay by a value within (1,L/Bw) where L is the packet length and Bw is the bandwidth of the output link. We use another method to get the delay of a packet. Suppose that the system starts its busy period from time zero. Let

$$D_i^k(t) = \sum_{(j,l)\subset A} L_j^l / Bw$$

denote the total delay of packet k of class i from time zero. The real delay of packet k of class i can be represented as:

$$delay_i^k = D_i^k(t) - t = \sum_{(j,l)\subset A} L_j^l / Bw - t,$$

where t is the real time when packet k of class i arrives.

Further, we assume that the traffic is greedy and the service start time of a packet is just the time that the previous packet finishes its service. Thus, we have $S_i^k = F_i^{k-1}$, and

$$F_i^k = S_i^k + L_i^k / (f_i^k \times Bw) = F_i^{k-1} + L_i^k / (f_i^k \times Bw).$$

Then we can get:

$$D_i^k = D_i^{k-1} + \sum_{(j,l)\subset B(i,k)} L_j^l / Bw$$

and

$$B(i,k) = \{(j,l) : F_j^l - F_i^{k-1} < L_i^k / (f_i^k \times Bw)\}.$$

The number of elements in set $B(i,k)$ can be interpreted as $N_B^{i,k}$, which satisfies

$$N_B^{i,k} < \frac{L_i^k}{f_i^k \times Bw} \times \sum_{j\ne i} (r_j / L_{\min} \times f_j^{\max}) < \frac{L_{\max}}{L_{\min}} \times \sum_{j\ne i} (r_j \times f_j^{\max}) \times \frac{1}{f_i^k} / Bw,$$

where r_j is the average rate of class j, $L_{\min}$ is the minimum size of a packet, and $f_j^{\max}$ is the maximum weight of class j.

Now, we can see

$$D_i^k < D_i^{k-1} + \frac{L_{\max}}{Bw} \times \frac{L_{\max}}{L_{\min}} \times \sum_{j\ne i} (r_j \times f_j^{\max}) \times \frac{1}{f_i^k} / Bw$$

$$D_i^k < D_i^{k-1} + \frac{1}{f_i^k} \times \frac{L_{\max}^2}{Bw^2 \times L_{\min}} \times \sum_{j\ne i} (r_j \times f_j^{\max}) = D_i^{k-1} + Cs \times \frac{1}{f_i^k}.$$

Consequently, we can derive the delay as follows:

$$D_i^1 < Cs \times \frac{1}{f_i^0},$$

$$D_i^2 < D_i^1 + Cs \times \frac{1}{f(D_i^1)},$$

and

$$D_i^k < D_i^{k-1} + Cs \times \frac{1}{f(D_i^{k-1})}.$$

The delay of packet k in class i is: $delay_i^k = D_i^k - t_i^k$.

Suppose the buffer of class i is $q\lim_i$, then the packet will be dropped when the queue size $Qlen_i^k$ is equal to $q\lim_i$, where $Qlen_i^k$ can be restricted by the following:

$$Qlen_i^k = k - \sum_{l=1}^{k}(t_i^l - t_i^{l-1}) \times \frac{f_i^l}{\sum f_j} \times Bw$$

$$\le k - \sum_{l=1}^{k}(t_i^l - t_i^{l-1}) \times \frac{f_i^l}{\sum f_j^{\max}} \times Bw,$$

$$Qlen_i^k \le k - \sum_{l=1}^{k}(t_i^l - t_i^{l-1}) \times \frac{f(delay_i^k)}{\sum f_j^{\max}} \times Bw.$$

4. Simulation Evaluation

We simulated one cell case and didn't consider interference between cells in the simulation. At initialization, K UEs are uniformly distributed in the cell under observation, and their moving directions are randomly chosen. The speed of a UE is decided as uniformly distributed with the mean of 3 km/h and standard deviation of 1 km/h. Every 5 seconds, a UE changes its speed according to the same manner as at the initialization. At that time, the UE ether updates its moving direction with probability of 0.3 or does not change its moving direction with probability of 0.7. If it decides to change its moving direction, the direction is chosen as uniformly distributed within [-π/2, π/2]. We use the following signal path loss model that is defined in 3GPP as the channel model.

L = 128.1 + 37.6 log 10 D,

where D is the distance (in km) between node B and a UE. We consider the following correlation model for shadow fading.

$$C(d) = e^{\frac{d}{d_{cor}}\ln 2}$$

$$S = C(d) \cdot S^* + \sqrt{1 - [C(d)]^2} \cdot N(0, s).$$

where C(d) is a normalized autocorrelation function of shadowing, dcor is the de-correlation length, and d is the distance that a UE has moved since the last calculation of shadowing. dcor was set to 5 m in the simulation. The lognormal shadowing in logarithmic scale is characterized by a Gaussian distribution with zero mean and a standard deviation σ. The lognormal shadowing with zero mean and a standard deviation of 8 dB was considered. S is the shadowing value represented in dB and is updated using the last calculated shadowing value S*. In simulation, we update shadowing value every 5 sec.

All four traffic classes were inputted into the simulation environment: E-mail for best effort class, web browsing for interactive class, video for streaming class and PCM voice for conventional class. As for video, we simulated both CBR video and self-similar VBR video with Pareto model, and three different bit rates were considered: 56kbps, 384kbps and 1.5Mbps.

We compared our DSDFQ scheme with WFQ and FIFO (no QoS) schemes. Fig. 2 and Fig. 3 demonstrate the delay performance of those three schemes.. Firstly, our scheme can achieve much smaller delay than WFQ and FIFO for both VBR and CBR videos. In the case of CBR, our scheme can control delay in 20 ms, but video delay with WFQ is almost 20 times larger than our scheme. In the case of VBR, our scheme reduces the delay in half when compared with WFQ. Secondly, our scheme can also achieve good delay performance for other traffic classes when compared with WFQ and FIFO.

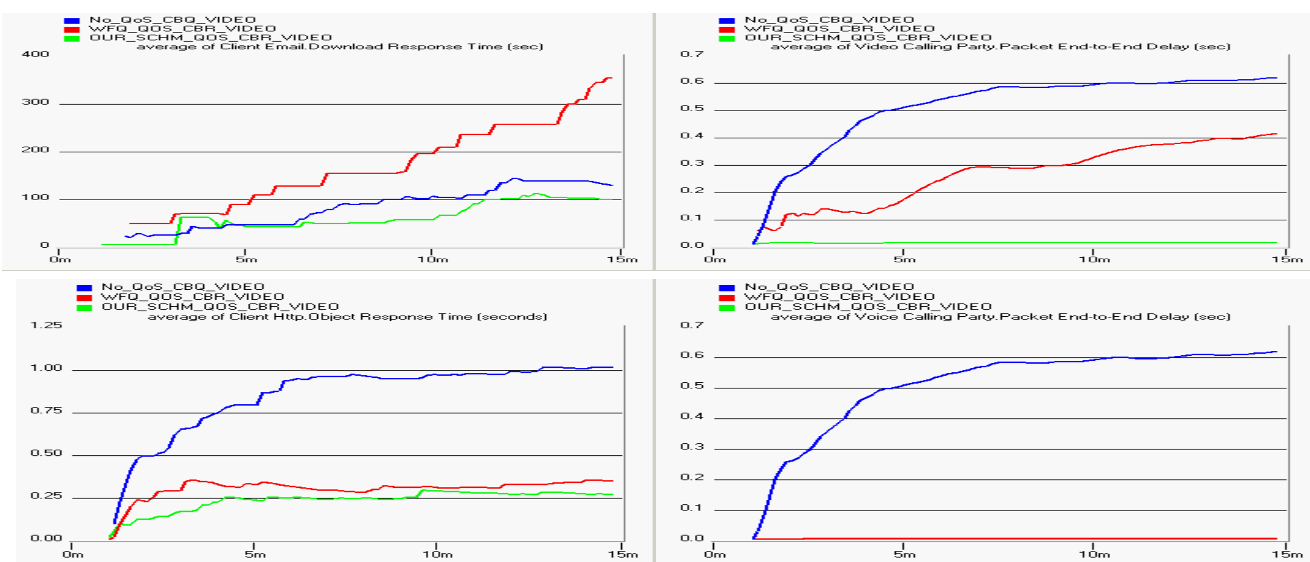

Fig.2: Delay Comparison for CBR video and other traffic classes

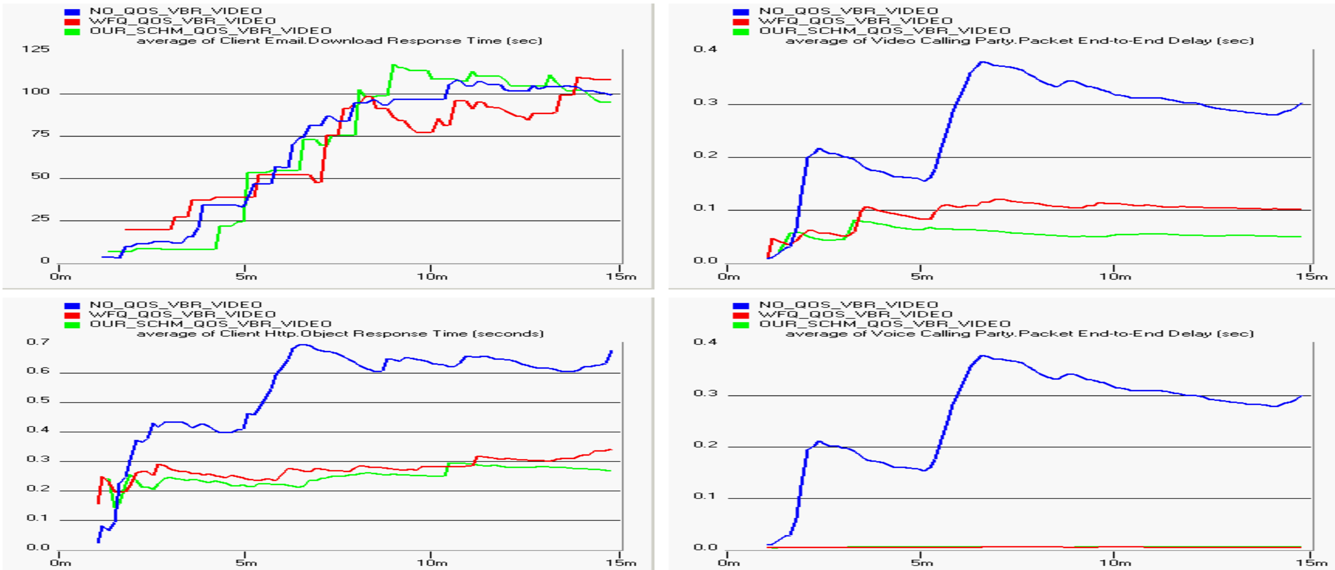

Fig.3: Delay Comparison for VBR video and other traffic classes

5.Conclusion and Future Work

This paper proposes a new dynamic resource control framework to support multiple traffic classes over the shared high-speed wireless packet downlink channel of next generation wireless network. In this framework, combined with AMC and H-ARQ, the proposed DSDFQ scheduler allocates both spreading codes and time slots to different UEs and traffic classes to meet the delay requirements from different multimedia applications. The proposed approach can also be easily adapted to load fluctuations from different traffic classes and dynamic wireless environment. The future work includes how to reduce signaling load between UEs and node B by decreasing spreading code switch times for each UE at the scheduler.

6. References

[1] P. Bender, et al. "CDMA/HDR: A bandwidth efficient high speed data service for nomadic users", IEEE Commun. Mag., Vol. 38, No. 7, pp. 70–77, July 2000.

[2] Motorola and Nokia, "3GPP2 1xTREME Presantation" C00-20000327-003, Mar. 2000.

[3] Motorola, "Feasibility study of advanced technique for High Speed Downlink Packet Access," TSG-R WG1 document, R1-556, April, 2000, Seoul, Korea.

[4]Y. Cao, V. Li, "Scheduling Algorithms in Broad-Band Wireless Networks", IEEE PROCEEDINGS OF THE IEEE, p. 76~87, VOL. 89, NO. 1, JANUARY 2001.

[5] C. Fragouli, V. Sivaraman, and M. Srivastava, "Controlled mul-timedia wireless link sharing via enhanced class-based queueing with channel-state dependent packet scheduling," in Proc. IN-FOCOM'98, vol. 2, Mar. 1998, pp. 572–580.

[6] S. Lu and V. Bharghavan, "Fair scheduling in wireless packet net-works," IEEE/ACM Trans. Networking, vol. 7, no. 4, pp. 473–489, 1999.

[7] T. S. Eugene Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," in Proc. INFOCOM98, Mar. 1998, pp. 1103–1111.

[8] P. Ramanathan and P. Agrawal, "Adapting packet fair queueing algorithms to wireless networks," in Proc. ACM MOBICOM'98, Oct. 1998.

[9] J. Gomez, A. T. Campbell, and H. Morikawa, "The Havana frame-work for supporting application and channel dependent QoS in wireless networks," in Proc. ICNP'99, Nov. 1999, pp. 235–244.

[10] Li Wang, Yu-Kwong Kwok, Wing-Cheong Lau, and Vincent K. N. Lau, "Channel Capacity Fair Queueing in Wireless Networks: Issues and A New Algorithm", ICC 2002, April, 2002, New York, U.S.A.

[11] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," IEEE Communications Magazine, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[12] L. Xu, X. Shen, J. Mark, "Dynamic bandwidth allocation with fair scheduling for WCDMA systems", IEEE Wireless Communications, April, 2002.

[13] Hui Zhang, "Service Discipline For Guaranteed Performance Service in Packet-Switching Networks", Proceedings of IEEE, 83(10), Oct. 1995.

[14] W. Jeon, D. Jeong, B. Kim, "Design of Packet Transmission Scheduler for High Speed Downlink Packet Access Systems", Proc. Of the IEEE VTC 2002, Spring.