Three Dimension QoS Deviation based Scheduling in Adaptive Wireless Networks

Ramtin Kazemi*, Mohammad Hossien Yaghmaee**, Jalil Chiti-zadeh*
*Electrical Engineering Department, Ferdowsi University of Mashad
**Computer Department, Ferdowsi University of Mashad

***Abstract*—QoS provisioning over wireless fading channel is challenging. We develop a novel scheduler design in cellular packet-switched wireless networks which provides QoS for users in all three aspects of QoS i.e. throughput, delay and packet loss simultaneously. We establish a three-dimensional space with specific basis vectors for QoS and find the efficient point of system performance in that space. Then we develop a generalized measure, the *QoS Deviation*, which is the Euclidean distance between flow QoS work point and the efficient QoS point in the 3D space. Based on this measure, a scheduling approach, namely BQDF is outlined and will be extended to AQDC scheduler for wireless channels which makes it possible to tune the tradeoff between QoS provisioning and optimizing throughput in an adaptive manner depending on current cell QoS Deviation level (CDL). Finally, we introduce a QoS Deviation-based CAC policy for the proposed system.**

***Index Terms*—three dimensional quality of service (3D-QoS), QoS Deviation, cell QoS Deviation level (CDL), call admission control (CAC), adaptive modulation and coding (AMC).**

I. Introduction

For many applications, it is important that certain QoS requirements be satisfied. At the same time, the demand for enabling such applications in a wireless mobile environment is growing rapidly. The third generation and beyond wireless communication systems will offer higher data rates and flexible packet-switched services with negotiable QoS at startup and are anticipated to provide a broad range of multimedia services including voice, data and video to mobile users each with different certain QoS requirements. Multimedia application can be classified in two categories: QoS-guaranteed and best-effort ones [1]. In the first category despite of the second one, providing some QoS requirements is desired [2], [3]. Notice that within this paper, we mean the QoS-guaranteed services, not best-effort ones unless something else is mentioned.

Actually, supporting QoS guarantee is accomplished by means of various components in different layers. QoS-aware MAC algorithms at MAC sub layer of data link layer, QoS-aware routing algorithms at network layer, packet queuing and scheduling algorithms at data link layer are some of these components. In this way, scheduling is one the most prominent ones. The task of a scheduler is to make decisions about how flows (user or session) are selected to be serviced to guarantee their desired QoS in a shared resource environment.

Generally, QoS has three aspects including throughput (rate), packet delay and packet loss. A scheduler may not take into account all these three aspects simultaneously and will consider only one or two. Schedulers like JoBS [4] and those which are proposed in this paper do scheduling based on all three aspects of QoS. However, many schedulers like WFQ, DRR and SFQ [5] don't consider user individual QoS parameters at all and deal with the fairness just as an equal chance to transmit data according to the required or reserved service portion.

There are a large number of traffic scheduling approaches available for wireline networks [6]. However, they cannot be directly applied to wireless networks because of the fundamental differences between the two. In wireline networks despite of the wireless ones, link capacity is constant and scheduler just tries to distribute it between users according to their QoS requirements, but supporting QoS guarantee in wireless networks is very challenging [7]. The major problem in wireless networks is channel. Multipath fading and Doppler Effect due to users' mobility cause channel characteristics to be changed with respect to time and location. Scheduling regardless of these changes will degrade utilization of resources markedly. One solution to this problem is dynamic distribution of resources according to the wireless channel. This link adoption can be done in APP layer, e.g. by Video-Codec as well as in lower layers [8].

At the physical layer using AMC, efficient bandwidth utilization can be achieved [9], [10]. The objective of the AMC is to maximize the throughput by adjusting transmission parameters, including modulation and FEC coding to channel variations, while maintaining a prescribed error rate. Based on channel estimations obtained at the receiver, the AMC selector determines the modulation-coding pair (mode), which is sent back to the transmitter through a feedback channel, for the AMC controller to update the transmission mode. Therefore, depending on the selected mode in AMC, transmission with different rates will be possible. In this way, channel model will be the finite state Markov chain (FSMC) in which the number of states is equal to the number of AMC selectable transmission modes. Figure 1 illustrates this system and the relation between its components.

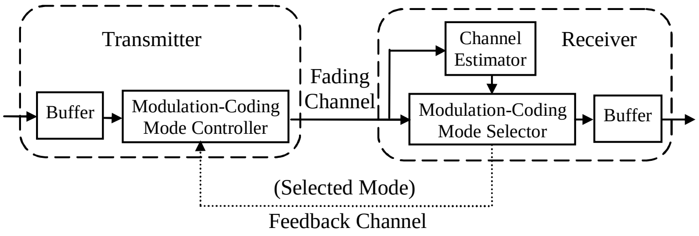

Figure 1. AMC transmitter and receiver

Convolutionally coded M_n-ary rectangular/square QAM adopted from the HIPERLAN/2, or, the IEEE 802.11a standard [11] listed in TABLE I. There are more transmission modes in other standards like CDMA2000, HSDPA and WPAN which can be found in [12], [13], [14].

TABLE I. TM TRANSMISSION MODES

Mode *n*	**1**	**2**	**3**	**4**	**5**
Modulation	BPSK	QPSK	QPSK	16QAM	64QAM
Coding Rate	1/2	1/2	3/4	3/4	3/4
R*n*	0.5	1.0	1.5	3.0	4.5

A scheduler which always selects the user who currently can transmit with the highest rate (has the best channel) will maximize system throughput. This scheduler is referred to as Maximum Rate (MR). At the other side, Shortest Residual Time First (SRTF) by scheduling the user who mostly needs service to meet QoS requirements (but probably does not have best channel), at the expense of decreased system throughput will provide QoS guarantee. Obviously, there is a tradeoff between providing QoS guarantee and optimizing throughput. Many efforts has been accomplished in the literature before and several approaches have been developed which provide some QoS while still achieving reasonable throughput [15], [16], [17]. However, adaptively compromising this tradeoff has been also the challenging subject of some efforts in the literature and few approaches have been proposed which tune the tradeoff in an adaptive manner. The most prominent of proposed strategies is UF [18] which allows for an adaptive tradeoff between QoS and throughput. Furthermore, the UF considers all three metrics of QoS i.e. throughput, delay and loss for users. The UF calculates a measure, namely, the *Residual Time* for individual QoS metrics and schedules users on the basis of this measure. By properly selecting parameter values, the UF can operates between the MR scheduler –which only considers throughput optimization– and the SRTF scheduler –which only considers QoS provisioning– adaptively depending on current cell urgency level (CUL).

The reminder of this paper is organized as follows: in section II, a new measure for analyzing the performance of schedulers which provide 3D-QoS guaranteeing will be introduced and based on this measure a novel scheduler, namely BQDF will be developed. In section III, the BQDF will be extended to AQDC scheduler which assumes AMC at physical layer and makes it possible to tune the tradeoff between QoS and throughput adaptively. Then a CAC policy for the system will be proposed. In section IV, simulation environment and model is described, before simulation results are presented and discussed. Some future directions on this matter are listed in Section V and finally, section VI concludes this paper.

II. DESIGN PRELIMINARIES

A. *QoS Deviation*

One of the most important properties of a scheduler is fairness. In wireless schedulers, efforts to increasing throughput and efficient resource utilization will lead to degrading fairness. In most schedulers, a weight is assigned to each flow (user or session) based on required portion of total bandwidth and flows are serviced according to their weights. In these schedulers, fairness is not an equal distribution of resources (bandwidth) between flows, but each flow is serviced proportionally to its reserved portion. Therefore, the fairness measure is described precisely as:

$$E = \left| \frac{W_f(t)}{w_f} - \frac{W_g(t)}{w_g} \right| \qquad (1)$$

where $W_f(t)$ and $W_g(t)$ are received service rate by flow f and g, respectively in $[0, t]$, w_f and w_g denote reserved portion of total bandwidth for flow f and g, respectively. The more E is near to zero, the fairer scheduler performs. In these schedulers, QoS has only one dimension and that is received service rate (throughput). By using the fairness measure as defined above, it is possible to assess scheduler performance in terms of QoS provisioning quite accurately. However, some schedulers pay attention to QoS on all three metrics i.e. throughput, delay and loss. In this case, QoS is a three dimensional concept and therefore, the fairness as mentioned is not a suitable measure to assess the scheduler performance. Now we propose a new generalized measure, the *QoS Deviation* in which 3D-QoS is concerned. This measure can be employed to analysis and compare the performance of schedulers which deal with all three aspects of QoS simultaneously .

Suppose that QoS constraints on the flow are mean data rate (throughput) R, maximum mean packet delay D and maximum mean packet loss ratio L. we then have:

$$\begin{cases} R(t) \geq R \\ D(t) \leq D \\ L(t) \leq L \end{cases} \qquad (2)$$

The mean data rate is given by:

$$R(t) = \frac{S_b(t)}{t_{BL}} \qquad (3)$$

where $S_b(t)$ denotes the transmitted data in $[0, t]$ and t_{BL} is the flow backlogged time. Note that only backlogged time is taken into account to calculate the mean data rate.

The mean packet delay is denoted as:

$$D(t) = \frac{T(t)}{S_p(t)} \qquad (4)$$

where $T(t)$ is the accumulated delay of all past packets and $S_p(t)$ is the number of packets sent in [*0*, *t*].

D(*t*) is also updated at each iterative interval according to:

$$D(t_{n+1}) = \frac{D(t_n) \times \lfloor S_p(t) - 1 \rfloor + t_w}{S_p(t)} \qquad (5)$$

where t_w is the time that the HOL packet is already waiting for service so far.

The mean packet loss ratio is represented by:

$$L(t) = \frac{L_p(t)}{L_p(t) + S_p(t)} \qquad (6)$$

where $L_p(t)$ is the number of packets dropped in [*0*, *t*] due to queue fullness or excess delay.

Let us define new variables as:

$$r(t) = \frac{R}{R(t)} \qquad (7)$$

$$d(t) = \frac{D(t)}{D} \qquad (8)$$

$$l(t) = \frac{L(t)}{L} \qquad (9)$$

Substituting these new variables into QoS constraints in Equation (2), we get:

$$\begin{cases} r(t) \le 1 \\ d(t) \le 1 \\ l(t) \le 1 \end{cases} \qquad (10)$$

Now suppose the three-dimensional space (*r*, *d*, *l*). The QoS guaranteed locus is a unity cube as depicted in Figure 2. We call this cube, the QoS guaranteed unity cube or briefly QoS cube.

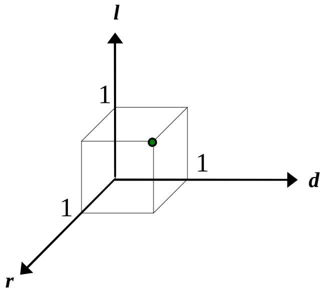

Figure 2. The QoS guaranteed unity cube

If $r_i(t)$, $d_i(t)$ and $l_i(t)$ denote mean data rate, mean packet delay and mean packet loss ratio for flow *i* at time *t*, respectively; the point ($r_i(t)$, $d_i(t)$, $l_i(t)$) in the space is called QoS work point of flow *i* at time *t*. For each flow whose the QoS work point is onto the QoS cube, the QoS is guaranteed and vise versa flow's QoS is guaranteed if the flow's QoS work point is onto the QoS cube. However, where is the efficient point of system operation in this space? Before finding the efficient point, let us make the concept of fairness more clear. Fairness is that available resources are allotted to all flows according to their needs but no a single flow is serviced more than needed amount to guarantee its QoS. Therefore, in order to realize fairness, scheduler should try to provide QoS guarantee with minimum service to each flow. Instead, it will allocate channel to flows which need more service to guarantee their QoS. For instance, if the work point of a flow is (0.6, 0.8, 0.7), QoS is guaranteed for it because the work point is in the QoS cube. However, scheduler can stop servicing the flow and still meets its QoS requirements and forces it to experience more delay and loss while decreasing its throughput. Normally, this penalty continues until no violation occurs in QoS constraints, that is, the work point still remains in the QoS cube. According to Equation (10), the maximum tolerable value for QoS guarantee in each dimension is 1 at which the constraint is met with minimum service. In other words, 1 is the boundary value between QoS guarantee in each dimension and receiving excess service in that dimension. So scheduler should try to move flows work point to point (1, 1, 1) to prevent servicing flows more than their needs. Therefore, it is understood that the efficient point for system operation is (1, 1, 1) in which in addition to distribute resources between flows fairly, their desired QoS is guaranteed. We call this point the QoS efficient point. Note that even if one of the QoS constraints is violated for a flow, its work point will be located out of the QoS cube and in this condition, the Euclidean distance between the work point and the efficient QoS point indicates the difference between the current flow QoS and desired QoS. Besides, when all three QoS constraints is met, the work point is in the QoS cube and in this condition the Euclidean distance between the work point and the efficient QoS point implicates the penalty that the flow can tolerate without QoS violation while receiving no service. With attention to what explained above, now we can declare a measure, the *QoS Deviation*, as:

$$d_{Q_i} = \begin{cases} -\left\| \vec{Q}_i - \vec{U} \right\| \ , \ \left\| \vec{Q}_i \right\| \le 1 \\ +\left\| \vec{Q}_i - \vec{U} \right\| \ , \ \left\| \vec{Q}_i \right\| \ge 1 \end{cases} \qquad (11)$$

where symbol $\|.\|$ denotes the vector norm, $\vec{Q}_i$ is the location vector of flow *i* QoS work point and $\vec{U}$ is the location vector of efficient QoS point.

Equation (11) implicates that when the QoS work point is in the QoS cube, d_Q is negative and otherwise d_Q is positive. Generally, total variation range of QoS Deviation is:

$$-\sqrt{3} < d_{Q_i} < \infty \qquad (12)$$

In order to omit the effect of an arbitrary dimension in the QoS Deviation, simply set the corresponding dimension variable to unity for all times. For example by setting *r(t)*≡1, the dimension *r* is ignored and two dimension QoS Deviation (delay and loss) will be obtained or by setting both *r(t)*≡1, *l(t)*≡1 QoS Deviation is affected only by the dimension *d* (delay). This is useful for studying the performance of schedulers in terms of only one or two certain aspects while other aspects are ignored.

B. QoS Deviation based scheduling

In order to meet QoS requirements, the user U who has the biggest d_Q is always selected to be serviced by scheduler :

$$U = \max_i \{d_{Q_i}\} \tag{13}$$

where $\{d_{Q_i}\}$ denotes the QoS Deviation of all flows in the cell. We call this scheduler BQDF[1]. The BQDF tries to minimize the QoS violation level of the cell by servicing the most urgent flow. The BQDF and its peer the SRTF will find the most urgent flow using their measure. The urgency measure in the BQDF and the SRTF is QoS Deviation and Residual Time, respectively

C. Transmission Urgency within the cell

Now that we have a measure for the urgency of single flows, we are furthermore defining the *Cell QoS Deviation Level* (CDL) as a measure for the general urgency of transmissions within the cell. The CDL will be used to tune the tradeoff between QoS and throughput. However, this measure is also useful for CAC. Based on the QoS Deviation of the individual flows, we can, for instance, define the CDL based on the QoS Deviation of the most urgent flow in the cell:

$$CDL = \max(\{d_{Q_i}\}) \tag{14}$$

Alternative definitions of the CDL can be based on the mean or standard deviation of the QoS Deviation of all flows in the cell:

$$CDL = \text{avg}(\{d_{Q_i}\}) \tag{15}$$

$$CDL = \sqrt{\text{var}(\{d_{Q_i}\})} \tag{16}$$

III. System Design

A. AQDC scheduler

The introduced BQDF scheduler tries to guarantee QoS for users regardless of their channel state information (CSI). In a wireline network in which channel has no variation, the scheduler performs very well in terms of throughput optimization. However, this performance is not acceptable in a wireless channel with time and frequency fading in which available bandwidth is changing with respect to time and location frequently. Degrading throughput in this condition may also influence QoS provisioning severely. To solving this, we present AQDC[2] scheduler which exploits both CSI and QoS Deviation to schedule users. The AQDC assumes that all users rely on AMC at physical layer and each user is possessed with multiple transmission modes that one of them is chosen by AMC based on channel status. The AQDC schedules flows according to the following algorithm:

After calculating the QoS Deviation of all flows in the cell, p percent of flows which mostly need service to meet their QoS are selected. The p is called *candidate percentage*. This selection is done by comparing $\{d_{Q_i}\}$ with a threshold level denoted by $d_Q(p)$ as follows:

$$C = \{\ flow_i \,|\, d_{Q\,i} \geq d_Q(p)\,\} \tag{17}$$

The candidate set C comprises first p percent of the most urgent flows. The threshold $d_Q(p)$ is calculated based on the candidate percentage p and flows Qos Deviation $\{d_{Q_i}\}$ in order to be less than the Qos Deviation of p percent of flows. If we assume that only one flow can transmit at a time, the AQDC now will service to flow from candidate set C which can transmit with the maximum data rate, in order to optimize the overall system throughput. If there is no flow in C which can be serviced due to deeply fading channel, the AQDC will increment p by *p_step* and the algorithm will be restarted. As a result, this time the candidate set C will become bigger and will contain more flows. Therefore, the probability of that no candidate flow can transmit data will decrease. The algorithm will continue until a flow is chosen from C or no flow can be serviced (due to bad channel) while C comprises all flows in the cell (p=1).

The init value of candidate percentage i.e. p_0 at the start of the algorithm is not fixed, but is rather a function of the CDL[3] according to Equation (18):

$$p_0 = \exp(-a \times [\max(\{d_{Q_i}\}) + \sqrt{3}]) \quad , a \geq 0 \tag{18}$$

Equation (18) represents that the better QoS of the worst flow (biggest d_Q among all flows) becomes, that is, its d_Q decreases, the more p_0 increases in order to let the candidate set to become bigger and there is more chance to optimize system throughput. Vise versa, the worse it becomes, this time by decreasing p_0, system will act according to guarantee QoS at the expense of degrading throughput. As a result, the AQDC will be able to tune the compromise between QoS provisioning and optimizing throughput in an adaptive manner depending on the current CDL.

Figure 3 depicts these variations with different values for parameter a. By using parameter a, it is possible to modify the AQDC attention to QoS provisioning with respect to optimizing throughput. The more parameter a is selected, the more QoS provisioning is highlighted. By setting $a = \infty$, the AQDC performs exactly like the BQDF and only regards guaranteeing QoS. In other hand, the less parameter a is selected, the more throughput optimization is taking place. By setting $a = 0$, the MR scheduler is obtained and this will lead to pure throughput optimization.

Best-effort users are also serviced by the AQDC in three cases:

1) There is no user waiting for service.
2) All flows have negative QoS Deviation.
3) None of users is able to transmit data due to bad channel.

[1] Biggest Qos Deviation First

[2] Adaptive Qos Deviation Control

[3] Here we use the CDL according to Equation (14).

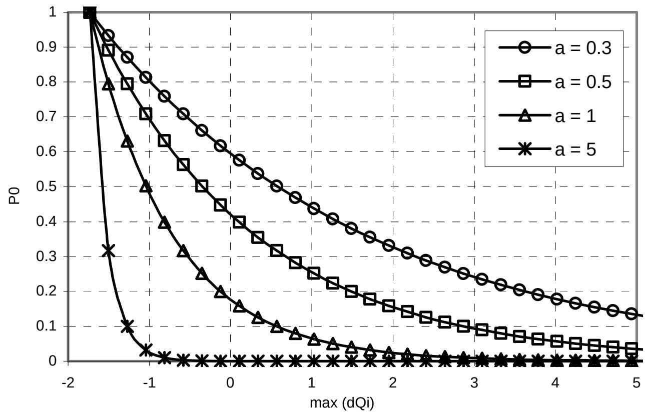

Figure 3. p_0 versus max(d_{Qi}) with different values for parameter a

B. *Call Admission Control policy*

Here we offer a CAC policy for the proposed system according to the following algorithm:

When a new call arrives, it will be accepted if it is the first one. Otherwise, if concurrent calls in the cell exceed a certain value –which is determined by the maximum tolerable capacity of system– the call is rejected. If none of these two cases occurs, system first calculates the QoS Deviation of all flows in the cell. Then QoS Deviations are compared to a threshold level that is called *rejection threshold*. The arrival call is admitted to system only if less than a certain percent which is called *admission percentage* of flows, have unacceptable QoS. In other words, the arrival call is rejected if the percentage of flows with QoS Deviation greater than the rejection threshold exceeds the admission percentage. Best-efforts calls are always accepted to system with no consideration.

Typical values for admission percentage and rejection threshold is 10% and 0.87, respectively. This value for rejection threshold is equivalent to 50% violation from desired QoS[1]. In this way, calls are accepted if no more than 10% of flows have QoS Deviation greater than 0.87.

IV. Simulation Model and SCenario

The simulation environment that we have used is NS2 [19]. We assume a single cell with a BS at the center which controls the calls between MSs and the scheduler is located there. Actually, the performance of schedulers is greatly impacted by hand-off users between cells [7]. However, here we consider only one cell without hand-off in order to compare the fairness of different schedulers. Data packets are forwarded by the BS to MSs and there is no possibility of direct communication between MSs (ad-hoc network). There is all information about call setup, QoS classes' specifications and each user CSI in the BS. Calls are generated according to Poisson model with average 2 calls per second and call duration time has exponential distribution with 60 seconds average. Call QoS class is selected uniformly from four QoS classes, each one with specific requirements as illustrated in TABLE II. We use four different traffic for various QoS classes including real audio, exponential variable bit rate (VBR), Pareto VBR and constant bit rate (CBR). The bit rate of each traffic type is adjusted so that the required QoS minimum bit rate is achieved averagely. The last column in the table shows the maximum tolerate time that a packet can wait to receive service. Arrival packets to the BS are stamped with the arrival time and HOL packets waiting more than the maximum tolerable delay will be force dropped by the system.

TABLE II. QoS Classes

Class	Traffic Type	Min Mean Rate (Kbps)	Max Mean Loss (percent)	Max Mean Delay (ms)	Max Delay (ms)
1	Audio	2.0	0.05	100	550
2	Expo VBR	4.0	0.07	150	850
3	Pareto VBR	6.0	0.10	200	1100
4	CBR	8.0	0.12	250	1500

At the call admission time, a unique Id is assigned to each flow by the system. In the BS, on the basis of this Id, packets of each flow are pushed into a distinct FIFO queue with finite length set to 5 packets. If a queue is full, next arrival packets of the corresponding flow are tail dropped.

The transmission modes set which is used by nodes in the network are listed in TABLE III. Mode n = 0 is the condition that channel is deeply fading and transmission is not possible. Channel in mode n = 5 is at the best condition and user can transmit with maximum rate i.e. R_{max} which is set to 128 Kbps. Channel is constant for a duration of 250 ms and is then randomly changed. At the end of each constant interval, a mode is chosen from the transmission modes set uniformly.

TABLE III. Transmission Modes Set

Mode *n*	**0**	**1**	**2**	**3**	**4**	**5**
Trans. Rate	0	$0.33R_{max}$	$0.5R_{max}$	$0.67R_{max}$	$0.75R_{max}$	R_{max}

We analyze and compare the performance of four different schedulers including the AQDC, the UF, the BQDF and the MR schedulers. By adjusting the CAC parameters, the simulation is performed in two various cases. In case I, the CAC parameters are chosen so that no call is rejected by the system. Whatever interested here is the performance of scheduler in terms of QoS provisioning. In case II, we let the CAC to controls call admissions so that QoS is provided at a certain level and then system capacity in the sense of the number of successful flows will be assessed.

A. *Case I*

In this case, rejection threshold is set to a high value e.g. 1000 and admission percentage is set to 100%. As a result, all calls are admitted by the system. However, in order to control the arrival data rate and prevent extreme congestion, the number of concurrent calls is limited to 60. This will lead to about 300 Kbps arrival data bit rate to system that is 2 times more than the maximum available channel bandwidth approximately. Now in such a critical condition, we assess the

[1] 50% violation in QoS means $r(t)$=1.5, $d(t)$=1.5, $l(t)$=1.5, now we have:
$$\sqrt{(r(t)-1)^2+(d(t)-1)^2+(l(t)-1)^2} = \sqrt{3\times(0.5)^2} \cong 0.87$$

QoS violation level of the cell (QoS provisioning) with regard to the overall system throughput.

Figure 4 illustrates the mean 3D-QoS Deviation versus time for different schedulers. We observe that the AQDC scheduler performs best compared to all other approaches and its QoS Deviation is about 6 averagely at steady state. In other hand, the MR scheduler performs worst among all other schedulers (except the UF scheduler) in QoS provisioning. This is because the MR scheduler is not considering QoS but is concerned only with throughput optimization. The BQDF scheduler with QoS Deviation equal to about 7, performs worse than the AQDC scheduler and better than both the UF and the MR schedulers. We also observe that the QoS Deviation in the UF scheduler is not reasonable. This seems in contrast with the general notion that the UF has a good performance in wireless fading channels as has been described in [18]. However, the key is that the UF scheduler performs exactly like the SRTF scheduler in high load traffic conditions and does not care users channel fluctuations (see [18]). The graph confirms that users in the AQDC and BQDF experience less deviation form their desired QoS than users in the MR or UF scheduler.

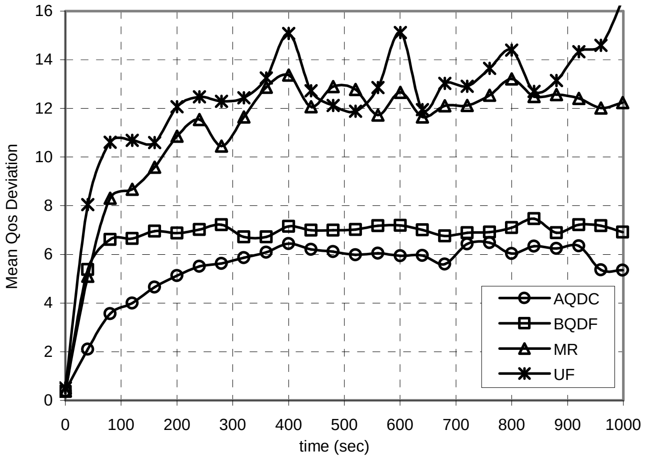

Figure 4. *Mean 3D-QoS Deviation versus time (Case I)*

Let us now take a closer look at the performance of schedulers in terms of per-dimension QoS provisioning. The results for throughput, delay, and loss are plotted in Figure 5 to Figure 7, respectively. These figures enable us to assess and compare the performance of schedulers in terms of providing QoS guarantee in each dimension individually. We observe that the AQDC and the BQDF schedulers have a close contention in the rate and the delay dimensions QoS provisioning and perform best compared to other schedulers in both dimensions. The graphs show that the UF scheduler performs worst in both rate and delay dimensions among all other schedulers. Figure 7 shows that in the loss dimension QoS provisioning, the AQDC scheduler performs best and the MR scheduler performs worst. The pieces of curves below the zero line (negative QoS Deviation) indicate the times in witch QoS is provided more than required amount in corresponding dimension. As expects, this happens only at the first of simulation when data traffic is not yet very high. For instance, in the loss dimension, this condition continues for the AQDC scheduler until 40 seconds elapsed from the simulation start time. Note that although the channel-ignore BQDF scheduler performs better than the channel-aware MR and UF schedulers in terms of QoS provisioning, but it will degrade the overall system throughput markedly. In fact, optimizing throughput is also one of the most important of a “good” scheduler [7], [20].

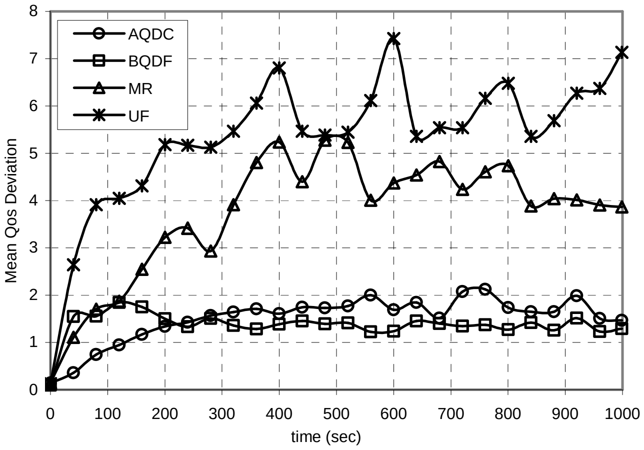

Figure 5. *Mean Rate 1D-QoS Deviation versus time (Case I)*

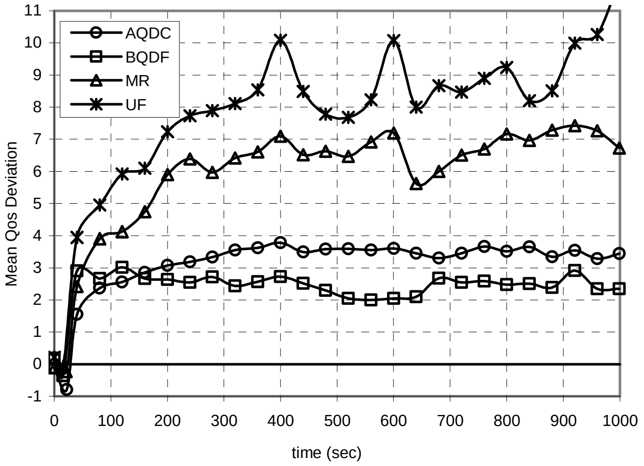

Figure 6. *Mean Delay 1D-QoS Deviation versus time (Case I)*

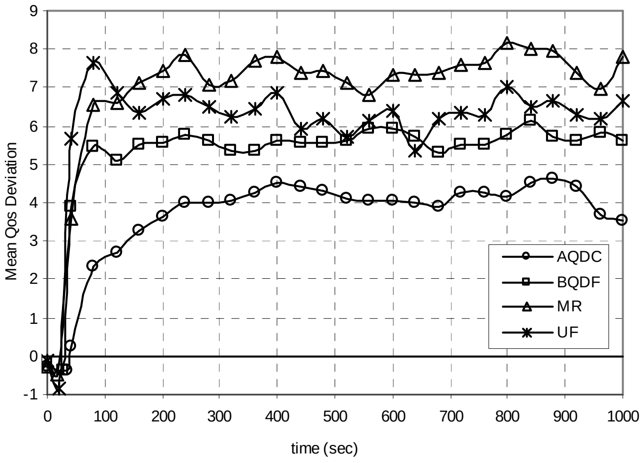

Figure 7. *Mean Loss 1D-QoS Deviation versus time (Case I)*

Figure 8 is an interesting one which shows the QoS Deviation of the worst user from the viewpoint of QoS (the most urgent user) in the system. We observe that users in the MR and UF schedulers may be QoS violated much severely than users in the AQDC and BQDF schedulers.

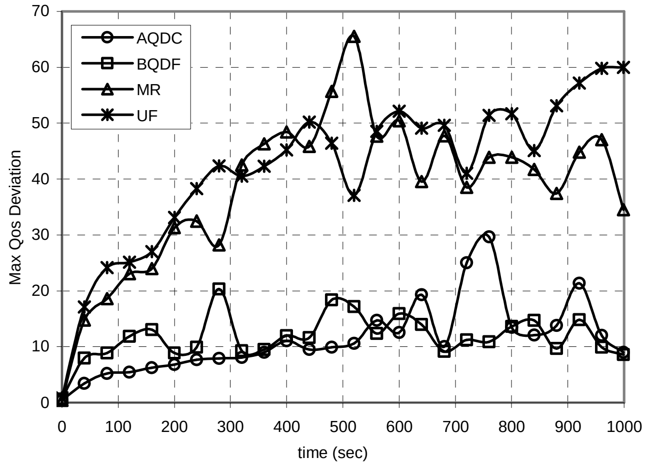

Figure 8. *Max 3D-QoS Deviation versus time (Case I)*

Packet drops rate in the schedulers are depicted in Figure 9. As observes, in the AQDC scheduler, least packets are dropped and the UF drops most packets compared to other schedulers. The packet drop rate in the AQDC and the UF scheduler is 80 packets per second (pps) and 135 pps, respectively. Packet drop rate in the BQDF is a little more than packet drop rate in the MR scheduler. However, the BQDF performs fairer than the MR scheduler does because its loss 1D-QoS Deviation (Figure 7) is less for all times.

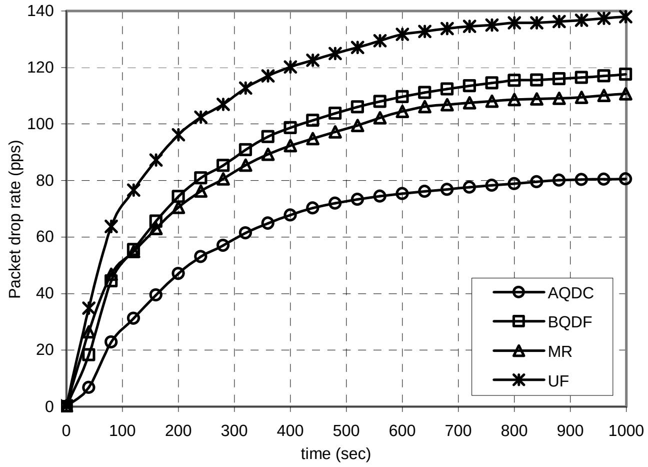

Figure 9. *Packet drop rate versus time (Case I)*

Figure 10 depicts the throughput of the system with various schedulers. As expected, we observe that the MR scheduler achieves the best throughput among other schedulers. The graph shows that the UF scheduler does not achieve an acceptable throughput compared to the AQDC or MR scheduler. This is also because of the previously mentioned fact that the UF scheduler performs exactly like the SRTF scheduler in high load traffic conditions and schedules users regardless of their channel status which will lead to wasting resources.

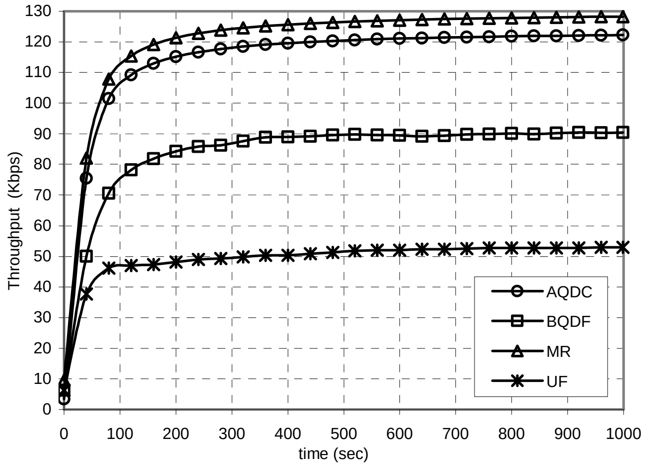

Figure 10. *System throughput versus time (Case I)*

B. *Case II*

In this case, rejection threshold is set to 0.87 and admission percentage is set to 10%. The maximum number of concurrent calls is also set to a high value e.g. 5000 in order that no call is rejected due to the lack of system resources. Now in this condition, we assess the system capacity with regard to the overall system throughput. The capacity here is the number of successful users which is obtained by multiplying the total number of admitted users to system by the percentage of successful users i.e. users with no QoS violation.

Figure 11 depicts the number of admitted users to system with different schedulers. We observe that the AQDC scheduler with about 450 admitted users more than the MR scheduler performs best among other schedulers and admits 450 users to system totally. Both the BQDF scheduler and the UF scheduler admit the same number of users approximately and service 230 users.

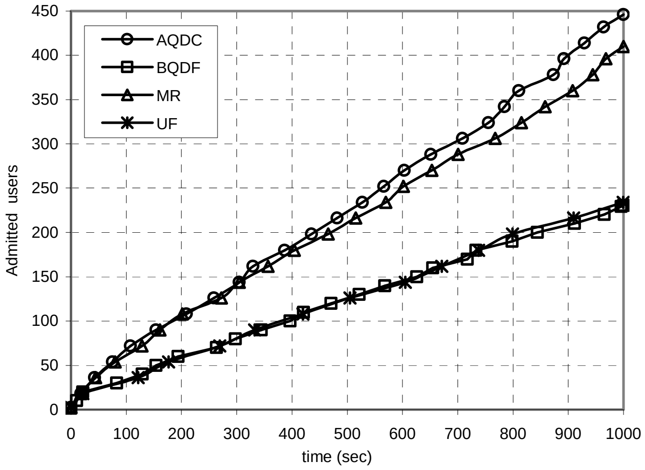

Figure 11. *Number of admitted users versus time (Case II)*

Figure 12 illustrates the percentage of successful users to admitted users versus time in the system. We observe that the AQDC scheduler performs best compared to all other schedulers and services about 50% of users successfully. In other hand, the BQDF scheduler performs worst and actually provides no QoS guarantee. This is because the BQDF schedules users regardless of their channel conditions and wastes the resources. The performance of the UF scheduler is not too bad here compared to the MR and BQDF schedulers and services 30% of users successfully. However, why the CAC does not rejects more calls while the BQDF cannot guarantee QoS for users? Figure 11 shows that the same number of users are admitted to system in both the BQDF and the UF schedulers whereas Figure 12 shows that the percentage of successful users in the UF is 30% more than the BQDF. This seems to be a drawback in the CAC and the problem is under further study.

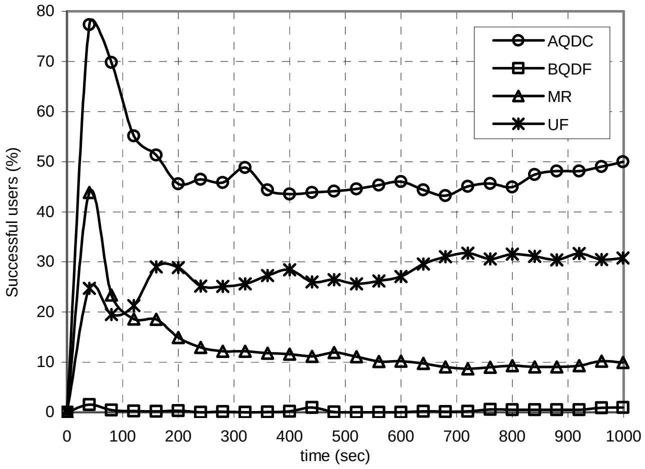

Figure 12. *Percentage of successful users to admitted users (Case II)*

The number of successful users in the system is depicted in Figure 13. As expected, the AQDC scheduler performs best among all other schedulers and services 210 users successfully. The number of successful users in the UF, MR and BQDF is 70, 35 and 5, respectively.

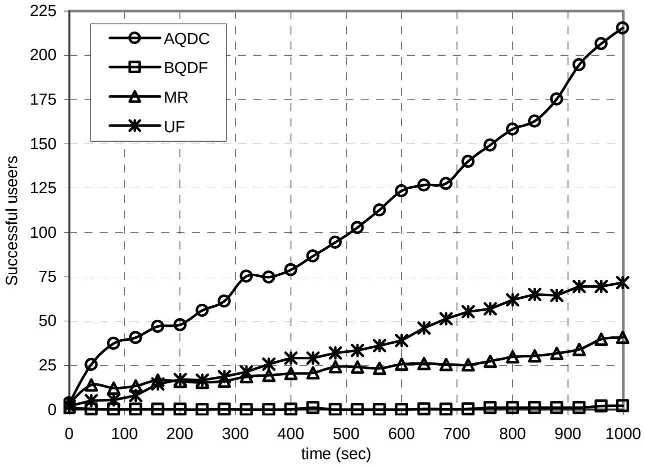

Figure 13. *Number of successful users versus time (Case II)*

Packet drops in the system are depicted versus time in Figure 14. Note that in this case despite of the case I, the arrival packet rate to system is not the same for different schedulers and depends on the number of admitted users. The more users are admitted to the system, the more packet drops is expected. However, we observe that packet drops experience smoothest slope in the AQDC and sharpest slope in the UF scheduler compared to other approaches. As observes, least packets in the AQDC scheduler and most packets in the UF scheduler are dropped.

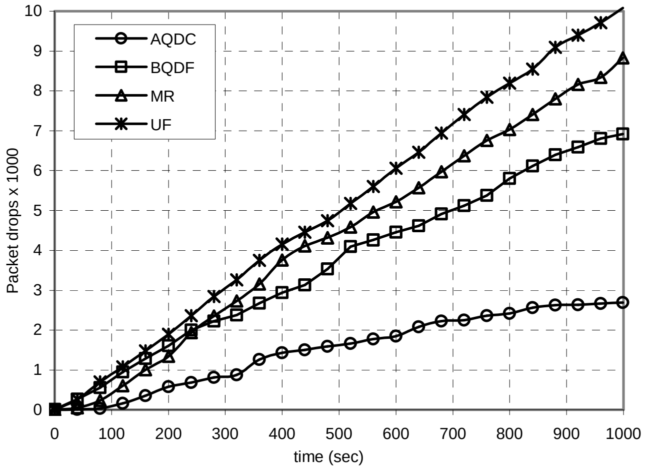

Figure 14. *Packet drops versus time (Case II)*

Finally, Figure 15 illustrates the throughput of the system versus time. The throughput here is also influenced by the system capacity. We observe that the AQDC scheduler performs well and achieves a throughput close to the MR scheduler. As observes, throughput of the BQDF scheduler decreases markedly here with compared to the case I.

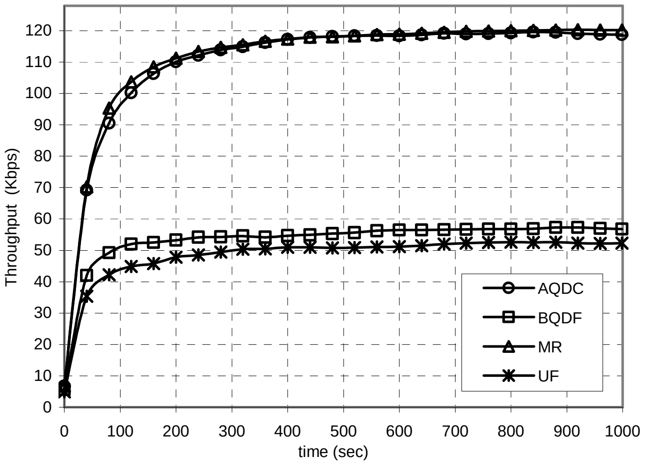

Figure 15. *System throughput versus time (Case II)*

V. Future Works

1) We established a three-dimensional space for QoS by using a specific mapping on throughput, packet delay and

packet loss. However, the obtained basis vectors are not orthogonal completely and there is some cross correlation between them. Finding more orthogonal base vectors for QoS will enhance efficiency of the QoS-Deviation measure and related schedulers.

2) The proposed CAC policy is primarily designed for the AQDC scheduler and its functionality for other schedulers may be sometimes inefficient. Therefore, designing a more intelligent QoS Deviation-based CAC which can be applied to all schedulers regardless of their structure is under further investigation.

3) In this paper, we explored single cell connections without any hand-off. Since the QoS is affected by hand-off (horizontal and vertical) severely in the real world, it is necessary to assess the performance of the AQDC scheduler in a system with handoff and enable it with handoff-aware scheduling.

VI. Concluding Remarks

We proposed a novel scheduler design in cellular packet-switched wireless networks which provides QoS for users in all three aspects of QoS i.e. throughput, delay and packet loss simultaneously. We established a 3D space with specific basis vectors for QoS and found the efficient point of system performance in that space. Then we developed a generalized measure, the *QoS Deviation*, which is the Euclidean distance between the user QoS work point and the efficient QoS point in the 3D space. Based on this measure, a scheduling approach, namely BQDF is outlined and will be extended to AQDC scheduler for wireless channels. The channel is time and frequency fading and using the AMC at PHY layer enables users with multirate transmission. The AQDC scheduler makes it possible to tune the tradeoff between QoS provisioning and optimizing throughput in an adaptive manner depending on current cell QoS Deviation level (CDL). We also introduced a CAC policy for the proposed system which exploits the QoS Deviation to make call admission or rejection decisions. By adjusting the CAC parameters, system performance are assessed in two various cases with four schedulers including the AQDC, the UF, the BQDF and the MR schedulers. In case I, the scheduler performance in terms of QoS provisioning with regard to the system throughput is analyzed and compared. The results implicate that the AQDC scheduler performs best among all schedulers and make users less QoS violated. In case II, the system capacity with regard to the system throughput is analyzed and compared. The results confirm that the AQDC scheduler also performs best in terms of the system capacity and admits more users to the system while services more users successfully compared to all other schedulers. The results also indicate that our proposed measure, the QoS Deviation, obviously is very successful to act as an urgency measure to find the mostly service needed user, as well as can be employed as a generalized measure to assess the performance of schedulers in terms of QoS provisioning.

References

[1] Vincent Huang and Weihua Zhuang, “QoS-Oriented Packet Scheduling for Wireless Multimedia CDMA Communications,” *IEEE Transactions On Mobile Computing*, Vol. 3, No. 1, January-March 2004.

[2] Qingwen Liu, Shengli Zhou, and Georgios B. Giannakis, “Cross-Layer Scheduling With Prescribed QoS Guarantees in Adaptive Wireless Networks,” *IEEE Journal On Selected Areas In Communications*, Vol. 23, No. 5, May 2005.

[3] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” *IEEE Communication Magazine*, vol. 41, no. 10, pp. 74–80, Oct. 2003.

[4] J. Liebeherr and N. Christin., “JoBS: Joint buffer management and scheduling for differentiated services,” *In Proceedings of IWQoS 2001*, pages 404.418, Karlsruhe, Germany, June 2001.

[5] Lars Wischhof, John W. Lockwood, “Packet Scheduling for Link-Sharing and Quality of Service Support in Wireless Local Area Networks,” Department of Computer Science, Applied Research Lab, Washington University, November 2001.

[6] H. Zhang, “Service Disciplines for Guaranteed Performance Service in Packet-switching Networks,” *Proc. IEEE*, vol. 83, pp. 1374–96, Oct. 1995.

[7] H. Fattah and C. Leung, “An overview of scheduling algorithms in wireless multimedia networks,” *IEEE Wireless Communication Mag.*, vol. 9, no. 5, pp. 76–83, Oct. 2002.

[8] S. Valentin, J. Gross, H. Karl, and A. Wolisz, “Adaptive Scheduling for Heterogeneous Traffic Flows in Cellular Wireless OFDM-FDMA Systems,” University of Paderborn, Germany, *IEEE Wireless Communication Mag*., March 2005.

[9] Qingwen Liu, Shengli Zhou, and Georgios B. Giannakis, “Queueing with adaptive modulation and coding over wireless links: Cross-layer analysis and design,” *IEEE Trans.Wireless Communication*, vol. 4, no. 3, May 2005.

[10] Q. Liu, S. Zhou, and G. B. Giannakis, “Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links ,” *IEEE Transaction Wireless Communication*, vol. 3, no. 5, pp. 1746–1755, Sep. 2004.

[11] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P.Karlsson, “A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards,” *IEEE Commun. Mag*., vol. 40, no. 5, pp. 172–180, May 2002.

[12] Physical Layer Aspects of UTRA High Speed Downlink Packet Access (Release 4), 2001. 3GPP TR 25.848 V4.0.0.

[13] Physical Layer Standard for cdma2000 Spread Spectrum Systems, July 1999. 3GPP2 C.S0002-0, Version 1.0.

[14] J. Karaoguz, “High-rate wireless personal area networks,” *IEEE Commun. Mag.*, vol. 39, no. 12, pp. 96–102, Dec. 2001.

[15] S. Borst., “User-level performance of channel-aware scheduling algorithms in wireless data networks”, *In Proceedings Infocom 2003*, San Francisco, CA, March 2003.

[16] P. Liu, R. Berry, and M. L. Honig., “Delay-sensitive packet scheduling in wireless networks,” *In Proceedings WCNC 2003*, New Oreans, Lousiana, March 2003.

[17] P. Viswanath, D. N. C. Tse, and R. Laroia. “Opportunistic beamforming using dumb antennas,” *IEEE Transactions on Information Theory*, 48(6):1277–1294, June 2002.

[18] Christian Hartmann, Robert Vilzmann, and Jorg Eberspacher, “Channel Aware Scheduling for User-Individual QoS Provisioning in Wireless Systems,” *In Proceedings of IEEE Vehicular Technology Conference, VTC2004*, pp. 1009-13, Sep. 2004.

[19] URL: http://www.isi.edu/nsnam/ns/.

[20] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, Oct. 2003.