Multiclass Priority Fair Queuing for Hybrid Wired/Wireless Quality of Service Support

Jay R. Moorman
jrmoorma@uiuc.edu
Beckman Institute
University of Illinois

John W. Lockwood
lockwood@ipoint.vlsi.uiuc.edu
Beckman Institute
University of Illinois

Abstract

The widescale deployment of packetized wireless network services currently lacks a mechanism to provide QoS in the framework of existing backbone networks. This paper describes a wireless scheduling algorithm that can provide QoS bounds to ATM traffic. It transforms the standard ATM traffic classes into parameters for a priority, fair queuing algorithm, implemented at the wireless MAC layer. By supporting existing ATM traffic, the scheduler avoids the need to redefine QoS specifically for the packetized wireless channel. Bounds are derived for delay and throughput on the individual CBR and rt-VBR virtual connections.

The scheduler maps ATM QoS parameters into priorities and weights for the wireless MAC. The protocol supports CBR, rt-VBR, nrt-VBR, ABR, and UBR traffic. Real-time data uses a wireless fair queuing model. Non-real-time data is processed by a weighted round robin scheduler and a recirculating service queue that guarantees minimum cell rate (MCR). Best-effort traffic is serviced using a simple FIFO mechanism. Buffers are shared across priorities to make more efficient use of memory.

1 Introduction

As the demand for wireless data connectivity grows, the type and diversity of applications have placed new demands on the scheduling algorithms for packet transmission on the wireless link. Extremely limited bandwidth must be shared among many flows of traffic, each with different requirements in terms of delay, bandwidth and loss. We propose an algorithm that fairly shares this wireless link bandwidth among ATM classes, and provides bounded delay for real-time traffic.

The notion of QoS over a wireless link has been the focus of much recent research. Work has been done to support certain bandwidth and delay requirements over bandwidth limited links [1, 2, 3]. Some of this has been specific to wireless ATM traffic [4, 5, 6], focusing on virtual circuit call admission, or scheduling for a particular type of traffic. In the wireline domain, a large amount of research has been done with packet-scheduling schemes to approximate fluid fair queuing. This work includes weighted fair queuing (WFQ) [7], worst-case fair weighted fair queuing (WF^2Q) [8], self-clocked fair queuing (SCFQ) [9], start-time fair queuing (STFQ) [10], hierarchical packet fair queuing (H-PFQ) [11], and the many other variations. More recently, variations of these algorithms have been developed that address issues specific to a wireless link [12, 13, 14]. These wireless queuing algorithms are similar to the wired fluid fair queuing schemes, but they must additionally compensate flows for wireless channel errors that tend to be both bursty and location-dependent.

The contribution of this paper is the development and analysis of a wireless scheduling algorithm to provide QoS bounds to all the ATM traffic classes as defined in [15]. This system allows the wireless media to use traffic with current ATM QoS specifications. It is unnecessary to define new QoS requirements specific to the wireless channel when these have already been defined and supported in the wired environment. This paper is a companion paper with [16] where the implementation details of the algorithm are presented along with comparison to a single fair queue scheduler. The work builds on the Wireless Packet Scheduling (WPS) of [13], with modifications to support ATM QoS parameters.

The Multiclass Priority Fair Queuing (MPFQ) algorithm provides the means to transform each ATM traffic class into the necessary parameters for the wireless channel scheduling. By calculating the bounds on the delay and reserving the required bandwidth, each connection's QoS guarantees, within the schedulable region, can be met in the worst case and exceeded in general.

This paper is organized as follows. Section 2 describes the wireless network model. Section 3 proposes a model for prioritized fair queuing of ATM traffic. Fluid fair queuing is discussed in Section 4 in context of the wired network and the wireless network. This section also outlines the modified WPS algorithm. Section 5 extends the original fair queuing algorithm into a prioritized multiclass scheduler, derives the bounded delays, and presents the multiclass scheduler under the wireless fair queuing constraints. Simulation results are presented in Section 6, and finally Section 7 concludes the paper.

2 Model

In this paper we assume a high-speed wired ATM backbone that is extended to a packetized wireless cellular network. The wireless cellular network is divided into partially overlapping cells that transmit on different logical channels. Each cell contains a base station that is connected to the wired network. The base station schedules packets for transmission in its particular cell for both the uplink (mobile host to base station) and the downlink (base station to mobile host) channel.

The channel is said to be in error for a particular flow if the base station cannot communicate with the particular mobile using that flow. Errors are assumed to be location dependent since one mobile experiencing a fade will not affect the transmission of another mobile at a different location. Errors are also assumed to be typically bursty in the wireless environment.

QoS is supported by meeting bandwidth and delay parameters that are negotiated at call admission. These parameters, such as cell rate or maximum delay, are then guaranteed by the network throughout the lifetime of the connection.

In the rest of the paper we will use the notation specified in Table 1. We also do not distinguish between a packet and cell. For all ATM traffic the cells are fixed packet size L_p.

Notation	Definition
$P_{i,n}$	Packet n of flow i.
ϕ_i	The weight for flow i.
ϕ_{k_i}	The weight for flow i in class k.
r_i	The rate for flow i.
λ_{k_i}	The arrival rate of flow i in class k.
$A_{i,n}$	Arrival time of packet n in flow i.
$s_{i,n}$	The virtual start time for packet n in flow i.
$f_{i,n}$	The virtual finish time for packet n in flow i.
L_p	The length of fixed packets (cells) in all flows.
C_k	The relative capacity of class k.
C	Channel Capacity.
V	Virtual time.
Q_{k_i}	The current buffered queue length of flow i in class k.
$D_{i,n}$	Delay for packet n in flow i.
D_k^{max}	The maximum delay for a head-of-line packet of flow i in traffic class k.
W_i	The amount of work for flow i.
F	The set of all flows in the channel
F_k	The set of all flows in class k.
$B(t)$	The set of all backlogged flows at time t.

Table 1: Fair Queuing Notation

3 Multiclass Priority Fair Queuing

The Multiclass Priority Fair Queuing (MPFQ) algorithm defines the mapping of ATM traffic onto a wireless link in order to maintain QoS guarantees. The algorithm provides a novel way to seamlessly support the different types of QoS parameters needed for current applications. This mapping is nontrivial since the characteristics of the classes as defined in [15] must not be violated. By extending ATM QoS into the wireless communication link, the end-to-end flow requirements can be satisfied since the wired ATM network already provides hard guarantees on QoS. The MPFQ maps ATM QoS parameters into weights and priority levels. A scheduling scheme specific to the traffic type is used at each level in order to maintain the QoS guarantees. Real-time traffic uses a fair queuing algorithm specifically adapted to the wireless medium. The non-real-time traffic uses a Weighted Round Robin (WRR) scheme with an adaptation for wireless compensation. The remaining traffic receives best-effort service.

The base station scheduler assigns the traffic flows to the channel according to a hierarchy of priorities. The first two priorities are used for real-time data that requires tight bounds on delay. These delays are satisfied and the channel allocated fairly through the use of the wireless fair queuing model.

The third priority in the system is used for flows that require guaranteed bandwidth, but not tight delay bounds. MPFQ uses a simple WRR scheme to ensure that traffic bounds are met over time. Traffic will be entered into per flow queues as it arrives at the base station. These queues will then be accessed when packets are needed for the outgoing queue. However, this type of traffic is sensitive to loss and requires a modification to the scheme. If the flow detects a channel error, then its packet will be skipped in that round. At a later time, the flow will have a chance to send the packet when the channel is error-free. The third priority uses the WRR scheduling to access both nrt-VBR and ABR traffic at the Minimum Cell Rate (MCR).

The fourth priority in the system also does not require tight delay bounds. It compensates errored flows by recirculated slots for a later time when the channel is good. This is done by keeping the incoming packets in separate packet queues and using a single slot queue to reserve slots for the head-of-line (HOL) packets. Incoming data is added to the per-flow packet queue and a slot is added to the recirculating queue. This slot has a pointer to the packet queue as required by the scheduler. If the HOL slot must wait, due to channel error, it is entered into the back of the recirculating queue, reserving space for later transmission. Once the channel is good, the first slot in the outgoing queue is used to get the HOL packet in the flow's packet queue and transmit it. By recirculating the slots in this way we provide wireless compensation without packet loss.

The lowest priority in the system processes flows that require no guarantees in either bandwidth or delay. Packets are entered in the order of arrival. An arriving packet is placed in the outgoing queue to be sent in FIFO order.

In order to better utilize the buffer space of the base station and client network adapter, the total buffering available to the system shares the buffer space for both the middle- and lower-priority traffic. If the middle-priority flows have available buffer space, the lower-priority flows may use it. However, this space is marked so that any incoming middle-priority traffic can overwrite the buffers holding lower-priority traffic. In the worst case, when the higher-level buffers are full, new incoming packets of the lower priority will be dropped. However, on average the higher priority buffers will not be full, allowing the lower priorities to buffer more data and reduce packet loss.

By scheduling flows within prioritized traffic classes, the MPFQ algorithm distinguishes itself from other wireless algorithms. This differs from class-based queueing (CBQ) [1] by supporting different scheduling policies at different levels, using ATM QoS parameters to specify flow requirements, keeping per-flow state, and compensating flows for channel error on the wireless medium.

4 Fluid Fair Queuing for Wired and Wireless Networks

Fair queuing allows a traffic scheduler to allocate network resources according to the required bandwidth proportions while providing fair access to all nodes within the necessary delay bounds. Wireless fair queuing is performed through implementation of a WPS modified algorithm.

4.1 Wireline Fair Queuing

Fluid fair queuing [7] models the communication channel as a pipe carrying flows of traffic. The model allows any flow i, over the time interval $[t_1, t_2]$, to be granted channel capacity so as to minimize Equation 1, where $W_i(t_1, t_2)$ is the work (measured in packets transmitted) by flow i over time $t_2 - t_1 = \triangle t$.

$$|\frac{W_i(t_1,t_2)}{\phi_i} - \frac{W_j(t_1,t_2)}{\phi_j}| \simeq 0 \quad (1)$$

When only a subset of flows are backlogged ($B(t)$), an individual flow i will be able to transmit at the following instantaneous rate:

$$r_i^{inst} = \frac{\phi_i}{\sum_{j \in B(t)} \phi_j} C \quad (2)$$

To apply these results in a real implementation the packetized approximation (PGPS) is used from [17], also known as Weighted Fair Queuing (WFQ)[7]. A virtual start and finish time is computed for each packet, and the next packet to send is chosen from among those packets that have arrived. The packets are sorted according to the minimum eligible finish time. In order to decouple bandwidth and delay as described in [13] these times are found as follows:

$$s_{i,n} = max\{V(A_{i,n}), s_{i,n-1} + \frac{L_p}{r_i}\} \quad (3)$$

Thus the start time $s_{i,n}$ is chosen as the larger of either the scheduled start time according to the packet arrival, or the finish time of the last packet sent in flow i according to the guaranteed rate.

$$f_{i,n} = s_{i,n} + \frac{L_p}{\phi_i} \quad (4)$$

The finish time $f_{i,n}$ is computed from the start time plus the time it takes to send a packet of size L_p within the guaranteed delay.

The arrival time of an incoming packet $V(A_{i,n})$ is found using the virtual time. Virtual time advances according to the currently backlogged flows in the system as shown in Equation 5.

$$\frac{dV(t)}{dt} = \frac{C}{\sum_{j \in B(t)} r_j} \quad (5)$$

One key idea to note is that a flow that gives up its allocated transmission time because it has nothing to send, cannot later reclaim time from another flow that used its empty slot. In other words, a flow is not penalized for sending during another flows idle period, since that period would otherwise be wasted.

The results of the weighted fair queuing algorithm have several important properties. First the algorithm is work conserving and causal. Secondly, flows are separate from each other in that no one flow with an excess of packets can negatively affect the amount of traffic that another flow will be able to send. Finally, the algorithm provides both a bounded delay and an associated minimum throughput. These two guarantees are required for true support of QoS in any system.

4.2 Wireless Fair Queuing

A wireless fair queuing algorithm needs to slightly modify WFQ to compensate for channel error. Departing packets are swapped as needed and flows are allowed to either lag or lead their scheduled times. Lagging flows then make-up their lag by leading flows giving up their lead.

The crux of the WPS algorithm is this alteration to the weighted fair queuing algorithm to account for conditions specific to the wireless channel [13]. It modifies WFQ to account for errors in the channel that may otherwise prevent a flow from sending data during its scheduled time slot. Since location-dependent error is unique to wireless media, this scheduling is important in maintaining a fair algorithm over the error-prone channel.

The scheduling is done by the base station. It has knowledge of both the state of the channel and which mobile stations need to transmit data in the uplink direction. Once this information is known, the base station's scheduler can modify the original WFQ schedule. The next packet to send is chosen according to the minimum eligible finish time from the set of backlogged flows with good channels. Therefore, channels in error will hold back their transmission until they detect a clear channel. Since the virtual time does not change, these flows will have the opportunity to regain some of the lost transmission time while the channel was in error.

It should be noted that the backlogged flows are bounded to prevent a channel that is backlogged for a long period from exclusively grabbing the channel for a significant time upon perceiving a good channel. This prevents the starvation of some flows by another flow that had been waiting for a good channel. However, the other flows must wait while the backlogged flow clears its buffer. This is the same compensation scheme used in [12] which is simpler then the scheme in [13]

The worst case bounded delay for the HOL packet in the WFQ model is the time to send its own packet plus the time for all other flows to finish using their share of the bandwidth.

$$D_{WFQ}^{max} = \frac{L_p}{C} + \frac{L_p \cdot \sum_{j \in F} \phi_j}{\phi_i \cdot C} \quad (6)$$

In the original wireless case, the time to transmit a completely backlogged channel bounded by B bits was added.

$$D_{Wireless}^{max} = D_{WFQ}^{max} + \frac{B}{C} \quad (7)$$

4.3 Wireless Packet Scheduling

Details of the Wireless Packet Scheduling (WPS) functions are implemented in the scheduling protocol of the higher priorities. It operates by using the fair queuing weights to schedule the wireless channel access. These individual weightings are given to each connection attempting to use the channel by the higher layer applications. This layer is where the ATM QoS parameters are mapped into the desired weights for the algorithm.

Due to the inherent problems with wireless channels, there can be certain times allotted to a host when that host is unable to transmit data. These channel errors can occur from multipath fading, shadow fading, or interference from another device. The scheduling algorithm uses a one-step prediction to determine if a mobile host will be able to transmit during its next assigned time slot [12].

The protocol provides wireless compensation through a system of per-flow credits and debits. These credits and debits account for lagging and leading flows. If a flow is behind its ideal service due to wireless channel error, the flow is said to be lagging and thus has accrued credit. Conversely, if the flow is ahead of its ideal service, because of sending data when another flow had to wait, the flow is said to be leading and thus has accumulated debits. As scheduling progresses, flows with credit and debit are swapped to compensate for the wireless channel error. In effect, lagging flows can make up their lag by causing leading flows to give up their lead.

5 The Multiclass Priority Fair Queuing (MPFQ) Algorithm

The MPFQ algorithm is a priority scheduling algorithm for the ATM traffic classes. It uses the underlying WPS model at higher priorities, WRR scheduling for middle priorities, and FIFO scheduling at lower priorities.

5.1 MPFQ Priorities

The implementation of traffic guarantees relies on the determination of weights for weighted fair queuing and priorities for a multiclass scheduler as shown in Table 2. Determination of both parameters are needed to satisfy the requirements of the defined ATM traffic classes. A nonprioritizing fair queue scheduler alone is unsuitable for ATM traffic which was shown in [18].

Priority	Traffic Type	Parameters	Queuing Policy
1	CBR	PCR, CTD	WPS
2	rt-VBR	SCR, MBS, CTD	WPS
3	nrt-VBR, ABR_{MCR}	SCR, MBS, MCR	WRR
4	ABR	Unspecified	Recirc. Queue
5	UBR	Unspecified	FIFO

Table 2: ATM Prioritized Traffic

Real time traffic takes precedence over non-real-time traffic in the allocation of priorities. We have chosen to further divide CBR traffic from rt-VBR traffic. By placing CBR traffic into its own priority, it will be provided a lower delay bound then if it had to share weights with all CBR and rt-VBR flows. These priorities are the most delay sensitive and are the major factor in using a fair queuing algorithm.

Non-real-time traffic (nrt-VBR) has been assigned to the middle-priority level. This traffic tolerates little loss but can accommodate a greater delay. An algorithm such as fair queuing is not needed, since the delay is not tightly bounded. MPFQ uses a WRR scheduler with wireless compensation so that fairness is maintained. When a connection is admitted at this level, the scheduler will be able to guarantee that enough buffer space is available to handle the requested rate and burst size in order to avoid cell loss. Outgoing traffic may need to be delayed due to channel error, but instead of discarding the packet, the slot is just skipped for later transmission.

ABR traffic is given strictly higher priority than UBR connections. Note that ABR traffic is divided across two different priorities. This results from the fact that ABR traffic must meet the MCR guarantee. In order to meet this requirement, part of the ABR traffic (ABR_{MCR}) must have a sufficient priority to obtain guaranteed access to the channel. This traffic has been grouped with the nrt-VBR class at priority level 3 using the WRR scheduler. The traffic that arrives at rates above the MCR may use the remaining bandwidth only after all higher-priority traffic has been serviced. The queue is therefore serviced both when the MCR-portion is scheduled and when the channel is idle from higher priority traffic flows.

In ATM traffic, reordering of the cells violates the specifications for a virtual circuit. This is avoided by using the outgoing queue to hold only pointers to each flow's virtual circuit packet queue [19] and not the packets themselves. When a slot is recirculated, the pointer is sent to the back of the service queue. Once the channel is error-free, the earlier pointers will be used to transmit the HOL packets, thus avoiding reordering.

An example of the ABR dual priority service is shown in Figure 1 with one VBR flow and two ABR flows. Notice that the ABR flows can be serviced by either the level 3 scheduler or the level 4 scheduler. The outgoing queue holds pointers to the HOL packets in the respective queues. When it is time to schedule a packet for the output, the pointer is placed in the last slot. Both ABR flows will be serviced by the level 3 WRR scheduler in order to satisfy the MCR requirements. When packets are being scheduled, the ABR packets will be included in the WRR rounds to guarantee that the MCR is currently being met. For packets exceeding the MCR, the recirculating queue will be used to track which packet is to be sent next.

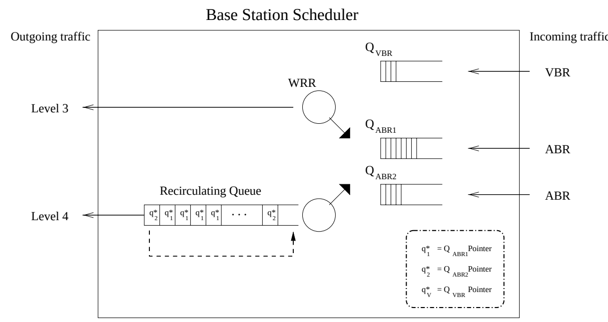

Figure 1: Cross Level Scheduler

The lowest priority level uses a simple FIFO for the scheduling. This includes the UBR traffic at level 5. Packets are placed in the FIFO as they are received. Since no guarantees are made on delay or bandwidth the scheduler can drop packets as needed. This dropping can occur on the outgoing path when a channel has location dependent error. It also can occur for incoming data if the buffer space is full and can no longer accept more packets.

To maximize utilization of the buffers, MPFQ uses a general buffer space for all flows in level 3 and below. Buffer space that is not being used at level 3 can then be marked and used by a lower level. If the level 3 traffic needs to reclaim its space, the lower level packets in the marked buffer memory will be dropped.

The advantages of the prioritized MPFQ scheme include the following:

1. CBR traffic has a minimum delay with highest precedence.
2. All real-time traffic will be serviced quickly with minimum possible buffering.
3. The algorithm is work conserving.
4. The separation of ABR and UBR traffic is maintained.
5. The MCR requirements of the ABR traffic are satisfied.

5.2 MPFQ mapping

To work within the context of the WFQ algorithms, the specific levels of ATM QoS need to be mapped into individual weights for use by the MAC. These weights are determined by the QoS algorithm, and depend on the type of traffic being used and the QoS parameters requested at connection setup.

The algorithm for mapping traffic classes into weights can be rephrased into three parameters for each traffic connection. Bandwidth must be reserved in order to satisfy the cell rates at call admission; the weights must be assigned for the fair queuing algorithm in order to satisfy delay; and buffers must be allocated in order to guarantee a flow will not lose data.

For CBR connections, the Peak Cell Rate (PCR) translates directly into a percentage of bandwidth. The maximum Cell Transfer Delay (maxCTD) can be used to find the necessary weight since it is inversely proportional to the desired weight ($\frac{1}{maxCTD} \sim \phi_{CBR}$). The MPFQ normalizes the weight ($\phi_n = \frac{\phi_{CBR}}{\sum_{j \in F} \phi_j}$) for the equality $\phi_n = \frac{1}{maxCTD}$. The amount of buffers needed for CBR are nonzero only because they are not perfectly synchronized in time. Thus a small buffer needs to be allocated.

For VBR connections, the bandwidth requirements are based on the Sustained Cell Rate (SCR). This is the average rate of the flow bounded by fixed size bursts. The maxCTD is also directly proportional to the desired weight. Again the weight is normalized ($\phi_n = \frac{\phi_{VBR}}{\sum_{j \in F} \phi_j}$) for the equality $\phi_n = \frac{1}{maxCTD}$. The bound on the buffer size can be determined from the Maximum Burst Size (MBS). On average the connection will have SCR traffic which is continually emptied by the WFQ algorithm. In the worst case, a new burst arrives at PCR while the flow is receiving its minimum service level. The buffer thus needs to be as large as $(MBS - \frac{MBS}{PCR} \cdot SCR)$.

The effect of the priority levels on the buffer size must also be taken into account. Since rt-VBR connections are lower priority than CBR connections, these flows must be able to buffer sufficient data for the higher level queues to drain. The buffer for rt-VBR must be able to store data while all the CBR queues ($\sum Q_{CBR}$) drain. The CBR data is being serviced at PCR and VBR must buffer for its own SCR. When considering the CBR class as an entire level, the service rate can be considered to be link speed (C). It is also known that the total CBR buffer space is equivalent to enough space for each CBR connections to buffer one packet. Let the total number of CBR connection be N_{CBR} for the following:

$$\sum Q_{CBR} = L_p * N_{CBR} \tag{8}$$

This leads to an additional buffer requirement for rt-VBR traffic producing a total requirement of the following:

$$Q_{rtVBR} = \left(MBS - \frac{MBS}{PCR} \cdot SCR\right) + \frac{L_p * N_{CBR}}{C} \cdot SCR \tag{9}$$

In the case of nrt-VBR, the connection must be able to store data while both the CBR traffic and the rt-VBR traffic transmit data and drain their queues. This amounts to an additional buffer size reservation for the $\sum Q_{rtVBR}$ service at each respective SCR. The total buffer space required for a nrt-VBR connection will be:

$$Q_{nrtVBR} = \left(MBS - \frac{MBS}{PCR} \cdot SCR\right) + \left(\frac{L_p * N_{CBR}}{C} + \frac{\sum Q_{rtVBR}}{\sum SCR_{rtVBR}}\right) \cdot SCR \tag{10}$$

Finally, ABR_{MCR} parallels a CBR connection. The strict delay bound does not constrain the traffic, but the MCR can be used to obtain the desired rate. As with CBR, the effective buffering needed at this level may be negligible. Due to the externally controlled nature of the ABR stream, little buffering should be needed. However, a buffer is needed in practice in order to accommodate cells that are already in transit when feedback is being sent to the source. This buffer must be able to hold enough data to support the MCR while the higher priority buffers are draining. ABR_{MCR} is scheduled in the same priority class as nrt-VBR so that these buffers do not need to be taken into account. Only the higher-priority buffers will affect the required buffering at this level. Therefore, ABR connections must be able to buffer the following:

$$Q_{ABR} = \left(\frac{L_p * N_{CBR}}{C} + \frac{\sum Q_{rtVBR}}{\sum SCR_{rtVBR}}\right) \cdot MCR \tag{11}$$

For the remaining ABR traffic (after MCR) there are no delay bounds and no throughput guarantees. The UBR best-effort traffic also has no hard guarantees. Since these two classes do not operate within the constraints of fair queuing no weights need to be calculated. Packets are allowed to be dropped, and buffering is handled through the use of higher-level marked buffers.

A summary of the parameter mappings are shown in Table 3.

5.3 MPFQ delay bounds

A vital part of the MPFQ algorithm is the analysis of the delay bounds for the priority levels and for the multiple classes themselves. Once these bounds are determined they can be used along with the individual connection weights both for call admission and for scheduling. The analysis is based on the delay bounds shown previously in Equation 6.

At the first priority level the traffic is scheduled according to the single-class fair queuing model. The level 1 (k=CBR) HOL bounded worst case delay will be:

$$D_{CBR}^{max} = \frac{L_p}{C} \cdot \left[1 + \frac{\sum_{j \in F_{CBR}} \phi_{CBR_j}}{\phi_{CBR_i}}\right] \tag{12}$$

Thus in the worst case a HOL packet in flow i will be delayed while it waits for every other flow to send its share of the channel bandwidth, plus the time to send its own packet. The capacity is $C = C_{CBR}$ since the entire bandwidth is available.

The second priority level again uses the single-class fair queuing model. However, the additional delay due to higher-priority traffic must be considered. Traffic must wait for the higher-level queues to completely drain before gaining channel access. The delay must also account for the arriving traffic at the higher level over the time period in question. This leads to an effective reduced rate at this level. Once the higher level traffic has been serviced, the rt-VBR traffic might still be delayed until it gets its fair share of the channel. It then will need the time to send out its own packet. This produces the delay shown below:

$$D_2^{max} = \frac{\frac{\sum_{j \in F_1} Q_{1_j}}{C} + \frac{L_p}{C} \cdot \left(1 + \frac{\sum_{j \in F_2} \phi_{2_j}}{\phi_{2_i}}\right)}{\left(1 - \frac{\sum_{j \in F_1} \lambda_{1_j}}{C}\right)} \tag{13}$$

This means level 2 traffic has an effective throughput of:

$$C_2 = C - \sum_{j \in F_1} \lambda_{1_j} \tag{14}$$

The HOL packet in level 2, flow i must wait until the level 1 queue has drained plus all arriving level 1 packets have been serviced over the delay time. The packet

Traffic	Rate	Weight (ϕ_n)	Buffer Size
CBR	PCR	$\frac{1}{maxCTD}$	1
rt-VBR	SCR	$\frac{1}{maxCTD}$	$\left(MBS - \frac{MBS}{PCR} \cdot SCR\right) + \frac{L_p * N_{CBR}}{C} \cdot SCR$
nrt-VBR	SCR	N/A	$\left(MBS - \frac{MBS}{PCR} \cdot SCR\right) + \left(\frac{L_p * N_{CBR}}{C} + \sum Q_{rtVBR} / \sum SCR_{rtVBR}\right) \cdot SCR$
ABR	MCR	N/A	$\left(\frac{L_p * N_{CBR}}{C} + \sum Q_{rtVBR} / \sum SCR_{rtVBR}\right) \cdot MCR$
UBR	N/A	N/A	N/A

Table 3: ATM Traffic Mappings

might then have to wait for all other flows in level 2 to get their share of the bandwidth before gaining access and sending the flow i packet. In terms of the rt-VBR traffic the arrival rates of CBR traffic and the buffered data queue lengths must be known. The CBR arrival rates are defined by the PCR, and each CBR queue length is one. Again defining the total number of CBR connection as N_{CBR} we have $(\sum_{j \in F_{CBR}} Q_{CBR_j} = L_p \cdot N_{CBR})$. For notation let PCR_i be the PCR of the flow i CBR traffic. With the effective throughput from Equation 14 we have $C_{rtVBR} = C - \sum_{j \in F_{CBR}} \lambda_{CBR_j} = C - \sum_{j \in F_{CBR}} PCR_j$ providing a bounded delay as follows:

$$D^{max}_{rtVBR} = \frac{\frac{(N_{CBR} \cdot L_p)}{C} + \frac{L_p}{C} \cdot \left(1 + \frac{\sum_{j \in F_{rtVBR}} \phi_{rtVBR_j}}{\phi_{rtVBR_i}}\right)}{\left(1 - \frac{\sum_{j \in F_{CBR}} PCR_j}{C}\right)} \quad (15)$$

This can be simplified to:

$$D^{max}_{rtVBR} = \frac{L_p}{C_{rtVBR}} \cdot \left(1 + N_{CBR} + \frac{\sum_{j \in F_{rtVBR}} \phi_{rtVBR_j}}{\phi_{rtVBR_i}}\right) \quad (16)$$

For the level 3 HOL packet, flow i must wait for both the level 1 and level 2 queues to drain before being serviced. However, since this traffic does not have a specified QoS delay parameter, there is no need to calculate a delay bound. The throughput guarantees for this traffic (nrt-VBR and MCR-ABR) are provided by the admission control. The delay bounds can be very large relative to the size of the buffers.

ABR and UBR levels have no delay restrictions. Delay bounds therefore are unnecessary. These levels are serviced by simpler scheduling methods.

5.4 Inclusion of Channel Error

The MPFQ parameters specified for the ideal weighted fair queuing case can now be applied to the wireless model. Since the MAC layer will use the system of credits/debits to swap flows to compensate for local errors, the delay bound for our prioritized classes can exceed the maximum delays calculated in Section 5.3.

For CBR traffic we use Equation 7 and 12 to find the bounded wireless delay.

$$D^{max}_{wireless_{CBR}} = \frac{L_p}{C} \cdot \left[1 + \frac{\sum_{j \in F_{CBR}} \phi_{CBR_j}}{\phi_{CBR_i}}\right] + \frac{B}{C} \quad (17)$$

The rt-VBR traffic delay can be bounded similarly using Equation 7 and 16. Here it should be noted that the use of the wireless protocol does not change the time a flow is delayed due to a higher level. All flows at the higher priorities must be drained prior to the lower-level gaining channel access. This renders the higher priority swapping inconsequential to the lower-priority delay. However, at the level in question the swapping is important and will cause the overall worst-case delay to degrade. This will produce a maximum worst-case delay for rt-VBR traffic that will be larger when using the wireless protocol. This bounded delay is as follows:

$$D^{max}_{wireless_{rtVBR}} = \frac{B}{C} + \frac{L_p}{C_{rtVBR}} \cdot \left(1 + N_{CBR} + \frac{\sum_{j \in F_{rtVBR}} \phi_{rtVBR_j}}{\phi_{rtVBR_i}}\right) \quad (18)$$

6 Simulation Results

Simulations were performed on various flow scenarios to test the MPFQ algorithm. These results were then analyzed for further insight into the MPFQ scheduling algorithm. To illustrate the core functionality of the algorithm three flows including a CBR, a rt-VBR, and a nrt-VBR flow, were generated. The requested rates were chosen so as to stress the scheduler into a congestive state. The flows are fully representative of a fully loaded system with a multiple number of flows of each class combined into an aggregate rate for that traffic type.

The simulator consists of an event queue that generates traffic and processes packets. The traffic for the CBR flows is generated at a constant rate. The bursty VBR traffic is generated and run through a traffic shaper to guarantee that the SCR, PCR, and MBS bounds are not violated. This shaper is a queue with two leaky buckets holding tokens, where a token from both buckets is necessary to allow traffic through the shaper. The first bucket of size 1 gets tokens at PCR and the second bucket of size MBS gets tokens at SCR. Shaping of the VBR traffic is necessary for the fair queuing derived bounds to be valid.

The VBR traffic is generated by using a two state Markov model. Traffic is statistically generated in state 0 at a reduced cell rate and in state 1 at PCR. At each cell period state 0 or state 1 transitions to the opposing state with probability q or p respectively. These values are determined by knowing that $(P[0] * PCR) + (P[1] * ReducedRate) = SCR$ where $P[1] = \frac{1}{1+q/p}$ is the probability of being in state 1, and $P[0] = 1 - P[1]$ is the probability of being in state 0. The expected time in state 1 is also known to be $E[1] = 1/q = MBS/PCR$. The values of p and q can then be calculated such that the Markov model generates bursty VBR traffic at PCR with an average rate of SCR.

6.1 Simulation of Scheduler without Channel Error

The simulator was first used to verify the derived bounds on delay and buffer size. The characteristics of the generated traffic flows are summarized in Table 4:

Flow	CBR	rtVBR	nrtVBR
PCR	0.20	0.95	0.40
SCR	0.20	0.40	0.25
MBS	1	200	400
Level	0	1	2
Measured Queue Size	1	58	142
Calculated Queue Size	1	115	225
Measured Delay	1	72	514
Calculated Delay	2	147	N/A

Table 4: MPFQ Simulation Data

The three flows have a sustained link utilization of 85%. The rates have been normalized to a link capacity of 1 and delays are given in terms of packet/cell times. The measured values were taken from simulation data and the calculated values were determined using the equations in Table 3 for queue sizes and Equations 12 and 16 for maximum HOL delays. The calculated delay was then found by taking the HOL packet worst-case delay with all other packets served at SCR. As shown in Table 4, simulated results fall within the calculated bounds.

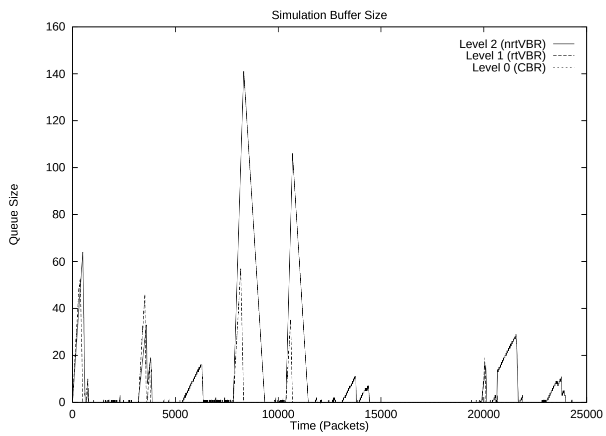

Figure 2: Accumulated Queue Size per Level

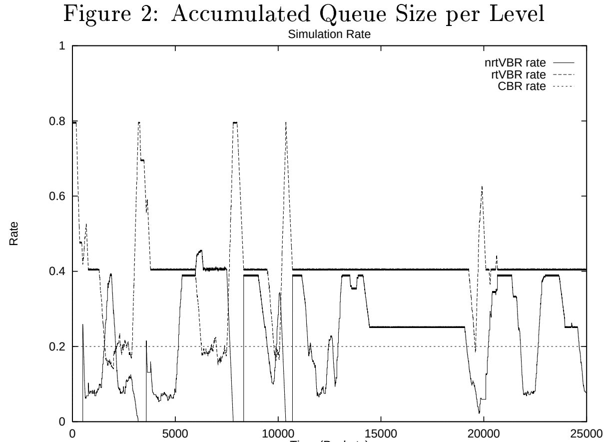

Figure 3: Rate per Flow

The graphs of the simulations are presented to show how the multiclass flows interact. Figure 2 shows the accumulated queue sizes (in cells) for each prioritized level as time progresses. Time in this graph is given in terms of cells processed by the scheduler.

Figure 3 shows the processed rate of traffic for each of the traffic classes. The rate is determined as a moving average over a small window of instantaneous rate samples at the scheduler output. Note that both of the figures are on the same time scale for easy comparison, and that as the rate of a flow bursts or subsides the appropriate queue size can be seen to grow or drain respectively. The flatlines on Figure 3 are a result of VBR bursting traffic being limited to SCR by the traffic shaper. This occurs when it was generated at its PCR for more than MBS. The aggregate rate of each flow can be seen to average out to the SCR. Also notice that the CBR flow effectively reduces the link capacity for lower-priority flows, limiting their peak burst rates to 0.8 or 80% of the link rate.

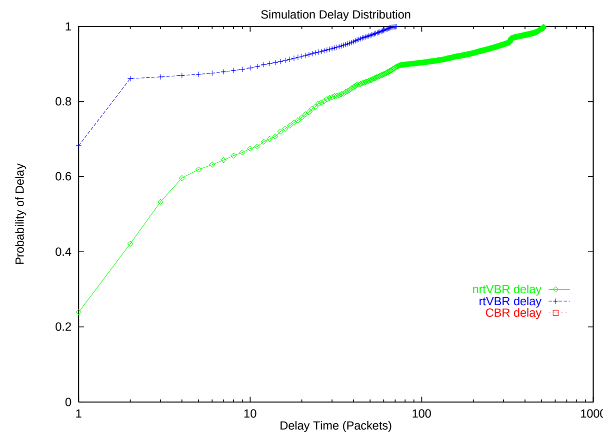

Figure 4: Cumulative Distribution of Delay

The cell delays are presented in Figure 4 as a cumulative distribution. Each flow has also been normalized into percentages for comparison across traffic classes. Notice that all of the CBR traffic has a delay of 1 or less. Approximately 70% of rt-VBR traffic and 25% of nrt-VBR traffic also have a delay of 1 or less. All rt-VBR traffic has a bounded delay. Once this delay has been exceeded, the nrt-VBR traffic that had been waiting on the higher-level traffic, can quickly drain. The maximum delay values correspond to the values in Table 4.

6.2 Simulation of Scheduler with Channel Error

The simulator was also used to analyze the operation of the algorithm under varying wireless error conditions. A two state Markov model was used for the wireless channel error. The first state signifies the clean channel condition and the second state signifies the errored channel condition. Two parameters were needed to determine the state transition probabilities of the error model. The first parameter is the percentage of time that a flow will experience errors. The second parameter was the expected time that the model should stay in each state.

The traffic flows used in these simulations were the same scenario as in Table 4. The two error model parameters were varied to determine the effect of wireless error on the algorithm. All of the simulation runs used the exact same traffic generation pattern.

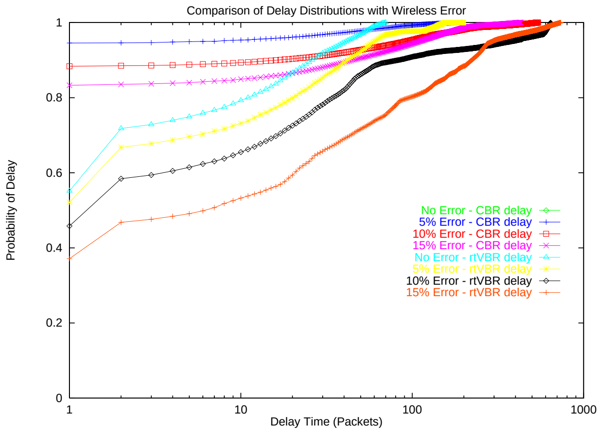

Figure 5: Effect of Wireless Channel Error on Delay

In Figure 5 the effects of increasing channel error are shown on the flow delay distributions. The curves can be seen to shift down as the percentage of time the channel is in error increases. The increasing error also extends the worst case delays as expected. This is especially true when the channel goes from no error to just a slight error. The change is less significant as the error continues to increase. Also, as the channel error increases the worst case delays from the different levels start to converge. This occurs since the delay becomes more highly dependent on the wireless error and less on the scheduling between levels. Most importantly, it can be seen from the graph that the overall distributions do not significantly change. This bodes well for the algorithm since most packet delay is still the same or just slightly more than in the error free case.

7 Conclusion

The Multiclass Priority Fair Queuing (MPFQ) algorithm is a novel multi-level scheduler that supports QoS over a packetized wireless channel for all types of ATM traffic. The scheduler maps QoS parameters specified at call setup into priorities and weights for the packetized wireless channel. Rather than redefining QoS parameters for the wireless channel, the model uses and enforces the wired standards across the combined network.

ATM QoS classes are mapped into multiple priorities to provide service for real-time CBR, rt-VBR, nrt-VBR, ABR, and UBR traffic types. Each level is then scheduled to meet its bandwidth and delay guarantees. Real-time traffic uses a wireless packet scheduling algorithm for fair and bounded channel access. Non-real-time data uses a WRR scheduler and a recirculating service queue to guarantee throughput in the presence of errors. Unspecified traffic uses simple FIFO processing of service queues. ABR traffic flows span priorities in order to provide both minimum cell rates and best-effort service.

Simulation results demonstrated the MPFQ algorithm. Derived delay bounds and buffer sizes were verified and plots of queue size, rate, and delay were presented. The results from the analysis and simulation show that the MPFQ algorithm can be used to bound real-time traffic and fairly share all ATM traffic over a wireless link.

References

[1] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, Aug. 1995.

[2] G. R. J. Linnenbank, P. Venkataram, P. J. M. Havinga, S. J. Mullender, and G. J. M. Smit, "A request-TDMA multiple-access scheme for wireless multimedia networks," in *MoMuC-3*, (Princeton,New Jersey), Sept. 1996.

[3] S. Choi and K. G. Shin, "Uplink CDMA systems with diverse QoS guarantees for heterogeneous traffic," in *MOBICOM '97*, (Budapest,Hungary), pp. 120–130, Sept. 1997.

[4] J. G. Kim and M. Krunz, "Effective bandwidth in wireless ATM networks," in *MOBICOM '98*, (Dallas,Texas), pp. 233–241, Oct. 1998.

[5] A. Acampora and M. Naghishineh, "Control and quality of service provisioning in high-speed microcellular networks," *IEEE Personal Communications*, vol. 1, no. 2, 1994.

[6] J. M. Capone and I. Stavrankakis, "Achievable QoS in a shared wireless channel," in *HPCS '97*, (Chalkidiki,Greece), June 1997.

[7] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *SIGCOMM '89*, (Austin,Texas), pp. 1–12, Sept. 1989.

[8] J. C. R. Bennett and H. Zhang, "WF^2Q: Worst-case fair weighted fair queueing," in *INFOCOM '96*, (San Franciso,California), Mar. 1996.

[9] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *INFOCOM '94*, June 1994.

[10] P. Goyal, H. Vin, and H. Cheng, "Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks," in *SIGCOMM '96*, (Stanford,California), pp. 157–168, Aug. 1996.

[11] J. C. R. Bennett and H. Zhang, "Heirarchical packet fair queueing algorithms," in *SIGCOMM '96*, (Stanford,California), pp. 143–156, Aug. 1996.

[12] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," in *SIGCOMM '97*, (Cannes, France), 1997.

[13] S. Lu, T. Nandagopal, and V. Bharghavan, "A wireless fair service algorithm for packet cellular networks," in *MOBICOM '98*, (Dallas,Texas), pp. 10–20, Oct. 1998.

[14] P. Ramanathan and P. Agrawal, "Adapting packet fair queueing algorithms to wireless networks," in *MOBICOM '98*, (Dallas,Texas), pp. 1–9, Oct. 1998.

[15] "Traffic Management Specification v4.0." ATM Forum Document AF-TM-0056.000, Apr. 1996.

[16] J. Moorman and J. Lockwood, "Implementation of the multiclass priority fair queuing (MPFQ) algorithm for extending quality of service in existing backbones to wireless endpoints." Submitted to Globecom '99.

[17] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–347, June 1993.

[18] J. R. Moorman, "Supporting quality of service on a wireless channel for ATM wireless networks." MS Thesis, University of Illinois, Jan. 1999.

[19] H. Duan, J. W. Lockwood, S. M. Kang, and J. Will, "High-performance OC-12/OC-48 queue design prototype for input-buffered ATM switches," in *Infocom '97*, (Kobe, Japan), pp. 20–28, Apr. 1997.