Token bank fair queuing: a new scheduling algorithm for wireless multimedia services

William K. Wong1,*,†, Helen Y. Tang1,‡,§ and Victor C. M. Leung2,†

1 Communication Research Centre Canada (CRC), 3701 Carling Ave., Box 11490, Station H, Ottawa, Ont., Canada K2H 8S2
2 Department of Engineering and Computer Engineering, University of British Columbia, Vancouver, BC, Canada

SUMMARY
The token bank fair queuing algorithm (TBFQ) is a novel scheduling algorithm that is suitable for wireless multimedia services. The bandwidth allocation mechanism integrates the leaky bucket structure with priority handling to address the problem of providing quality-of-service (QoS) guarantees to heterogeneous applications in the next generation packet-switched wireless networks. Scheduling algorithms are often tightly integrated with the wireless medium access control (MAC) protocol. However, when heterogeneous wireless systems need to be integrated and interoperate with each other, it is desirable from the QoS provisioning standpoint to decouple scheduling algorithm from the MAC protocol. In this paper we propose a framework of seamless QoS provisioning and the application of TBFQ for uplink and downlink scheduling in wireless networks. We study its performance under a generic medium access framework that enables the algorithm to be generalized to provide QoS guarantees under various medium access schemes. We give a brief analysis of the algorithm and compare its performance with common scheduling algorithms through simulation. Our results demonstrate that TBFQ significantly increases wireless channel utilization while maintaining the same QoS, unlike many fair queuing algorithms, TBFQ does not require time-stamping information of each packet arrival—an impractical feature in an already resource scarce environment. This makes TBFQ suitable for wireless multimedia communication. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: wireless scheduling; fair queuing; QoS; heterogeneous network; call admission control; medium access control protocol; interoperability

1. INTRODUCTION
In response to the growing diversification of wireless/mobile systems, there is a need to efficiently integrate different wireless systems to enable interoperability and maximize wireless

*Correspondence to: William K. Wong, Communication Research Centre Canada (CRC), 3701 Carling Avenue, Box 11490, Station H, Ottawa, Ont., Canada K2H 8S2.
† E-mail: william.wong@crc.ca
‡ Currently a PhD student at the Department of System and Computer Engineering, Carleton University, Ottawa, Ont., Canada.
§ E-mail: yhtang@sce.carleton.ca
† E-mail: vleung@ece.ubc.ca

Received 15 October 2003
Revised 15 February 2004
Accepted 30 April 2004

Copyright © 2004 John Wiley & Sons, Ltd.
utilization. In addition, wireless systems will see growing use of multimedia applications (heterogeneous traffic types). To allow heterogeneous traffic roaming across different wireless systems requires sophisticated underlying QoS provisioning mechanisms. For example, the roaming of multimedia applications between third-generation cellular networks and wireless local area networks poses challenges in maintaining QoS for handoff terminals.

In next generation wireless/mobile packet-switched networks, MAC protocols play a crucial role in QoS guarantees. Specifically, the scheduling algorithm in MAC protocols plays an important role in QoS guarantees as future applications increasingly demand more QoS support. Wireless MAC protocols have been studied extensively since the 1970s [1]. Recently, we have seen the growing importance of underlying scheduling algorithms [2], as more sophisticated multimedia applications are being developed and making use of the wireless broadband networks. The issues that separates wireless scheduling and wireline are mainly wireless link variability, which is caused by interference, fading, shadowing in the wireless channels (time-dependent problems), and location-dependent problems. Power constraint in the wireless terminals (WT) is also a significant issue in wireless networks. Algorithms should be kept simple in wireless terminals. In order to do this, less control or signaling information sent across the wireless channel is more desirable. Channel state dependent packet scheduling CSDPS [3] addresses location-dependent and bursty errors in wireless scheduling, but it does not address the issues of delay guarantee, fairness, and throughput. It gives credits to mobiles in bursty errors rather than scheduling bandwidth for them. Idealized wireless fair queuing (IWFQ) [4] is proposed for packet scheduling in cell-structured wireless networks. It is defined with reference to an error-free weighted fair queuing (WFQ) service system [5], therefore it is not practical to implement. When it is replaced by weight round robin (WRR) instead of WFQ, the worst-case performance of the real implementation is much worse than that of IWFQ. Another limitation is that when a flow is compensated for its previous lagged service all other error-free flows will not be served at all. For the same reason, a lagging flow will receive compensation at a rate independent of its allocated service rate; this violates the semantic that a larger guaranteed rate implies better QoS. Channel-condition independent packet fair queuing (CIF-Q) [6] is very similar to IWFQ in that it is an approximation of the ideal error-free systems, and it also defines a flow as being leading, lagging, or satisfied at any time instant if it receives more, less or the same amount of service as it would have received in the corresponding error-free system. CIF-Q mainly addresses the fairness issue (both long- and short-term), delay and throughput guarantees, and graceful degradation for leading flows. However, achieving fairness and fast convergence to throughput under erroneous conditions is mutually exclusive [7].

In a MAC protocol, channel access and scheduling algorithms both affect QoS performance. Channel access contention is an undesirable effect in applications where QoS requirements are critical. Thus, the role of a packet-scheduling algorithm becomes more important. To understand the effect of scheduling algorithms on the overall QoS performance, it is desirable to decouple its effect from that of channel access as much as possible as it gives clarity in understanding its contribution. Hence, we choose to use contention-free access and emphasize bandwidth reservation.

The function of MAC is to moderate access to the shared medium by defining rules that allow these devices to communicate with each other in an orderly, efficient and fair sharing of the scarce wireless bandwidth. The function of scheduling algorithms is to determine the amount of bandwidth allocation to all traffic while providing QoS according to the various losses, delays and bandwidth requirements. Although the two functions are often integrated to provide QoS,
it is important to understand the difference between their roles when designing a QoS provision scheme. If a scheduling algorithm is designed for a specific MAC protocol, it is unlikely that it can be exported easily to another MAC system and serve efficiently. Therefore different scheduling algorithms would be needed for different MAC protocols in heterogeneous wireless systems. Different wireless systems often come with its specific MAC protocol [8–23], but when a mobile terminal roams heterogeneous networks, seamless QoS should be maintained as much as possible. It is therefore desirable to have one bandwidth allocation algorithm that could easily be deployed and interoperate with various MAC protocols. Such bandwidth allocation algorithm should also be capable of interoperating with other schedulers within the wireline network. Under our framework, we envision the decoupling of MAC and scheduling algorithm to focus on the design of a scheduling algorithm that anticipates interoperability with other MAC protocols in the future. One benefit of this framework is that QoS can be provided seamlessly across different wireless systems.

The first contribution of this paper is to propose an effective scheduling and call admission control algorithm for providing seamless QoS in heterogeneous wireless networks. Secondly, we propose a framework under which scheduling algorithms can operate seamlessly across multi-access multi-service networks. We study the behaviour of our algorithm through a combination of theoretical analysis and discrete-event simulation in both wireless uplink and downlink channels using heterogeneous traffic types. The QoS parameters under investigation include link utilization, delay, throughput, and fairness. We introduce the definition of graceful degradation as it is an important parameter often omitted in wireless QoS studies.

The paper is organized as follows. Section 2 provides a brief overview of wireless scheduling algorithms and outlines the desired properties of wireless scheduling algorithms, which help to understand the significance of the TBFQ algorithm. In Section 3, we propose a network architecture and framework for seamless QoS provisioning, and present traffic models used for our evaluation. Section 4 proposes the CAC and scheduling algorithm, and describes how the algorithm can be used in various MAC schemes. Section 5 provides preliminary analysis of the algorithm. Section 6 presents the simulation environment and results. Finally, conclusions are made in Section 7.

2. DESIRED PROPERTIES OF WIRELESS SCHEDULING ALGORITHMS

Traditionally, MAC protocols represent a key part of every wireless systems employing statistical multiplexing of real-time and non-real-time data traffic over the uplink and determine the QoS. However, to provide efficient bandwidth allocation in a packet-switched network with appropriate QoS support for diverse multimedia applications, a robust scheduling algorithm is required. To compare various scheduling algorithms, we introduce some of the desired properties of a scheduling algorithm.

(1) Efficient bandwidth utilization: One of the fundamental reasons for a scheduler is to exploit statistical multiplexing while providing QoS guarantees. This is very important in wireless networks, as bandwidth is scarce. The scheduler should use statistical multiplexing to the fullest possible and provide maximum number of traffic streams with QoS guarantees.
Bounded delay: Multimedia applications are usually delay sensitive. This suggests that the scheduler should have a limit on the delay of packet transmission. A delay bound can be achieved through the use of connection admission control, and a traffic-shaping device such as a leaky bucket.

Fairness/isolation: The algorithm should reimburse system resources that are not used by idle sessions to the backlogged sessions in a fair manner. In other words, the amount of available bandwidth distributed to backlogged sessions should be proportional to their contracted bandwidth. At the same time, the algorithm should isolate well-behaved traffic from ‘malicious’ traffic, so that if a malicious traffic is drawing more bandwidth than it originally intended, the QoS of other well-behaving traffic should not be degraded.

Low complexity: The complexity in calculation may affect the performance of an algorithm in practice. Algorithms should be kept simple yet efficient. The amount of sorting and searching should be limited. Ideally, $O(1)$ complexity is desirable.

Heterogeneous traffic support: The algorithm should be able to multiplex diverse classes of traffic and provide service differentiation within the same class.

Graceful service degradation: QoS performance can degrade for many reasons; bursty channel errors, location-dependent channel capacity and errors [24], unanticipated traffic behaviour (bursty traffic) to name a few. Although sometimes it is unavoidable to interrupt QoS abruptly, a scheduler should allow QoS performance to degrade gracefully as much as possible. As graceful degradation is subjective, we attempt to quantify the degree of graceful degradation as the following:

$$DGD_\Phi = \frac{\Phi[t_0, t] - \Phi_0[t_0, t]}{\Phi_0[t_0, t]}$$

where

$$\Phi[t_0, t_n] = \sum_n (\Phi(t_n)),$$

for $n = 0, 1, 2, \ldots$, and $\Phi[t_0, t_1] = \Phi[t_0, t_2]$ for $t_0 < t_1 < t_2$

$\Phi$ is a QoS measure, for example, throughput or violation probability. The degree of graceful degradation in a QoS measure is the relative difference between the QoS measure under normal or ideal condition ($\Phi_0$) and the QoS measure under non-ideal condition ($\Phi'$) during the period $[t', t_0]$ that abnormality occurred.

Connection admission control (CAC): In both References [24, 25], the authors pointed out that different service disciplines must be accompanied by different connection admission control algorithms. The CAC must have the knowledge of the underlying scheduling algorithms in order to make wise decision as to whether to admit more sessions. Admitting too many sessions may cause the QoS of existing sessions to degrade, while admitting too few may lose multiplexing gain and bandwidth utilization. Therefore, a service discipline should facilitate the use of a CAC.

Interoperability: Roaming of services across different wireless technologies will pose a challenge when integrating heterogeneous wireless systems. It is desirable for a multimedia service algorithm to interoperate with other wireless systems for the purpose of integrating heterogeneous wireless systems. The algorithm should also allow easy translation of QoS requirements from different systems and be able to provide QoS to transiting mobile terminals.
3. NETWORK MODELS

In this section, we consider the network architecture, traffic and error model in our studies.

3.1. Network architecture

The amalgamation of various networks is a challenge in next generation networks beyond the third-generation (B3G). A mobile terminal that crosses from one network to another must be equipped with transceivers compatible with the systems it traverses; each system also has its physical and MAC specifications. Carrying system-specific hardware for all systems is unavoidable. However, QoS provisioning can be supported uniformly across various systems without burdening the mobile terminals. In order to achieve this, a few provisions will have to be in place. We propose an architecture that is necessary for seamless QoS across different networks. In the next section, we present the scheduling and CAC algorithms that complement this framework, although the algorithms can work independently.

Let us consider a mobile terminal traversing from a WLAN (802.11 based) to an UMTS network where QoS is not to be interrupted. The same scheduling algorithm should be used across different networks. To this end, it is necessary to invoke remotely the QoS service components from a central location that contains all the service components such as scheduling and CAC (Figure 1). We assume dedicated high-speed connections are established between the services and the mobile access nodes. The de-coupling of scheduling algorithm from physical and data-link specifics not only allows the access of the same scheduling algorithm by different

![Figure 1. Network architecture of multi-access multi-service networks.](image-url)
systems, it also permits ease of future enhancement of services. The framework coupled with the ability of the scheduling algorithm to work with various MAC is vital if seamless QoS is to be provided. One may argue that it is conceivable to have different systems maintaining their own scheduling algorithms. In regard to seamless QoS provisioning, QoS mappings would be necessary for all possible combination of interoperable systems, however mappings may not always be possible.

3.2. Traffic models

For voice traffic model, we assume speech codec rate of 32 kb/s with voice activity detector [15]. It follows a pattern of talk-spurt and silent gaps, represented by two-state discrete Markov chain (ON–OFF) model. The mean talk duration is 1.0 s and the mean silence duration is 1.35 s.

Our video traffic model is based on the work done in Reference [26] which was derived from a real videoconference data stream conforming to the H.263 standard. We use the real stream as well, because we believe this would provide more realistic results. The video has an encoding format of 352 × 288, and mean rate of 77.63 kb/s. The minimum and maximum frame sizes are 1615 and 66416 bits, respectively. The frame rate is 15 frames per second.

For data traffic model, we use the model provided in Reference [27]. Packet call size is modelled as truncated Pareto \( A = 1.1, k = 4.5 \text{ kbytes}, m = 2 \text{ Mbytes}, \mu = 25 \text{ kbytes} \). Time between packet calls is geometric with \( \mu = 5 \) s. The number of packets and packet size in our case is determined by the MAC data slot size.

3.3. Channel error models

We have enhanced the static error model used in Reference [7] to a two-state Markov model (from Reference [27]) to emulate the process of packet transmission errors. The channel varies between a ‘good’ state and a ‘bad’ state, \( s_0 \) and \( s_1 \), respectively, for each packet transmission. During \( s_0 \), packets are transmitted error free, and errors occur during \( s_1 \). The probability of remaining in a good state is 0.328. The probability of transiting from good to bad state is 0.0000235 and 0.46945 for bad to good state.

4. PROPOSED BANDWIDTH ALLOCATION ALGORITHM

4.1. Connection admission control

The evaluation of performance of any broadband system that carries multiplexed streams of different traffic classes is meaningless without the use of a CAC algorithm [13]. The objective here is clearly very simple: given a call arriving, requiring a virtual connection with specified QoS (such as bandwidth, loss, probability, delays), should it be admitted? However, the implementation of the control can be quite complex. In a real network control, messages would have to be sent along the end-to-end path of a connection that would have to provide this connection to ascertain whether QoS objectives could be met without adversely affecting other calls already in progress. For homogeneous on-off sources, Guerin et al. [28] offered the equivalent bandwidth CAC algorithm formula. In a multi-access multi-service network scenario, the CAC problem becomes very complex and is a topic of research in itself. To reduce the complexity of our CAC algorithm, we make the following assumption: the forecast of
handoff and mobility model of wireless terminals is not considered. The requesting connections 
would come either from the same cell or handoff from another system, and once admitted they 
do not move beyond the coverage of the cell. We developed an algorithm similar to the one 
proposed in Reference [29]. Each connection \( i \) must provide both the desired bandwidth \( (B_{i,d}) \) 
and the minimum bandwidth \( (B_{i,m}) \) to the CAC. The desired bandwidth is between the 
average and maximum rate of the connection which is depended on the type of traffic. The minimum 
bandwidth is the minimum required by the connection for maintaining acceptable quality. The 
CAC first checks the total resources, \( B \), which is comprised of \( B_{\text{H}} \) (resources reserved for 
handoff connections) and \( B_{\text{N}} \) (resources for new connections). For new connections, the CAC 
tries to allocate \( B_{i,d} \) to the connection if possible, and if that is not possible it will try 
to allocate \( B_{i,m} \). If the requested bandwidth is larger than \( B_{i,m} \) the bandwidth compensation 
algorithm is invoked. Our bandwidth compensation algorithm is designed to work closely 
with TBFQ. The algorithm attempts to redistribute connections with bandwidth greater than 
\( B_{i,m} \). If the compensation algorithm fails to find additional bandwidth, the connection is 
rejected. For handoff terminals requesting access, \( B_{\text{H}} \) is used instead. However if handoff and 
mobility models were considered, sharing of \( B_{\text{H}} \) and \( B_{\text{N}} \) may be provided. The algorithm is 
shown in Figure 2.

4.2. Proposed scheduling algorithm

TBFQ was first used for wireless packet scheduling in the downlink channel [26] and it is 
modified to handle uplink channel scheduling as well. It integrates the policing and scheduling 
functions. Its predecessors, the LB mechanism and its many multi-level variants, stringently 
police the negotiated parameters for individual connection. Such restriction degrades the 
statistical multiplexing of group connections. TBFQ penalizes violating traffic less severely as it 
is able to service a packet, which might otherwise be discarded by the per-flow policing 
mechanism, by distributing unused bandwidth from other connections. TBFQ exploits the 
statistical multiplexing of group connections to enhance bandwidth utilization—an important 
factor in wireless links. Used in conjunction with the CAC, TBFQ guarantees the minimum rate 
of a connection in an error-free environment. To compensate for bandwidth, CAC interacts 
with TBFQ through the manipulation of token generation rates.

Referring to Figure 3 for the structure of TBFQ, a WT within a service group \( x \) can be 
represented as \( \text{WT}^x_i, i = 0, 1, \ldots, n_x; \) \( a : \{ a = \text{voice}, v = \text{video} \} \) where \( n_x \) is the total number 
of WTs in a service group \( x \). Each \( \text{WT}^x_i \) has a LB associated with it and characterized as \( \text{LB}^x_i = 
(\rho'_x, P'_x, D'_x); i = 0, 1, \ldots, n_x; \) \( x : \{ a = \text{voice}, v = \text{video} \} \) where \( r \) is the token generation rate 
(bytes/s), \( P \) is the token buffer size (bytes), and \( D \) is the data buffer size (bytes). For uplink 
traffic, the policing function is implemented in the WTs, and for downlink traffic the policing 
function is implemented in the BS.

It is assumed that each connection has a sufficiently large input buffer, and each token pool, 
\( P \), holds tokens for one packet only (assuming fixed size packets), so this scheme is suitable to 
air links with constant packet (or block) size (e.g. GPRS). A user’s (voice/video) connection QoS 
is determined in part by these parameters and may be predetermined by the operator for each 
supported service class. Since we are dealing with scheduling only, we assume CAC has taken 
place and consider only admitted connections in the system. The BS uses the traffic contract of 
each connection to reserve a fixed amount of bandwidth initially to the connection to satisfy the 
average data rate of the connection.
Connection admission control algorithm

// There are \( n_0 \) existing connections, and connection \((n+1)_0\) is requesting admission.
// \( n_0 \) connections are sharing bandwidth \( B_0 \) within a cell.
// There are \( n_0 \) existing handoff connections, and connection \((n+1)_0\) is requesting admission.
// \( n_0 \) connections are sharing bandwidth \( B_1 \) within a cell.
// This algorithm calculates whether there is enough resources for connection \((n+1)\).

\{
    CASE: SWITCH connection \((n+1)\):
        CASE NEW Connection:
            IF Available\( B_0 \) \(\geq\) \( B_0_{(n+1)} \) THEN
                Grant\( (B_0_{(n+1)} \) to connection \((n+1)_0\).
            ELSE IF Available\( B_0 \) \(\geq\) \( B_0_{(n+1)} \) THEN
                Grant\( (B_0_{(n+1)} \) to connection \((n+1)_0\).
            ELSE
                Compensate\( B_{n+1_0}, NEW \_Connection, B_0 \)
            END;
    CASE: NEW Handoff:
        IF Available\( B_1 \) \(\geq\) \( B_1_{(n+1)} \) THEN
            Grant\( (B_1_{(n+1)} \) to connection \((n+1)_0\).
        ELSE IF Available\( B_1 \) \(\geq\) \( B_1_{(n+1)} \) THEN
            Grant\( (B_1_{(n+1)} \) to connection \((n+1)_0\).
        ELSE
            Compensate\( B_{n+1_0}, NEW \_Handoff, B_1 \)
        END;
    CASE FUTURE TYPES:
    OTHERWISE:
        Reject();
\}

Bandwidth compensation calculation function

Compensate\( (b_i, FLOW\_TYPE, B_i) \)
// \( b_i \) is the bandwidth to compensate for
// \( FLOW\_TYPE \) is the type of connections
// \( B_i \) is the total bandwidth available for \( FLOW\_TYPE \)

\{
    \( B_{\text{Comp}} = B_i - \sum_{i \in \text{FLOW\_TYPE}} B_{i,0} \)
    \( B_{\text{Comp}} = B_i - \sum_{i \in \text{FLOW\_TYPE}} B_i \)
    IF \( B_{\text{Comp}} < b_i \)
        Reject();
    ELSE IF \( B_{\text{Comp}} \geq b_i \)
        Grant\( (b_i) \)
    ELSE:
        \( B_{\text{Comp}} = b_i - B_{\text{Comp}} \)
    \forall i \in k, B_{i} > B_{\text{Comp}}, \text{ subtract } \left( \sum_{i \in k} B_i \right) \text{ needed } \}
    Grant\( (B_{\text{Comp}} \sim B_{\text{Comp}}) \) to connection \((n+1)_{FLOW\_TYPE}\)
\}

Figure 2. CAC algorithm used in conjunction with TBFQ algorithm.
Each WT listens for its slot assignment in the downlink channel in order to access the appropriate uplink slots to transmit its data to BS. If more slots are required (due to traffic bursts), the WT conveys this request to the BS by in-band signaling. The BS then determines whether to grant more slots to the requesting WT based on the algorithm described below. Similarly, when reserved data slots are no longer needed, this information is also conveyed to the BS through in-band signaling. Conversely, for WT to receive data from the BS they listen for the appropriate downlink broadcast channel and receive packets addressed to it in the downlink data slots. The sources of downlink packets can be other WTs within the same cell, but most likely are fixed network sources that connect to the BS through a high-speed terrestrial link.

Each $L$-byte packet consumes $L$ tokens. For each connection $i$, $E_i$ is a counter that keeps track of the number of tokens borrowed from or given to the token bank. As tokens are generated at rate $r_i$, the tokens overflowing from the token pool are added to the token bank and $E_i$ is incremented by the same amount. When the token pool is depleted and there are still packets to be served, tokens are withdrawn from the bank by connection $i$, and $E_i$ is decreased by the same amount. Thus during periods that the incoming traffic rate of connection $i$ is less than its token generation rate, the token pool always has enough tokens to service arriving packets, and $E_i$ becomes positive and increasing. On the other hand, during periods that the incoming traffic rate of connection $i$ is greater than its token generation rate, the token pool is emptied at a faster rate than it can be refilled with tokens. In this case, the connection may borrow tokens from the bank.
The priority of a connection in borrowing tokens from the bank is determined by the priority index \(E_i/r_i\). Connections with the highest index have the highest priority in borrowing tokens from the bank; hence they will be serviced first. The number of tokens a connection may borrow from the bank at each time should be limited as it affects the burstiness of the outflow. To avoid starvation to other connections, ‘debt limit’ \(d_{ix}\) is imposed below which the connection can no longer borrow from the bank. The debt limit, \(d_{ix}\), for each connection in each service group (except CBR-type) is set to a negative value, so that a malicious connection in the same service group cannot affect the QoS of other well-behaved connections in the group. We also define ‘burst credit’, \(c_{ix}\), as the maximum number of tokens connection \(i\) from traffic type \(x\) can borrow from the bank each time. For a CBR-type source, \(r_i\) equals the source peak rate, and there is no need to borrow tokens from or deposit tokens in the bank. \(E_i\) ideally should stay zero all the time. However, for bursty sources, \(E_i\) can accumulate to a substantial level (due to lack of packet arrival at times), and then all of a sudden a sizeable traffic burst arrives. Therefore, \(c_{ix}\) should be set to a suitably large value for bursty sources. A connection may borrow tokens from the bank until its debt limit is reached, then it must wait until it has deposited enough tokens to the bank to reach the ‘creditable threshold’.

4.3. Slot allocation

The slot allocation is best described in the pseudo-code shown in Figure 4. The granting of slots is separated in two phases. First, the connections (or WTs) that have filled token pools will be granted a slot. If there are more connections than slots, the remaining connections will have to wait till the next frames. In phase two, connections may borrow tokens from the bank according to the algorithm described in the previous section.

4.4. Applicability of TBFQ under various MAC schemes

MAC protocols can be categorized as dedicated, random access, demand, and reservation. In dedicated assignment systems, there is no scheduling needed. One might argue that in random access systems scheduling is not required either. However, we argue that TBFQ scheduling algorithm can be used in random access MAC systems, as well as demand, priority or reservation-based systems. The range of MAC that TBFQ can work with covers multiple access MAC protocols such as TDMA TDD/FDD, CDMA, as well as hybrids.

DPRMA [9] is a TDMA/FDD based MAC. Its time slots are assigned to terminals according to bandwidth requirement. The reservation request bit in the header of the uplink slot is used for rate reservation. After a contention period, the base-station (BS) transmits in several reservation acknowledgement bits in the downlink header. To utilize TBFQ, connection parameters such as required rate should be exchanged during the CAC phase. Once admitted, the downlink of DPRMA could easily be modified for granting packet transmission using calculations determined by TBFQ algorithm. CPRMA [12] is a demand access scheme with contention-based reservation periods that grants transmissions at each slot according to the terminal with the most urgent need to transmit. Multimedia traffic is accommodated in this protocol through the polling process. The polling sequence for the reserved terminal is generated by a scheduling algorithm. However, no scheduling algorithm is specified. In order to implement TBFQ in CPRMA, reservation request contained in the mini-packet can include the number of packets that request transmission and the decision to grant the amount of tokens.
First stage: grant slot according to Token Pool
// This stage scans all the terminals and grants one slot to
// the terminal whose request and token pool is not zero.
// Terminal[i]: the i-th terminal, there are N terminals in total.
// Request[i]: Request from Terminal,
// TokenPool: the token pool size of Terminal,
// Grant[i]: slots that granted to Terminal,
// SatisfiedTerminals: total number of satisfied terminals.
FOR (i = 1; i <= N)
{
  IF ((Request[i] > 0) && (TokenPool[i] > 0)) THEN
    Assign one slot to Terminal[i];
    TokenPool[i]--; 
    UpdateCounters(i, 1)
}

Second stage: grant according to borrowing tokens from Bank
// This stage lets terminals to borrow from token bank until
// either all Slots_per_frame are consumed or all terminals are satisfied.
// TotalGrants: total number of slots granted in a frame.
// Slots_per_frame: number of slots per frame
// Credit[i]: number of slots borrowed by Terminal[i],
WHILE ((TotalGrants < Slots_per_frame) && (SatisfiedTerminals < N))
{
  i = HighestBorrowPriority()
  Borrow[i] = Borrow Budget(i)
  IF (Request[i] < Borrow[i]) THEN
    Borrow[i] = Request[i];
    Assign Borrow[i], number of slots to Terminal[i],
    Credit[i] += Borrow[i];
    UpdateCounters(i, Borrow[i])
}

Function UpdateCounters(i, slots_granted)
// Update the counters of Terminal[i], after slots granted
{
  TotalGrants += slots_granted
  Grant[i] += slots_granted
  Request[i] -= slots_granted
  IF (Request[i] == 0) THEN
    SatisfiedTerminals++
}

Function HighestBorrowPriority()
// Find the terminal that has the highest priority to borrow from the token bank.
{
  Max_borrow_allowed_this_frame = Slots_per_frame - TotalGrants;
  Find the unsatisfied Terminal[i] whose contribution to the bank is the maximum and it is eligible to borrow;
  RETURN j
}

Function Borrow Budget(i)
// Find the number of slots Terminal[i] is allowed to borrow.
{
  Borrow_allowed[i] = amount_contributed_to_token_bank - Debt_limit;
  RETURN min(Borrow_allowed[i], Max_borrow_allowed_this_frame, Burst_Credit)
}

Figure 4. Pseudo-code for slot allocation algorithm.
transmission would be determined by the TBFQ in the base-station and the terminals could be polled subsequently.

A prototype microcellular wireless ATM (WATM) network capable of providing QoS to multimedia traffic was developed in Reference [14]. The multiple access scheme used was dynamic TDMA/TDD based. The MAC accommodated both the dedicated, random, and demand assignment resource sharing schemes. Requests are sent to the base-station via dedicated reservation slots using slotted ALOHA. The requests are processed according to their QoS parameters and successful reservations are broadcasted in the downlink. This MAC framework can accommodate different scheduling algorithm including TBFQ. Another WATM MAC scheme developed in Reference [18] used a scheduling algorithm called priority regulated allocation delay-oriented scheduling (PRADOS) to determine transmission of packets over the radio interface. PRADOS combined leaky bucket flow control with earliest deadline first (EDF) scheme which required time stamping mechanism and an exchange of timing information. As we will demonstrate later in our results, this is both inefficient due to heavy exchanges required by the algorithm and ineffective due to stringent flow control.

CDMA access schemes offer co-existence of different types of traffic. Interference control is important to such co-existence. In Reference [21], a packet-oriented MAC protocol was used for carrying multiple traffic types based on the priority of the queue of each traffic type. An ideal feedback channel is assumed and users continue to transmit packets with probability \( P \) as long as their queue is not emptied. A lower probability corresponds to a higher priority. No limit on the number of traffic is imposed. Although this scheme is simple to implement and offers considerable multiplexing gain as bandwidth increases, packet loss probability and delay can be high due to contention resolution. This can easily be remedied by imposing CAC and simple scheduling algorithm to improve contention. The base-station can co-ordinate admissions process and allow for simple information exchange for scheduling. A light-weight scheduling algorithm will minimize information exchange and hence reduce packet loss probability and delay. TBFQ (in conjunction with a CAC) is a suitable candidate for such scheme as it requires only buffer occupancy information from the users.

Multidimensional PRMA [20] is a protocol suitable for TDMA/CDMA schemes. Again the mobiles will contend with specified probability. Probabilities for each type of service and each time slot of the next uplink frame are broadcast by the base-station in the downlink frame. The probabilities depend on the estimated number of backlogged terminals and are connected to a load-based access control to ensure the control of interference level of the CDMA components and the stability of the system. Although no scheduling algorithm was specified, the use of one would definitely be beneficial in terms of contention resolution and delay reduction. To deploy TBFQ for CDMA (or a hybrid) MAC schemes, priority calculation of TBFQ would have to be modified to incorporate interference calculation. However, this is beyond the scope of this paper. For performance demonstration purposes, we chose the TDMA/TDD contention-free MAC protocol used in Reference [30] because it minimizes the effect that contention could have on the overall systems performance, thereby providing clarity in understanding system performance due to scheduling.

The TDMA/TDD MAC scheme has a number of attractive features, including the possibility of ‘on-demand’ allocation of bandwidth. The fixed length frame is time-duplexed into an uplink and downlink channel, each further divided into control and data transmission periods. Slots
assigned for control purposes are divided into control mini-slots each holding a control packet. The BS has absolute control over the number of data/control slots in each frame and the WTs assigned to receive or send information during the data slots. Total channel data rate of 1.48 Mbps was chosen, with frame duration of 4 ms. There are 14 slots per frame for data transmission. Each packet size is 53 bytes. There are 20 bytes in the uplink control slot, 16 bytes in both the preamble and frame header.

5. PRELIMINARY ANALYSIS

Our analysis is focused on throughput fairness, complexity, and delay bound. The analysis of the algorithm is based on the MAC protocol discussed and will need to be adjusted accordingly when implemented in other wireless systems with different MAC protocol.

Definition 1
A connection is said to be backlogged during an interval $[t_1, t_2]$ if the queue for connection $i$ is never empty during that interval.

Theorem 1
The proposed scheduling algorithm has time complexity $O(1)$ for scheduling packets within $N$ admitted connections.

Proof
The computation complexity of packet generalized processor sharing (PGPS) [31] and TBFQ are $O(n)$ and $O(M)$, respectively, where $n \approx$ number of arrived packets among all backlogged sessions, $M \approx$ number of backlogged sessions, and $n \gg M$ generally. Since $M \leq N$, and $N$ is finite, the amount of calculation is known and takes constant time, hence the complexity of TBFQ is $O(1)$. □

Lemma 1
Rate guarantee: Under the CAC and the scheduling algorithm, if a connection is admitted with $r_i$, then the minimum rate that connection $i$ is served by the scheduler is $r_i$, where $r_i$ is the token generation rate for connection $i$.

Proof
Under the CAC defined, a connection can only be admitted if $\sum_n r_i < kC$ where $C$ is the system total bandwidth and $0 < k < 1$. For connection $i$, a token is generated every $1/r_i$ interval. If we let each TDMA frame to have a period of $T_f$ and there are total of $n$ backlogged connections. Ideally, it would be good if the tokens of each connection are generated and distributed evenly over time, but we know that is not the case here. And in the worst case, all tokens from each connection are generated at the same time $t_0$ and each backlogged connection makes request for packet transmission. Let the next frame after $t_0$ be $f_0$ which begins at time $t_1$. The scheduler receives all the requests and schedules them in the order, $\Omega$, where no preference is given to any connection because they all have their token pools filled. Let token $\varphi_i$ be the token generated for connection $i$ which is last in $\Omega$. If $n > \delta$, where $\delta$ is the number of slots in a frame, then $\varphi_i$ will be consumed in frame $f_0 + \lfloor n/\delta \rfloor$ which begins at time $t_2 = t_1 + \lfloor n/\delta \rfloor$, and a packet from connection $i$
is served. The next token generated for connection \( i \) is at time \( t_0 + 1/r_i \) and in the worst case a token is generated for each of the other connection \( n - 1 \). Since \( i \) will be served at the end of \( \Omega \), the packet from connection \( i \) will be transmitted in a frame no later than \( \lfloor n/\delta \rfloor \). This proves the interval of service for connection \( i \) is no later than \( 1/r_i \), the minimum rate of service is guaranteed to be at least \( r_i \).

**Definition 2**
We define the *Start-up Latency* to be the maximum length of time between the instant the first packet of a new flow arrives in its queue and the instant the last bytes of this packet is scheduled.

**Theorem 2**
During an execution of the TBFQ scheduling discipline serving \( n \) active connections at a link of maximum rate \( C \), the start-up latency, \( \text{Latency}_{-\text{TBFQ}} \), of a newly active flow has an upper bound given by:

\[
\frac{nM}{C} \quad \text{if } n \leq N
\]

\[
\frac{NM}{C} \times \left\lfloor \frac{n}{N} \right\rfloor \quad \text{if } n > N
\]

where \( M \) is the packet size in bytes and \( N \) is the number of slots in a frame.

**Proof**
When the first packet of a newly active flow arrives, the flow is served after all the \( n \) previously active flows are served. Since with TBFQ, the packets are of constant size, each flow can be served one packet (\( M \) bytes) maximum. If \( n < N \), all the active flows can be served in one frame, the bound is within one frame: \( (n \cdot M/C) \); if \( n > N \), all the active flows cannot be served in one frame, the bound is more than one frame:

\[
\frac{NM}{C} \times \left\lfloor \frac{n}{N} \right\rfloor \quad \text{if } n > N
\]

The statement of the theorem is proven.

**Definition 3**
Let us define the service received by a connection \( i \) during a backlogged period \( [t_1, t_2] \) to be \( S_{[t_1, t_2]} \).

**Definition 4**
Let us define a throughput fairness index, \( \text{FI} \) such that

\[
\text{FI}_{[t_1, t_2]} = \left| \frac{S_{[t_1, t_2]}}{r_i} - \frac{S_{[t_1, t_2]}}{r_j} \right| \quad \forall i \neq j
\]

where \( S_{[t_1, t_2]} \) is the service that connection \( i \) received during \([t_1, t_2]\). A service discipline is said to be fair if \( \text{FI}_{[t_1, t_2]} \) is bounded.

**Theorem 3**
For any backlogged interval \([t_1, t_2]\), the fairness index \( \text{FI}_{[t_1, t_2]} \leq d_x^i \), where \( d_x^i \) is the *debt limit* of a connection.
Proof
For any backlogged interval \([t_1, t_2]\), and for any connection \(i\) and \(j\), where \(i \neq j\), the service received by connection \(i\) is \(S_i[t_1, t_2] \geq r_i[t_1, t_2]\), which is from the rate guarantee in Lemma 1. However, it is possible for connection \(i\) to receive more than its minimum service. When the token from each token pool is consumed, the remaining bandwidth will be allocated to connections according to their priority index \(E_i/r_i\). The connection with the highest priority will receive burst credit, \(c_\Delta\), and \(E_i\) is decremented by \(c_\Delta\). If after the connection is awarded with \(c_\Delta\), it is still the highest in the priority, it will continue to receive \(c_\Delta\) until it has moved to lower priority or the debt limit \(d_i\) is reached. In other words, \(c_\Delta(t)\) is bounded by \(d_i\). So,
\[
r_i[t_1, t_2] \leq S_i[t_1, t_2] \leq r_i[t_1, t_2] + c_\Delta[t_1, t_2]
\]
for any connection \(i\) within the backlogged period. So among the backlogged connections, there exists a connection \(i\) with service rate
\[
S_i[t_1, t_2] \geq r_i[t_1, t_2]
\]
The normalized service received by \(i\) is
\[
\frac{S_i[t_1, t_2]}{r_i} \geq \frac{r_i[t_1, t_2]}{r_i}
\]
Equally, there exists a connection \(j\), where \(j \neq i\), with service rate
\[
S_j[t_1, t_2] \leq r_j[t_1, t_2] + c_\Delta[t_1, t_2]
\]
And the normalized service received by \(j\) is
\[
\frac{S_j[t_1, t_2]}{r_j} \leq \frac{r_j[t_1, t_2]}{r_j} + \frac{c_\Delta[t_1, t_2]}{r_j}
\]
Subtracting the equations, the fairness index \(FI[t_1, t_2]\) is
\[
\left| \frac{S_j[t_1, t_2]}{r_j} - \frac{S_i[t_1, t_2]}{r_i} \right| \leq \frac{c_\Delta}{r_j}
\]
The theorem follows because both \(c_\Delta\) is bounded by \(d_i\) and \(r_j\) is the rate of the token rate which is fixed. 

Definition 5
Work conserving: A scheduling discipline is called work conserving if the server is never idle when there is a packet to transmit. Non-work-conserving disciplines, however, allow the server to be idle if no packet is eligible to be transmitted.

6. SIMULATION RESULTS

We focus on the performance of packet transfer in both uplink and downlink wireless channels. All simulations are conducted using OPNET. We assume the quality of the wireless link is managed by the physical layer and the channel error model used as described in Section 3.3. During bad states, no re-transmission is allowed. Packets are transmitted without errors during good states.
6.1. Simulation parameters

Each WT must first pass through the CAC. We assume that the underlying link layer can determine the signal quality of each WT and that a connection is admitted only if that layer can maintain the loss requirement of such traffic class and that it satisfies the CAC in Section 4.1. All simulations were run for 5000 s to assure accurate results; the 95% confidence interval is, at worst, within 5% of the values shown. We compare the performance of TBFQ with FIFO (first-in-first-out), PGPS, EDF, and RR schemes. For PGPS and EDF, we allow the transmission of timing information to be transmitted over the wireless medium to the BS. These schemes are used as performance benchmarks. For the simulations with voice and video traffic, token bucket parameters are specified in Table I.

To ensure a fair comparison, same leaky bucket parameters are used. EDF is a scheme used in PRADOS which was designed to work with the MASCARA MAC protocol [18]. Leaky bucket was used as the flow control mechanism in PGPS, EDF, and RR; it facilitates delay bound calculation.

In implementing the leaky bucket with the ‘other’ schedulers, we define priorities for traffic classes similar to Reference [30]. When the scheduler services ‘conforming’ requests, defined as requests that belong to connections whose token pool is non-empty, it follows the priority table. Within each priority class, the scheduler serves the request of each connection as long as slots are available and the connection’s token pool is not empty. The order of the packets served in each class is determined by the algorithm of the schedulers. Every time a slot is allocated to a connection, a token is removed from that connection’s token pool. Within the same priority class, the scheduler gradually allocates one slot at a time to the connection that has the most tokens left in its token pool. When all the token pools are emptied, the scheduler serves ‘non-conforming’ requests. It starts allocating slots for connections, starting from the highest priority (CBR) down to the lowest priority (UBR) until all the requests are served.

<table>
<thead>
<tr>
<th>Table I. Scheduler parameters for voice and video traffic.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token bucket parameters for voice traffic</strong></td>
</tr>
<tr>
<td>TBFQ</td>
</tr>
<tr>
<td>‘Burst credit’</td>
</tr>
<tr>
<td>‘Debt limit’</td>
</tr>
<tr>
<td>‘Creditable threshold’</td>
</tr>
<tr>
<td>Token generation rate</td>
</tr>
<tr>
<td>Token pool size</td>
</tr>
<tr>
<td>P-GPS, EDF, RR, FIFO</td>
</tr>
<tr>
<td>Token generation rate</td>
</tr>
<tr>
<td>Token pool size</td>
</tr>
<tr>
<td><strong>Token bucket parameters for video traffic</strong></td>
</tr>
<tr>
<td>TBFQ</td>
</tr>
<tr>
<td>‘Burst credit’</td>
</tr>
<tr>
<td>‘Debt limit’</td>
</tr>
<tr>
<td>‘Creditable threshold’</td>
</tr>
<tr>
<td>Token generation rate</td>
</tr>
<tr>
<td>Token pool size</td>
</tr>
<tr>
<td>P-GPS, EDF, RR, FIFO</td>
</tr>
<tr>
<td>Token generation rate</td>
</tr>
<tr>
<td>Token pool size</td>
</tr>
</tbody>
</table>
6.2. Downlink performance

We studied the effect of traffic arriving at the BS and sent to the WTs through the downlink broadcast channels. The traffic can be originated from either wireline network or uplink traffic. The QoS parameters that we are interested are delay and throughput.

In Figure 5, the cumulative distribution of delay for voice is shown. For voice traffic, we allow system loading of 0.86. For FIFO and RR, 50% of delay is below 40 ms. This shows the inability of FIFO and RR in coping with error conditions. In FIFO, the head-of-queue (HOQ) packet of each queues is compared to determine which comes first and is scheduled according to FIFO. This order is fixed, and if a connection goes into bad state it will have to wait until it comes out of bad state and the other HOQ packets are served first. The second packet in the same queue faces the same challenge. RR is slightly better in that there is a better chance that when an HOQ packet comes out of the bad state, it can be served earlier. If a queue has just become good, and RR is pointing at the previous queue (in the queue sequence), then it will be served almost immediately. TBFQ has very similar delay performance from EDF and TBFQ. This is mainly because the traffic is constant-bit-rate and system load is moderate. The minor difference comes from treatment of error conditions. The effect is amplified when we look at video frame delays of real-time video traffic (Figure 6).

In this case, TBFQ has a clear performance advantage. If we consider a large video frame that has arrived at the BS, the flow control mechanism in PGPS and EDF restricts it through the leaky bucket mechanism. Once admitted into the out-going queue, packets are then time-stamped and served according to their service scheme and wait for transmission. If the destination is in a bad state, sorting in the out-going queue is necessary to make room for packets whose destination is in good state. There is no mechanism in both EDF and PGPS to allow a burst to borrow bandwidth from the future; otherwise, it would not meet the instantaneous fairness that these schemes are designed for. In TBFQ, arriving burst first stay in their in-coming queue, they are accumulated in their queue until they are ready to be served for transmission. The out-going queue is maintained only for packets to be transmitted immediately. If a connection has accumulated enough $E$ before a large video frame arrives, it is very likely that it can borrow tokens in advance from the bank. If that is the case, the overall delay is lowered, which results in Figure 6. This is a tradeoff with instantaneous fairness. For

![Figure 5. Packet delay CDF for voice traffic only (load = 0.86).](image-url)
example, during the period when the large video frame of connection A is being served, connection B may have arrived in its queue a large frame of data also. Connection B would have to concede to connection A until its \( E/r \) exceeds that of connection A (assuming \textit{debt.limit} is not reached). At lower system loading scenario (0.52 or less), performance difference is negligible among TBFQ, EDF and PGPS—over 90\% of the delays are less than 12 ms. At higher loading scenario (0.94), the system saturates and the performance of TBFQ, EDF and PGPS all lower accordingly—48.7\% of delays are below 150 ms. The performance difference gap between TBFQ, EDF and PGPS starts to diminish. We believe that there is an optimum loading point where TBFQ can maximize its statistical multiplexing gain.

Mean throughput performance was measured for data traffic (Figure 7). The token generation rate of 64 kb/s is set for all connections. With 30 data connections admitted (81\% loading), 51.1\% of packets had 60 kbps for TBFQ, 45.3\% had 57.5 kbps for EDF, 58\% had 57.7 kbps for PGPS, 54.5\% had 55 kbps for RR, and 41.2\% had 45 kbps for FIFO. Maximum of 65 kbps was achieved by TBFQ, EDF and PGPS. This is because of the less than maximum loading so there is extra bandwidth for additional services.
6.3. Uplink performance

In the uplink channel, we mixed the voice and video traffic. We focus on the link utilization, isolation, fairness, and the selection of required bandwidth. In Figure 8, the effects of voice traffic load on voice packet mean delay is shown with different number of concurrent video connections. No packet discard or tagging was exercised. LB policing was used. The reason for lower delay performance seen in TBFQ is the same as what we have seen in the downlink. The poorer than expected performances of PGPS and EDF are caused mainly by the delays in the LB. Though the use of policing is an implementation decision, we believe it is necessary. If policing were removed, PGPS and EDF would have the best delay performance. However, the improvement over TBFQ is not significant (less than 5 ms) in the normal loading condition. However, the complexity of both EDF and PGPS makes them impractical to deploy in any wireless network. The computation complexity of PGPS and TBFQ are $O(N)$ and $O(M)$, respectively, where $N \approx$ number of arrived packets in the outgoing queue, $M \approx$ number of backlogged sessions. It can easily be determined that the amount of packets accumulate in the outgoing queue within $(t_1 - t_0)$ is $M \approx rN(t_1 - t_0)/L$, for constant-bit-rate connections with token generation rate $r$, and $L$ is number of bytes per packet. Within 100 ms, ten 64 kbps connections can accumulate as many as 150 packets in the queue. Generally $N \gg M$, and since $N$ is bounded and generally negligible compared to $M$, the complexity of TBFQ is approximately $O(1)$.

Figure 9 shows the schedulable region where the violation objective and delay tolerance for voice and video are (10%, 100 ms) and (10%, 40 ms), respectively. TBFQ supports the greatest load of video traffic for a given voice load, but EDF and PGPS provide performances that are nearly as good. The performances for EDF and PGPS would improve and exceed that of TBFQ if policing was removed or if the LB token rates were increased. A study has been performed to

Figure 8. Voice packet mean delay with video connection.
look at the impact of the token rate on the delay performance. The delay violation tolerance for video has been set to 100 ms. Figure 10 depicts the findings. The range of token rates was set so that it covers the average rate as well as several times that average. We found that EDF and PGPS would eventually perform better than TBFQ when the token rate was increased to a high enough level. However, we also note that TBFQ’s performance improves quicker than the others when the token rate is increased. The advantage of this becomes clear when the system has to determine (at the BS) the token rate (between the average and the maximum rate) to be used for a bursty stream. Determining the token rate of a bursty variable-bit-rate traffic is not a trivial task. TBFQ has a wider tolerance of token rates for acceptable performance and therefore for non-bursty sources such as the voice connections, it makes no difference whether we allocate the peak rate 64 kb/s or its average of \(0.6 \times 64\) kb/s. We can look at this from an alternate view; in bursty traffic streams there will be many occasions when the traffic will exceed the assigned token rate, which will cause QoS to degrade. By using TBFQ, the BS can gracefully accept the temporary traffic contract violations and maintain acceptable QoS to the WTs. This is due to the ‘soft’ QoS provisioning capability of TBFQ.

Another property of the scheduler that we have discovered is its alteration of the traffic profile. If traffic behaviour is modified significantly, it will be treated as violating the original
traffic contract and may be subject to discarding and/or tagging in the network core. This in effect will have an overall impact on the end-to-end QoS. We have studied the inter-arrival distribution of video packets arriving at the BS by varying token generation rates. It can be shown that the variance in traffic distribution decreases with increasing token rate, and that TBFQ maintains a lower variance in the traffic distribution than the others, even at lower token rates.

Fairness performance of TBFQ is demonstrated in Figure 11 where only a time segment of the simulation is shown. We modify the traffic models so that their rate profiles are increased. We load the system to 94% with a malicious video connection (connection 1) and 5 well-behaved video connections (connections 2–6). The video connection 1 has an average rate of 408 kb/s and peak rate of 1024 kb/s, and each of the remaining video connections is modified to have an average rate of 204 kb/s and peak rate of 512 kb/s. Token rate of 512 kb/s is assigned to each connection, so connection 1 is the ‘malicious’ source. By assigning peak rate as the token rate for connections 2–6, the packet delay performance is expected to be quite good. However, connection 1 is under-provisioned and the result shows poor delay performance as expected. The excess traffic from connection 1 (malicious) does not affect the delay performance of the other well-behaved connections. This can easily be explained. As connection 1 generates a large burst, it continues to borrow tokens from the bank. During that time, other connections may not be able to borrow from the bank but they will at least be served by their token rate which satisfies their peak rate requirement. When connection 1 reaches the debt limit, no token is allowed to be borrowed; however, it will continue to be served at its own token rate. The remaining connections will enjoy at least their minimum reserved rate plus a share of excess bandwidth, which is now share by only 5 connections instead of 6. Hence, more services can be provided.

7. CONCLUSION

We have proposed and described the TBFQ algorithm for uplink and downlink scheduling in next generation wireless packet networks under a generic contention-free MAC protocol with error-free requests. We believe this can be applicable to FDMA/TDMA/CDMA systems. By decoupling scheduling function from a specific MAC protocol, we proposed a framework that allows us to do the following: focus on the behaviour of our scheduling algorithm, extend the
algorithm to work with other MAC protocols, and offer potential for seamless QoS in heterogeneous environment. A CAC was also proposed to work in conjunction with the TBFQ. We used simple channel error model in our studies. The results presented show that, when compared with some of the well-known broadband packet scheduling techniques, TBFQ performs quite well in servicing multimedia traffic in heterogeneous wireless packet networks. The delay performance is comparable, and better in some cases, than the commonly used algorithms. Its QoS provisioning capability allows graceful acceptance of traffic that temporarily violates its profile. This is particularly important for managing bandwidth allocation in the BS because it is not always trivial to know a priori the correct parameters of a connection from a mobile terminal roaming from another system, and traffic often exceed their profile. We also feel that this is important when operating in heterogeneous wireless environment where connection profiles and bandwidth allocation often do not match when roaming across different wireless systems. In addition to being able to serve heterogeneous traffic, TBFQ has shown to be capable of diverse performance objectives. It was shown that the TBFQ scheme has good fairness and isolation properties. Service differentiation can be achieved within the same traffic class. TBFQ requires only simple WT status information to be transmitted to the BS for scheduling, and it has the benefit of minimizing the processing overhead in the WT that is often faced with power constraints. In terms of complexity, we have shown that TBFQ has the complexity of $O(1)$ which is desirable for any wireless system. However, there are much work remains to be done in the framework of using scheduling algorithms for QoS guarantees in a heterogeneous wireless environment—notably the roaming in ad hoc networks, its applications in other MAC systems, and QoS mapping. Future work will also include extended analysis of the performance of the algorithm.

REFERENCES


AUTHORS’ BIOGRAPHIES

**William Wong** received the BESc (honours) degrees in Electrical Engineering from the University of Western Ontario, in 1991, and MEsc and MSc degrees in electrical engineering and computer science in 1993 and 1994, respectively. He is a PhD candidate with the Electrical and Computer Engineering Department at the University of British Columbia.

Since 1994, Mr Wong had been working in the development and manufacturing of mobile and fixed handset, and managed a team of 60 engineering staff in the wireless hardware and software design. Later he joined Philips Research Lab and worked in the wireless ATM research project before joining the Department of National Research as a Defence Scientist in 1999. He was a founder of MySkyWeb, Inc., which was a software technology development company and had patents pending. In 2003, he was the national coordinator of the CWChet (Canada Network of Wireless Centre) and led the establishment of the first network in the world supporting the Canadian wireless industry. He is also a member of the Wireless Internet and Systems Experimentation Laboratory (WISELAB) in the Communication Research Centre (CRC). His research interests include B3G/4G networks, Quality-of-Service, Medium Access Control protocols, WWAN/WLAN interoperability, security, and wireless applications.
Helen Y. Tang received her BEng and MEng in computer engineering from Central South University, China, in 1992 and 1995, respectively. She is a PhD candidate at the Department of System and Computer Engineering, Carleton University, Ottawa, Ont., Canada. She is an Ontario Graduate Scholarship holder. She had worked at the Communications Research Center as a research engineer in 2003 and Spacebridge Networks as a research intern in 2001. Her research interests include medium access control, scheduling algorithms and performance analysis of communication networks.

Victor C. M. Leung received the BASc (Hons.) degree in electrical engineering from the University of British Columbia (U.B.C.) in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at U.B.C. on a Natural Sciences and Engineering Research Council Postgraduate Scholarship and obtained the PhD degree in electrical engineering in 1981.

From 1981 to 1987, Dr Leung was a Senior Member of Technical Staff at Microtel Pacific Research Ltd. (later renamed MPR Teltech Ltd.), specializing in the planning, design and analysis of satellite communication systems. He also held a part-time position as Visiting Assistant Professor at Simon Fraser University in 1986 and 1987. In 1988, he was a Lecturer in the Department of Electronics at the Chinese University of Hong Kong. He joined the Department of Electrical Engineering at U.B.C. in 1989, where he is a Professor, holder of the TELUS Mobility Industrial Research Chair in Advanced Telecommunications Engineering, and a member of the Institute for Computing, Information and Cognitive Systems. He was a project leader and a member of the Board of Directors in the Canadian Institute for Telecommunications Research, a Network of Centres of Excellence funded by the Canadian Government. His research interests are in the areas of architectural and protocol design and performance analysis for computer and telecommunication networks, with applications in satellite, mobile, personal communications and high speed networks.

Dr Leung is a Fellow of IEEE and a voting member of ACM. He is an editor of the IEEE Transactions on Wireless Communications, and an associate editor of the IEEE Transactions on Vehicular Technology.