

# MAC Protocol and Traffic Scheduling for Wireless ATM Networks

**Nikos Passas, Lazaros Merakos, Dimitris Skyrianoglou**

Communication Networks Laboratory

Department of Informatics

University of Athens

15784 Athens, Greece

E-mail: {passas,merakos,dimiski}@di.uoa.gr

Tel: +301 7248154 (x334)

Fax: +301 7219561

**Frédéric Bauchot, Gérard Marmigère**

Compagnie IBM France

CER, 06610 La Gaude, France

E-mail: {fbauchot,marmiger}@vnet.ibm.com

Tel: +334 92115592

Fax: +334 93247809

**Stéphane Decrauzat**

Institut Eurécom

2229 route des Crêtes

BP 193, 06904 Sophia-Antipolis Cedex, France

E-mail: decrauza@eurecom.fr

Tel: +334 93002645

Fax: +334 93002627

## Abstract

The Medium Access Control (MAC) protocol and the underlying traffic scheduling schemes defined in the Wireless ATM Network Demonstrator (WAND) system being developed within project Magic WAND are presented. Magic WAND is investigating wireless ATM technology for customer premises networks in the framework of the Advanced Communications Technologies and Services (ACTS) programme, funded by the European Union. The MAC protocol, known as MASCARA, is a hub-based, adaptive TDMA scheme, which combines reservation- and contention-based access methods to provide multiple access efficiency and quality of service guarantees to wireless ATM terminal connections sharing a common radio channel. The traffic scheduling policies are delay oriented to meet the requirements of the various traffic classes defined by the ATM architecture. Simulation results are presented to assess the performance of the proposed protocols.

# 1. Introduction

Wireless networking is one of the fastest growing telecommunication technologies today [1]. Asynchronous Transfer Mode (ATM) is also considered to be one of the major drives for world telecommunications, as it provides an easily scaleable, high-speed switching scheme for various types of services like sound, images and data [2]. Wireless ATM combines the advantages of wireless operation and freedom of mobility with the service advantages and quality of service (QoS) guarantees of fixed ATM. It aims to provide true ATM wireless access, flexible bandwidth allocation, QoS negotiation and guarantees, which cannot be supported by the early proprietary wireless local area networks (LANs), which were designed with mainly conventional LAN data traffic in mind. Emerging standards, such as HIPERLAN or IEEE 802.11, have been designed to provide wireless access to corporate networks, but do not incorporate the ATM technology over the air [3].

ACTS Project Magic WAND (Wireless ATM Network Demonstrator) is aiming to introduce ATM over the air all the way to the mobile terminal, and to cover a wide range of functionalities, from basic data transmission, to shared multimedia applications. The main components of the WAND system, as shown in Figure 1, are:

- *Mobile Terminals* (MTs), the end user equipment, which are basically ATM terminals with a radio adapter card for the air interface,
- *Access Points* (APs), the base stations of the cellular environment, which the MTs access to connect to the rest of the network,
- an *ATM Switch* (SW), to support interconnection with the rest of the ATM network, and
- a *Control Station* (CS), attached to the ATM switch, containing mobility specific software, to support mobility related operations, such as location update and handover, which are not supported by the ATM switch.

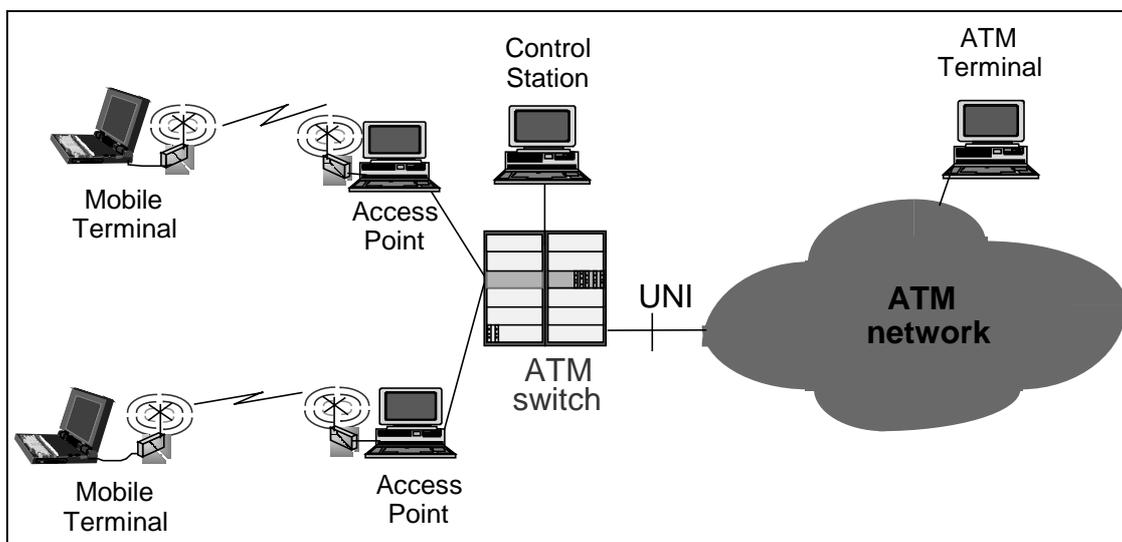


Figure 1: A WAND system

An important system design issue for WAND, and wireless ATM systems in general, is the design of an efficient medium access control (MAC) protocol for the radio interface. This protocol must be able to support all or a useful subset of ATM services with often conflicting requirements, and guarantee a QoS for every connection. Accordingly, advanced traffic scheduling is required to fulfil these requirements.

In this paper, we present the concepts of the MAC protocol and traffic scheduling in the radio interface, as are currently worked out in the WAND project. Since the MAC protocol is based both on reservation and contention techniques, it has been named **Mobile Access Scheme based on Contention And Reservation for ATM**, or MASCARA. We focus on the structure of the protocol and the scheduling of ATM traffic in the radio interface. Aspects, such as support of handover for mobility, are not discussed here, although they definitively impact the protocol. The reader interested in such a topic is referred to [4]. Network management considerations, which are also not addressed here, can be found in [5].

The paper is organized as follows. Section 2 discusses the general characteristics of an efficient MAC protocol for wireless ATM and describes the structure of MASCARA. Section 3 discusses traffic scheduling requirements for the radio interface of WAND and proposes a novel scheduling algorithm. In Section 4, simulation results on performance of the scheduling algorithm are presented. Finally, Section 5 contains our conclusions.

## **2. MAC in the Radio Interface of WAND**

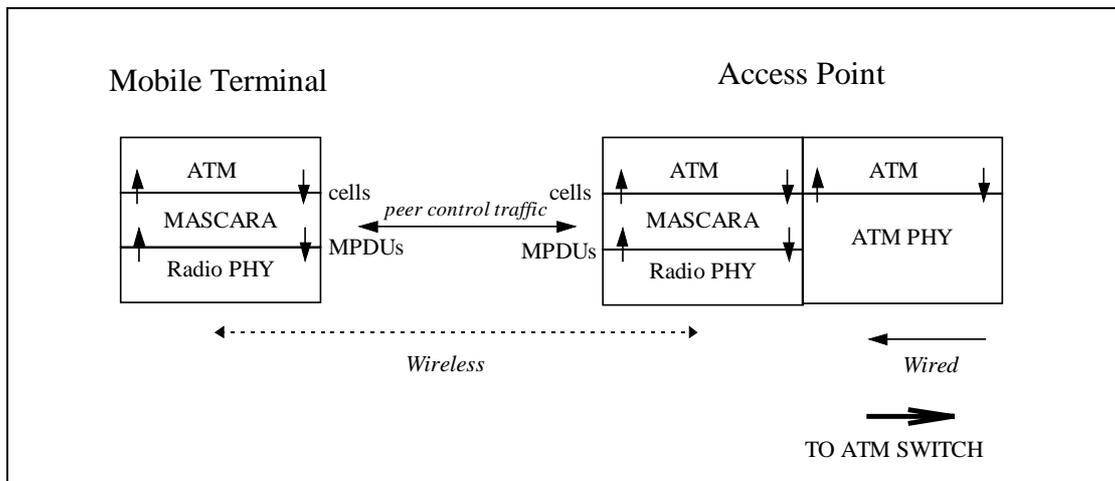
The MAC protocol is a critical component of WAND as its role is to provide wired-like services to ATM connections, in addition to controlling the access to the radio medium. This transparency is a challenge, because standard ATM has clearly not been designed to work in a wireless environment. Assumptions that have been taken into account for the design of ATM, like quasi-error free transmission medium, full duplex and point-to-point connections, are clearly not valid any longer. The radio medium is characterized by a high bit error rate (BER), the transmission mode is intrinsically broadcast or at least multicast, and the scarcity of available radio bandwidth calls for a time division duplex (i.e., half duplex) mode.

The multiple access technique used in MASCARA for uplink and downlink transmissions (respectively, from the MTs to the AP of their cell and from the AP to its MTs) is based on time division multiple access (TDMA), where time is divided in variable length time frames, which are further subdivided in time slots. Time slot duration is equal to the time needed to transmit the ATM cell payload (i.e., 48 bytes) plus the radio and MAC specific header. The multiplexing of uplink and downlink traffic is based on time division duplex (TDD). Slot allocation is performed in a dynamic and adaptive manner so that bit rates can readily match current user needs, by allocating more or fewer time slots per time frame. This is critical in servicing ATM connections, especially those with variable bit rate, because bandwidth can be allocated dynamically, and the resulting statistical multiplexing gain yields high resource utilization.

The MAC protocol belongs to the MAC layer of each MT and AP, which will be referred to as the MASCARA layer. The MASCARA layer interacts with

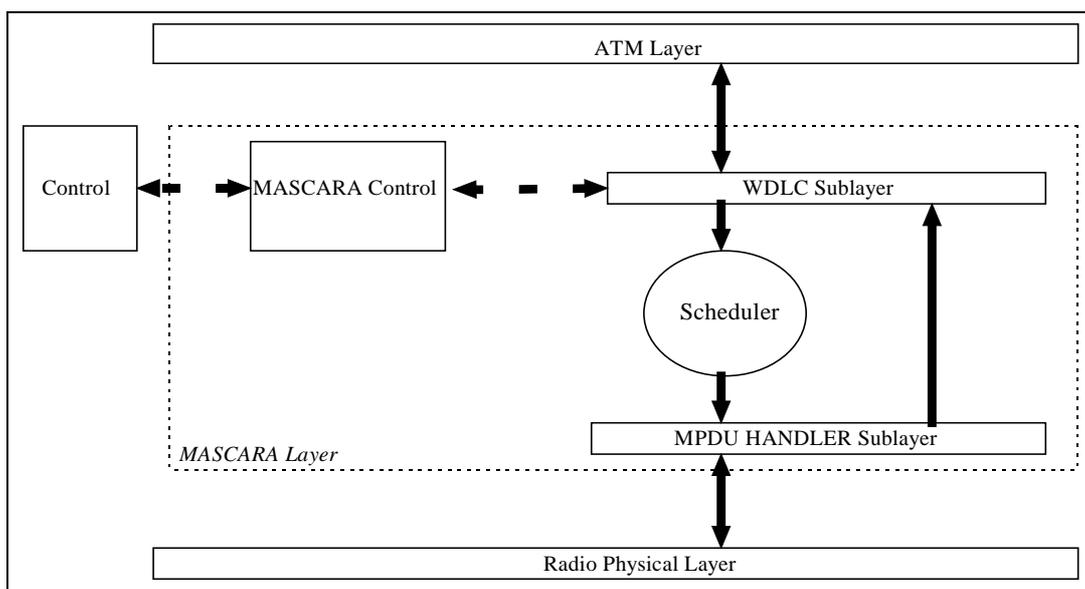
- the RF-based physical layer,
- the standard ATM layer, which sends and receives ATM cells exactly in the same way as if it were in a fixed ATM layer, and
- the peer MASCARA layer; peer-to-peer MAC control traffic is exchanged through specific control messages (Figure 2).

Cells coming from the ATM layer are formed into MAC Protocol Data Units (MPDUs) and are delivered to the radio physical layer for transmission, while MPDUs coming from the physical layer are processed and ATM cells are extracted.



**Figure 2: MASCARA layer interaction**

The MASCARA layer consists of several components, each one designated to a specific function. Figure 3 shows these different components, the data flow (solid lines), as well as the control information flow (dashed lines).



**Figure 3: MASCARA components**

The MASCARA control entity is responsible for all MASCARA components, and controls, through peer-to-peer control messages, the connection between a MT and an AP (e.g., provides control features to establish, maintain and release wireless connections). It also integrates radio power control and is involved in the handover process (e.g., detects that a radio quality threshold has been crossed).

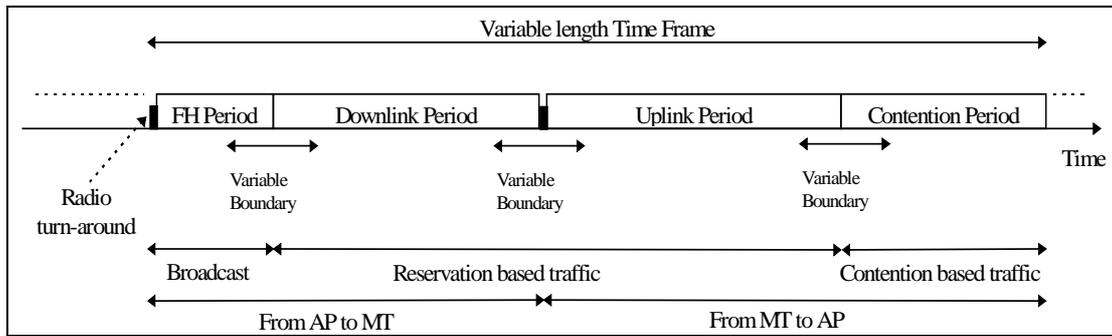
Data link control is required, as the quality of the wireless channel is significantly worse than conventional wired media (BER can reach values as bad as  $10^{-3}$ ). For this purpose, the MASCARA layer includes a Wireless Data Link Control (WDLC) sublayer, which is responsible for error control over the radio link. The selection of the WDLC technique depends on the exact constraints imposed on each ATM connection, such as delay or loss constraints. For instance, candidate strategies may be a combination of Forward Error Correction (FEC) or Automatic Repeat reQuest (ARQ) schemes, such as Go-back-N, Selective Repeat (SR), and extended error correction code. The selection of such a strategy is being finalized within the WAND project, but whatever the technique used, the current MASCARA design assumes that a WDLC overhead is added to each individual ATM cell transmitted (see Figure 6).

The MPDU Handler is in charge of handling MPDUs, both for the transmit and for the receive flows. As explained below, this corresponds to building MPDUs from a sequence of ATM cells (transmit flow) and extracting all the individual ATM cells from a received MPDU (received flow).

The Scheduler is responsible for scheduling the traffic transmitted through the wireless medium. In other words, it is the component that decides on the time an ATM cell will be transmitted. We introduce two types of Schedulers: the Master Scheduler, which runs in the APs, and the Slave Scheduler which runs in the MTs. The Master Scheduler of an AP determines how the slots of each time frame are allocated to its associated MTs and to downlink transmissions. The Slave Scheduler of a MT is responsible for prioritizing its own transmissions within the slots allocated to the MT by the Master Scheduler. In this way, protocol efficiency is improved since each MT sends, within the slots allocated to it, the part of its traffic that must be serviced first. A well designed scheduling mechanism should allocate the slots in such a way so that it maintains the agreed QoS to the uplink and downlink ATM connections sharing the radio bandwidth, and, at the same time, attains high bandwidth utilization. The scheduling algorithm for the Master Scheduler of MASCARA is described in Section 3.

## **2.1 The Multiple Access Protocol**

The MASCARA protocol is built around the concept of MAC Time Frame: a variable length timing structure during which ATM data traffic and / or MAC control information flows are exchanged through the wireless medium. As shown in Figure 4, the MASCARA time frame is divided into a downlink period for downlink data traffic, an uplink period for uplink data traffic and an uplink contention period used for MASCARA control uplink information. Each of the three periods has a variable length, depending on the traffic to be carried on the wireless channel.



**Figure 4: MAC Time Frame Structure**

A basic requirement for the multiple access protocol in WAND is to maintain fixed ATM QoS guarantees over the air interface. Such guarantees are easier to provide with a centralized AP-based protocol, and this is the approach chosen in MASCARA. The AP controls the allocation of bandwidth to uplink and downlink connections, and all MT communications pass through their associated AP.

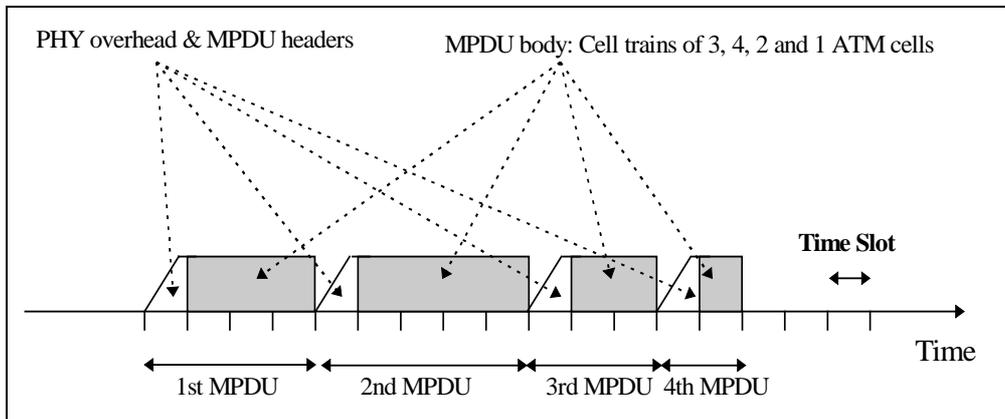
The AP schedules the transmission of its uplink and downlink traffic and allocates bandwidth dynamically, based on traffic characteristics and QoS requirements, as well as the current bandwidth needs of all connections. The traffic characteristics (e.g., peak and sustainable rate, burstiness) and QoS requirements (e.g., cell delay tolerance, cell delay variation tolerance) of a connection are made available to the AP scheduler during the call setup phase. The current needs of an uplink connection from a specific MT are sent to the AP through MT “reservation requests”, which are either piggybacked in the data MPDUs the MT sends in the uplink period, or contained in special “control MPDUs” sent for that purpose in the contention period.

To describe the protocol operation, let us consider a newly established uplink connection. In the first frame following the connection establishment, the MT sends a control MPDU in the contention period, in accordance with a random access protocol (e.g., slotted-ALOHA [6], stack algorithm [7]). The specific protocol to be used in WAND is currently under investigation. This control MPDU carries a reservation request for the number of ATM cells waiting in the MT connection output buffer. Upon successful reception of this reservation request, the AP will reserve the required slots for the requested transmission of the ATM cells for that connection, in the uplink periods in subsequent frames, in accordance with the scheduling algorithm described in Section 3. A MT that already has slots allocated to it in an uplink period, may send reservation requests contention-free, piggybacked in the MPDUs that it transmits in the reserved slots. Such reservation requests identify the MT connection and the number of ATM cells of that connection in the MT output buffer. Thus, the contention period is only used by MTs that have cells to send and have no reserved slots in the uplink period.

At the end of a frame, the AP constructs the next frame, according to the MASCARA scheduling algorithm, taking into account the reservation requests sent by the MTs, the arriving cells for each downlink

connection, and the traffic characteristics and QoS requirements of all connections. By frame construction we mean the length of the frame and of each of its periods, and the position of the slots allocated to each downlink and uplink connection. This information is broadcast to the MTs in the frame header (FH period) at the beginning of each frame. By reading the FH, a MT knows the slots in the downlink period it has to listen to, and the slots reserved for its connections in the uplink period. Next we elaborate on the MPDU construction used in MASCARA.

The physical layer overhead of the wireless medium is considerably larger than that of wired media. Hence, efficient data transmission can only be achieved if the length of transmitted data packets is not too small. On the other hand, the high BER, characterizing the wireless media asks for not-too-large data packets to keep the packet error rate at tolerable values. In the ATM world, the information granularity corresponds to a 53 bytes long ATM cell. This piece of data is considered short when compared, for instance, to conventional LAN MAC frames (such as IEEE 802.3 or 802.5 [8]), and it would be inefficient to send each individual ATM cell on the air as a single MPDU, due to high overhead. Therefore, the MASCARA protocol uses the concept of a “cell train”, which is a sequence of ATM cells sent as the payload of a MPDU. More precisely, each MPDU consists of a MPDU header, followed by a MPDU payload containing ATM cells generated by the same MT/AP (Figure 5). The time required by the physical layer to initiate a MPDU transmission (referred to as physical overhead) plus the time needed to send the MPDU header is equal to one time slot, whose duration is defined as the time required to transmit an ATM cell. This way it is possible to follow the slot-based timing structure, whatever the number of transmitted cells contained in a MPDU (Figure 5).



**Figure 5: MPDU structure along the time slot scale**

Figure 6 sums up the TDMA frame structure.

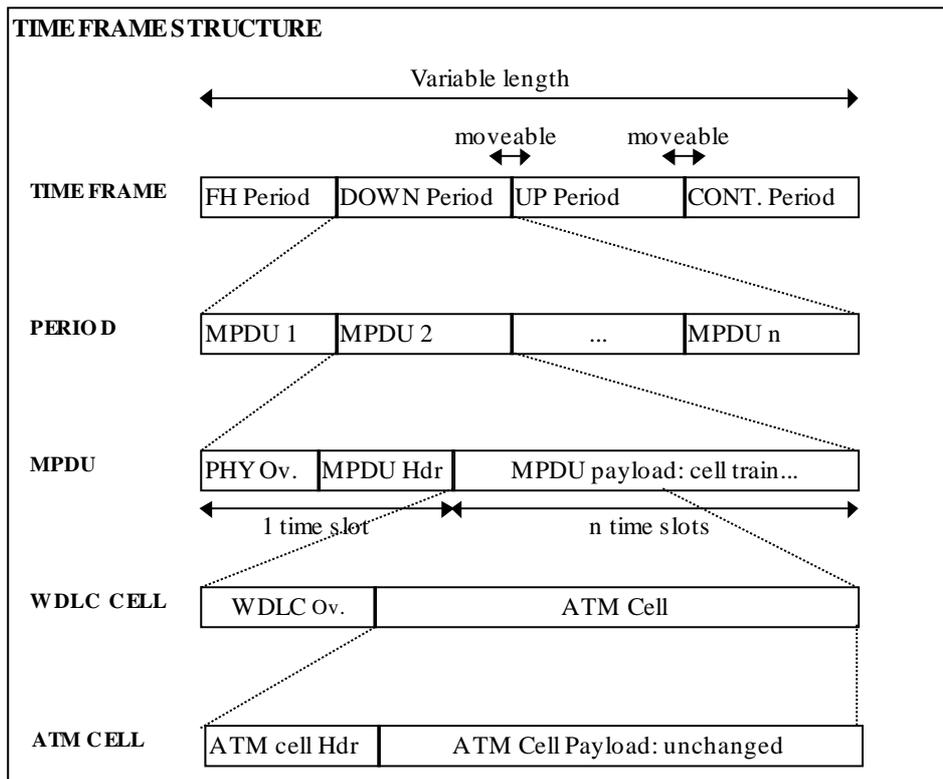


Figure 6: Time Frame Structure

### 3. Scheduling Algorithm for the Master Scheduler

#### 3.1 Scheduling Requirements

Scheduling in fixed ATM networks, together with call admission control (CAC) and resource allocation, can be used to guarantee the satisfaction of different QoS requirements for a wide range of traffic types. CAC and resource allocation are involved in the sharing of bandwidth by different connections, based on traffic characteristics and QoS requirements, and aim at high channel utilization and QoS guarantees. These two objectives are often conflicting, because usually high utilization implies temporary inability of QoS satisfaction. On the other hand, traffic scheduling is used to offer, among other things,

- statistical multiplexing gain,
- utilization of bandwidth unallocated or allocated to idle connections,
- declared and real traffic consistency,
- maintenance of traffic characteristics of the connections,
- QoS requirements satisfaction, especially related to delay and loss, and
- fair treatment of all connections [9].

A wireless ATM network is expected to offer services similar to those of a fixed ATM network. In the fixed part, scheduling and CAC techniques, introduced for fixed ATM, can be used to offer QoS guarantees. However, if the radio part does not offer the same functionality, the QoS seen by the end user will be compromised. Until now, MAC protocols for wireless networks had no or limited scheduling capability, and

were mainly focused on voice and messaging-type data applications [10] (e.g., [11], [12]). In MASCARA however, an advanced scheduling mechanism is required to decide on the allocation of the slots of each frame. Due to the lack of resources, compared to fixed links, bandwidth utilization and statistical multiplexing gain are important objectives of such a mechanism. Nevertheless, these objectives must not violate fairness and QoS guarantees.

In the environment under study, an arbitrary order of slot allocation, in accordance with some properties of the MASCARA protocol, such as uplink/downlink separation and cell train construction, can alter the traffic pattern of a connection. This may result in violation of the contractual values of QoS and traffic characteristics, such as peak cell rate (PCR), cell delay tolerance (CDT), and cell delay variation tolerance (CDVT), and cause discarding of ATM cells deeper in the network, or late arrival at the receiver. The traffic class that is the most affected by this phenomenon is real time VBR (rt-VBR).

The maintenance of contractual values for PCR and CDVT for uplink connections can be controlled with the use of a shaper at the fixed network port in each AP, while for downlink connections maintaining PCR and CDVT in the radio part is less important since this is the last hop of the connection. CDT values for both uplink and downlink can only be controlled by a scheduling mechanism that takes into account the delay constraints of individual connections in the allocation of bandwidth. On the other hand, the scheduler in MASCARA should also try to construct cell trains to the extent possible, and keep the uplink and downlink periods separate within a frame.

### ***3.2 The Proposed Algorithm***

In this section, we describe the proposed algorithm for the Master Scheduler, which schedules transmissions over the radio interface, based on the priority class, the contractual characteristics, and the delay constraints of each connection, as indicated by its name: **P**rioritized **R**egulated **A**llocation **D**elay **O**riented **S**cheduling (PRADOS). In the text below, we refer to the Master Scheduler as simply “the scheduler”.

By the beginning of each frame, the scheduler must determine the MT that each slot will be allocated to, and formulate the slot map, according to short term requests, traffic characteristics, and QoS agreements. As already mentioned, uplink requests are piggybacked into the data MPDUs, while downlink requests are immediately derived by the ATM cells arriving at the AP. Since both uplink and downlink share the same radio channel and traffic is usually unbalanced, a single slot allocation mechanism should be used, handling both directions the same way. The only difference between the two directions is that all downlink allocations are performed before all uplink allocations according to the time frame structure (Figure 6).

PRADOS combines priorities with a leaky bucket traffic regulator [13]. A priority is introduced for each connection, based on its service class:

<b>Priority number</b>	<b>Service class</b>
5	CBR (Constant Bit Rate)
4	rt-VBR (real time-Variable Bit Rate)
3	nrt-VBR (non real time-Variable Bit Rate)
2	ABR (Available Bit Rate)
1	UBR (Unspecified Bit Rate)

The service classes are defined in [14]. The greater the priority number, the higher the priority of a connection. Additionally, a token pool, located at the AP, is introduced for each connection. Tokens are generated at a fixed rate equal to the mean cell rate, and the size of the pool is equal to the “burst size” of the connection [14]. The burst size depends on the characteristics of each connection, and is the maximum number of cells that can be transmitted with rate greater than the declared mean. For every slot allocated to a connection, a token is removed from the corresponding pool. In this way, at any instance of time, the state of each token pool gives an indication of the declared bandwidth that the corresponding connection has consumed.

The priority leaky-bucket algorithm works as follows. Starting from priority 5 (CBR), and down to priority 2 (ABR), the scheduler satisfies requests of (uplink and downlink) connections belonging to each service class, as long as tokens are available. UBR connections have no guaranteed bandwidth, thus no token pools are maintained for them. For connections whose traffic can be anticipated by the scheduler, such as CBR, there is no need for transmission of requests from the MT. Therefore, for such connections, the MT does not transmit requests and the scheduler allocates slots to such connections on the basis of anticipated requests (in other words, the scheduler generates “imaginary” requests for the MT). For the purposes of describing the operation of the scheduler, real and imaginary requests on the uplink, as well as ATM cell arrivals for the downlink will be referred to and treated as “requests”.

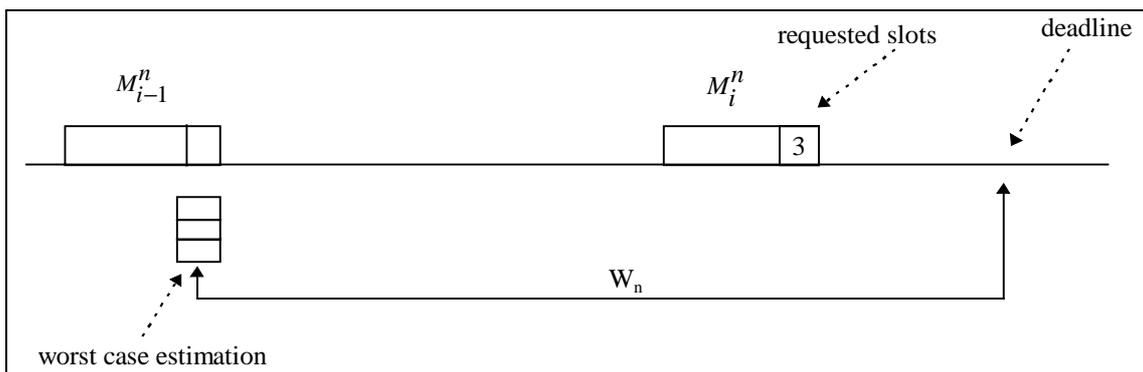
At every priority class, it is very probable to have more than one connections requesting slots. In that case, the scheduler gradually allocates one slot at a time to the connection (or connections) which possesses the most tokens (i.e., highest token variable), decreasing the token variable by one. The rationale is that the connection with the most tokens has consumed less bandwidth than declared, and thus has higher priority for getting slots allocated. When the satisfaction of “conforming” requests is completed, and if there are still available slots, the scheduler tries to satisfy “exceeding” requests. To avoid possible later congestion, these excess uplink cells will have the cell loss priority (CLP) bit set to zero.

At this state, the token variables of all connections requesting slots are less than or equal to zero. The scheduler follows the same procedure as before, starting from priority 5 (CBR), down to priority 1 (UBR). If more than one connections belonging to the same priority class request slots, one slot at a time is allocated to the connection with the highest token variable. Since this is excess traffic, decreasing will result in negative values for the token variables. The procedure stops when all requests are satisfied or all available

slots are allocated. An early version of this scheduler, together with performance evaluation can be found in [15].

What is not specified with the above algorithm is the exact order of allocation of slots per MT. To do that, the scheduler should consider the traffic and QoS characteristics of the connections. As already mentioned, an important QoS parameter to be maintained in the air interface is CDT. Scheduling in the radio interface should enforce the wireless hop CDT of each connection, and allocate slots so that the fraction of ATM cells whose delay exceeds this CDT is minimized. The wireless hop CDT can be evaluated by decomposing end-to-end CDT into CDT for each hop of the ATM connection path.

It is clear that a delay-oriented scheduler requires knowledge of the arrival time of ATM cells in the output queues of the AP and MTs. Due to the location of the scheduler, the arrival time of downlink ATM cells can be directly logged and used in the scheduling algorithm. But this is not the case for the uplink ATM cells. In MASCARA, uplink requests are either piggybacked in the data or sent through control MPDUs, or generated by the scheduler (e.g., for CBR connections). As already mentioned, this information is in the form of the number of ATM cells waiting to be transmitted per connection. In this way, the exact creation time of the corresponding ATM cells is not specified, and should be estimated. For piggybacked requests, let us denote by  $M_i^n$  the  $i$ -th transmitted MPDU of connection  $C_n$ . If it is assumed that requests in  $M_i^n$  correspond to cells created in the interval  $[M_{i-1}^n, M_i^n]$ , then a worst case estimation for the creation time of these cells is  $M_{i-1}^n$  (Figure 7). The estimated deadline, i.e., the latest time up to which these cells should be transmitted is  $W_n$  time units after  $M_{i-1}^n$ , where  $W_n$  is the wireless hop CDT for connection  $C_n$ . For imaginary requests, their number, arrival time and deadline time are derived from the ATM cell rate and from the last regularly scheduled ATM cell.



**Figure 7: Estimated deadline for uplink ATM cells**

The deadline of an ATM cell is not an absolute threshold but rather an indication of how quickly an ATM cell should be scheduled for transmission. The scheduler should try to schedule ATM cells before their deadlines, but if that is not possible, some specified maximum extra delay may be allowed.

PRADOS is based on the intuitive idea that, in order to maximize the fraction of ATM cells that are transmitted before their deadlines, each ATM cell is initially scheduled for transmission as close to its deadline as possible. This idea was first proposed for fixed ATM networks in [16]. To attain high utilization of the radio channel, the algorithm is “work-conserving”, meaning that “*the channel never stays idle as long as there are ATM cells requesting transmission*” [17]. Consequently, the final transmission time of an ATM cell will be the earliest possible given the ATM cell’s initial ordering. This way the deadline of an ATM cell in effect determines the transmission order of that ATM cell with respect to the others.

The construction of cell trains and the separation of uplink and downlink periods inside each frame play an important role in the operation of PRADOS. In that sense, cell trains are constructed gradually, according to the leaky bucket algorithm, and ordered according to their deadlines. The deadline of a cell train is considered equal to the deadline of its first ATM cell. An ATM cell is attached at the end of the corresponding cell train if this does not cause deadline violation of the existing cell trains.

Below we describe how PRADOS takes into account the deadlines in the radio interface. For simplicity, all times are measured in slot time units. We denote by  $c_n(i)$  the  $i$ -th ATM cell of connection  $C_n$ . When an uplink or downlink cell  $c_n(i)$  arrives at the MASCARA layer of a MT or AP it can be represented by the parameters given below. For the purposes of the algorithm, we assume that, at the beginning of each frame, the slots following or preceding the FH period are numbered, starting from 1 and -1 respectively, and the parameters are adjusted accordingly.

- $a(c_n(i))$  = the arrival time of  $c_n(i)$ . For downlink ATM cells, this is the actual arrival time, while for uplink ATM cells it is estimated.
- $m(c_n(i))$  = the maximum waiting time that  $c_n(i)$  can wait before being transmitted in the radio interface. Here we assume that  $m(c_n(i))=W_n$ , the same for all ATM cells of connection  $C_n$ , although the algorithm can also be used for different  $m(c_n(i))$ ’s.
- $d(c_n(i)) = a(c_n(i))+m(c_n(i))$ , the latest time by which  $c_n(i)$  can be transmitted without missing its deadline.

Assume that, during a frame, a number of requests have been issued to the scheduler. At the end of the operation of the algorithm, the exact position of the slots allocated to each MT in the next frame should be specified. We define the following notation:

- $D_n = \min\{d(c_n(i)) : c_n(i) \text{ is requesting slot in the current frame}\}$ , i.e.,  $D_n$  is the earliest deadline of all ATM cells of  $C_n$  requesting slots in the current frame,
- $F_n = \begin{cases} 0, & \text{if no slot is allocated to connection } C_n \text{ in this frame} \\ \text{the first slot allocated to connection } C_n, & \text{otherwise} \end{cases}$
- $L_n = \begin{cases} 0, & \text{if no slot is allocated to connection } C_n \text{ in this frame} \\ \text{the last slot allocated to connection } C_n, & \text{otherwise} \end{cases}$
- $O(x) = \begin{cases} 0, & \text{if slot } x \text{ is empty} \\ n : F_n \leq x \leq L_n, & \text{otherwise} \end{cases}$  (i.e.,  $O(x)$  is the identifier of the connection that slot  $x$  belongs to)

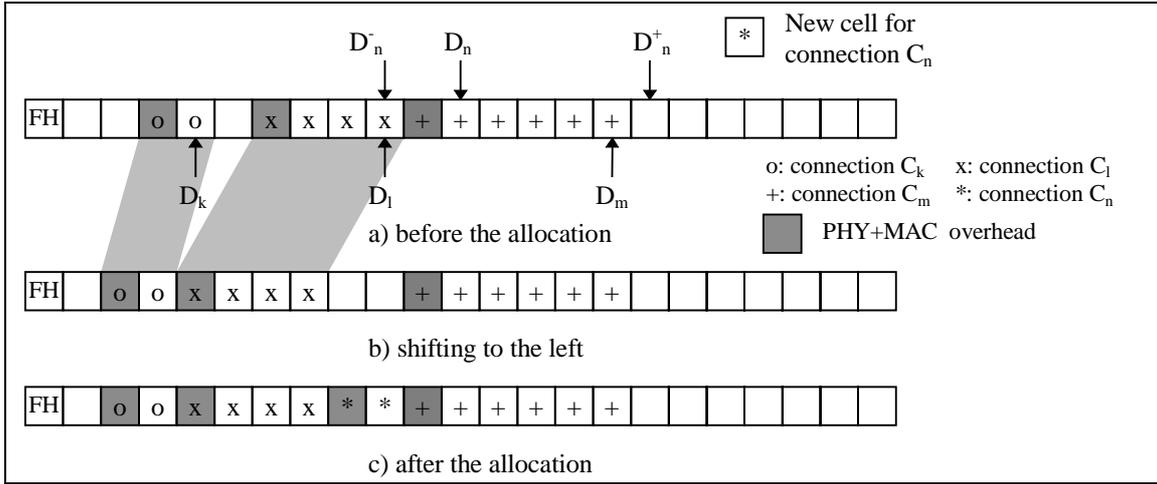
- $D_n^- = \begin{cases} D_n, & \text{if } O(D_n) = 0 \\ F_{O(D_n)} - 1, & \text{if } O(D_n) \neq 0 \end{cases}$  (i.e.,  $D_n^-$  is the slot preceding the first slot allocated to the connection owning  $D_n$ )
- $D_n^+ = \begin{cases} D_n, & \text{if } O(D_n) = 0 \\ L_{O(D_n)} + 1, & \text{if } O(D_n) \neq 0 \end{cases}$  (i.e.,  $D_n^+$  is the slot following the last slot allocated to the connection owning  $D_n$ )
- $E_n = \sum_{i=1}^{D_n} (O(i) = 0)$  (i.e.,  $E_n$  is the number of free slots in the interval  $[1, D_n]$ )
- $\Delta_n$  is the maximal allowed *extra* delay, for allocating slots to connection  $C_n$ , after the deadline  $D_n$ .
- $N(x)$  is the first empty slot after slot  $x$ .
- $\phi$  is the length of the “MPDU overhead”: for each MPDU transmitted over the air, a number of  $\phi$  slots must be reserved for transmission of the physical and MPDU headers. In the WAND system, this overhead consumes only one slot ( $\phi=1$ ), but for keeping the description general, we will not give any constant value to this overhead. Nevertheless the value 1 is assumed in the following illustrative figures to keep them easy to understand.
- $P$  is the length of the “period overhead”: at the boundary between the DOWN and UP periods, the RF modem must switch between transmit and receive mode, an operation which is assumed to last for  $P$  slots. In the WAND system, this overhead consumes only one slot ( $P=1$ ), but for keeping the description general, we will not give any constant value to this overhead. Nevertheless the value 1 is assumed in the following illustrative figures to keep them easy to understand.

PRADOS operates in frames, and can be divided into three steps.

**Step A:** The scheduler starts satisfying requests according to the leaky-bucket algorithm described above. When a request corresponding to cell  $c_n(i)$  is selected for service, the scheduler tries to allocate as many slots as needed for the transmission of  $c_n(i)$ . We consider two cases:

**Case I:** Request for cell  $c_n(i)$  is the first serviced request for connection  $C_n$  in the current frame. In this case, if slots  $[D_n^-, D_n]$  are all empty, the scheduler allocates them to the current request. Otherwise, we distinguish two subcases:

**Case Ia:**  $E_n > \phi$ , i.e., there are at least  $\phi+1$  empty slots in the interval  $[1, D_n]$ . From the definition of  $D_n^-$  and the operation of the scheduler, which results in no more than one cell train per connection in each frame, it is derived that all the  $E_n$  empty slots are in the interval  $[1, D_n^-]$ . In this case, the algorithm has to free  $\phi+1$  positions, as close to  $D_n$  as possible, without splitting any existing cell train, to place the new ATM cell. These positions are  $D_n^- - \phi$  up to  $D_n^-$ . For instance, when  $\phi$  is equal to 1, all allocations between the two last empty slots before  $D_n^-$  shift one position to the left and all allocations between the last empty slot and  $D_n^-$  shift two positions to the left to leave  $D_n^- - 1$  and  $D_n^-$  empty for the new allocation (Figure 8). Since the allocations move to the left, none of them exceeds its deadline.

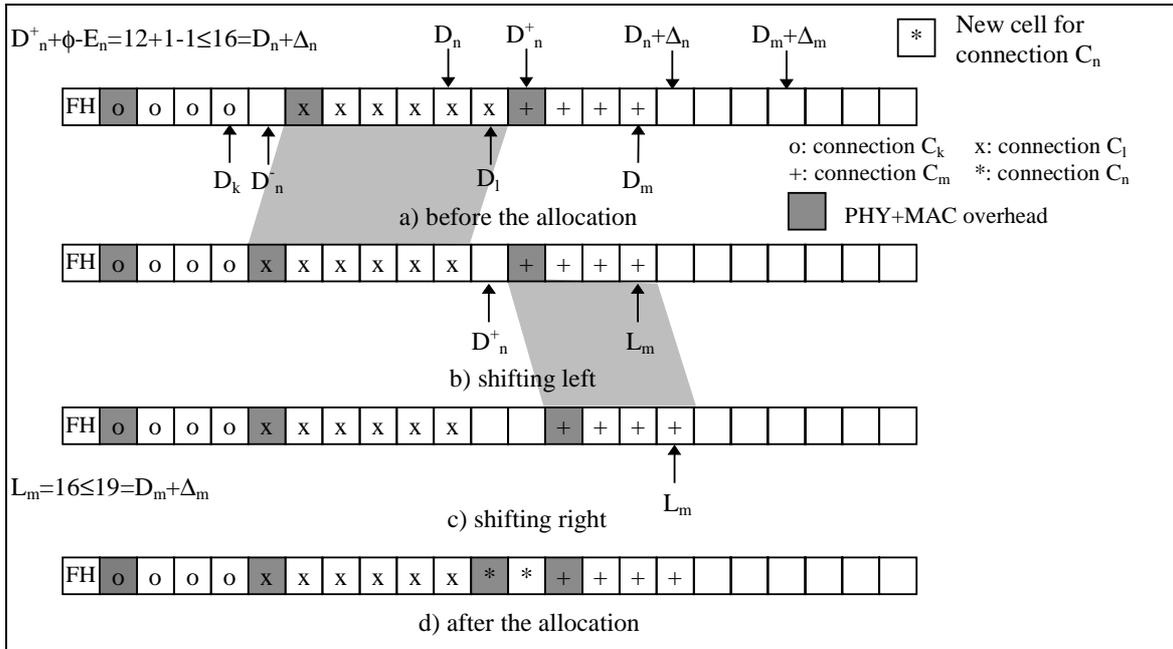


**Figure 8: Allocation when there are at least  $\phi+1$  free slots before  $D_n$**

*Case 1b:*  $E_n \leq \phi$  (i.e., there are  $\phi$  or less empty slots in the interval  $[1, D_n]$ ). In this case, there is not enough available “free space” prior to the deadline to allocate slots for connection  $C_n$ . Therefore the allocation can only be done on the “right side” of the deadline, while ensuring that the introduced delay does not cause exceeding of the “extended deadline”. The extended deadline of a connection  $C_i$  is defined as the sum of  $D_i$  plus  $\Delta_i$ . The proposed approach is first to fill the  $E_n$  empty slots before the deadline  $D_n$  by shifting to the left the allocations before  $D_n^+$  and then start the allocation on the new position of  $D_n^+$ , ensuring that shifting to the right existing allocations, if needed, does not violate their extended deadlines. Therefore the strategy consists of the following steps.

1. Verify that  $D_n^+ + \phi - E_n \leq D_n + \Delta_n$  to ensure that the extended deadline of  $C_n$  is not exceeded. If found false, no allocation is performed.
2. For all connections  $C_i$ , that have allocations on the right of  $D_n$ , and need to be shifted to the right to make space for the new allocation, verify that after the shifting  $L_i \leq D_i + \Delta_i$  for all connections. If this is not the case even for one connection, no allocation is performed. Otherwise:
3. Allocations up to  $D_n^+ - 1$  are moved to the left to fill all the empty  $E_n$  slots, while allocations after  $D_n^+ - 1$  are shifted to the right, to make  $\phi+1$  slots available for the new allocation.
4. The allocation is performed starting at  $D_n^+$  (note that the value of  $D_n^+$  after step 3 differs from the value before step 3, as it has been decreased by  $E_n$ ).

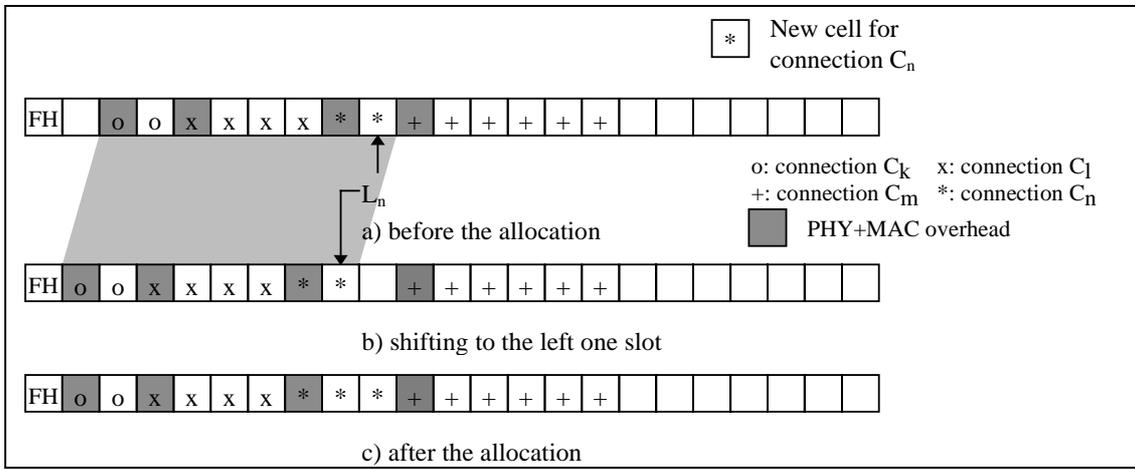
An example is shown in Figure 9.



**Figure 9: Allocation when there is no more than  $\phi$  slots before  $D_n$**

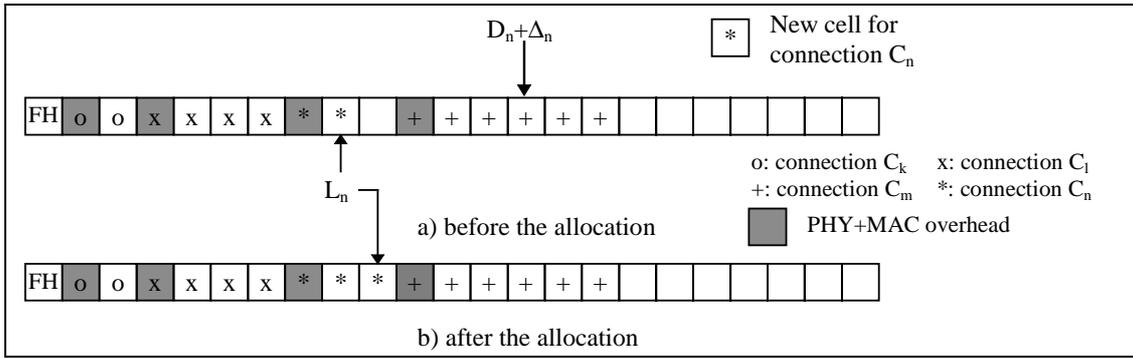
**Case 2:** Request for packet  $c_n(i)$  is not the first serviced request for connection  $C_n$  in the current frame. We distinguish three sub-cases.

**Case 2a:** There is at least one empty slot in  $[1, L_n]$ . In this case, all allocations between the last empty slot before  $L_n$  and  $L_n$  shift one position to the left to leave  $L_n$  empty for the new allocation (Figure 10). Moving to the left, does not violate any deadlines.



**Figure 10: Allocation when there is at least one free slot before  $L_n$**

**Case 2b:** There are no empty slots in  $[1, L_n]$  and the slot  $L_n+1$  is empty. In this case, the allocation is performed in slot  $L_n+1$ , if  $L_n+1 \leq D_n + \Delta_n$  (Figure 11), otherwise it is not performed.

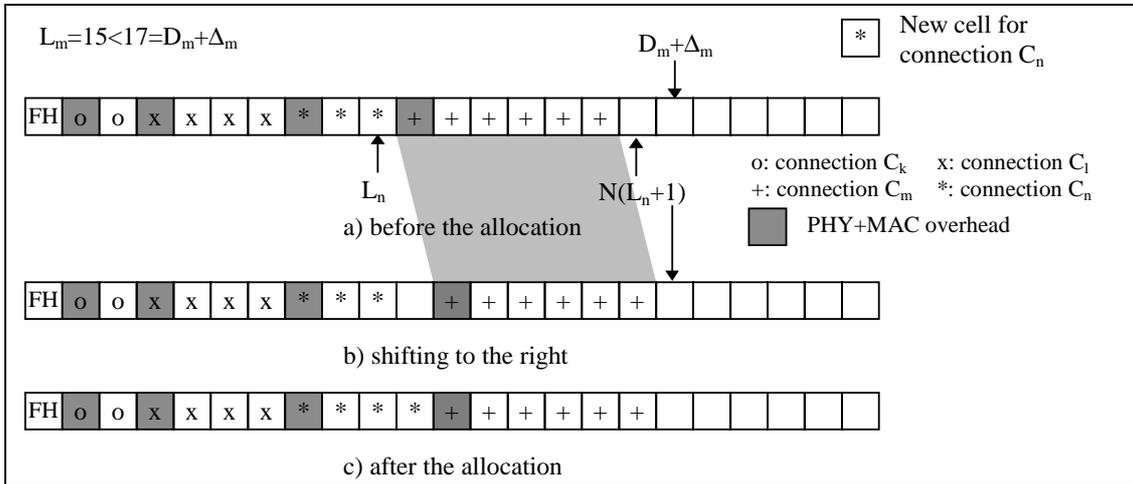


**Figure 11: Allocation when there is no free slot before  $L_n$  and slot  $L_n+1$  is free**

*Case 2c:* There are no empty slots in  $[1, L_n]$  and the slot  $L_n+1$  is not empty. In this case, the allocation can only be performed in slot  $L_n+1$  if connection  $C_n$  and all the connections occupying the slots up to the first empty slot have not yet reached their extended deadline. The strategy consists of the following steps (see Figure 12):

1. If  $L_n = D_n + \Delta_n$  then no allocation is performed.
2. Check that the following relation is true, for all connections  $C_i$  occupying slots in the interval  $[L_n+1, N(L_n+1)-1]$ :  

$$L_i < D_i + \Delta_i$$
 If found false, even for only one connection, then no allocation is performed. Otherwise:
3. Shift to the right by one position the allocations in the interval  $[L_n+1, N(L_n+1)-1]$  (this step frees the slot  $L_n+1$ ).
4. The allocation is performed in slot  $L_n+1$ .



**Figure 12: Allocation when there is no free slot before  $L_n$  and  $L_n+1$  is not free**

**Step B:** The purpose of this second step is to build the DOWN interval of the MAC frame. This operation consists of three sub-steps:

1. When all requests have been processed, the scheduler packs as close to the beginning of the frame as possible, all allocations between the beginning of the frame and the first slot allocated to an uplink connection (clearly all these allocations correspond to downlink connections) (Figure 13 step b).

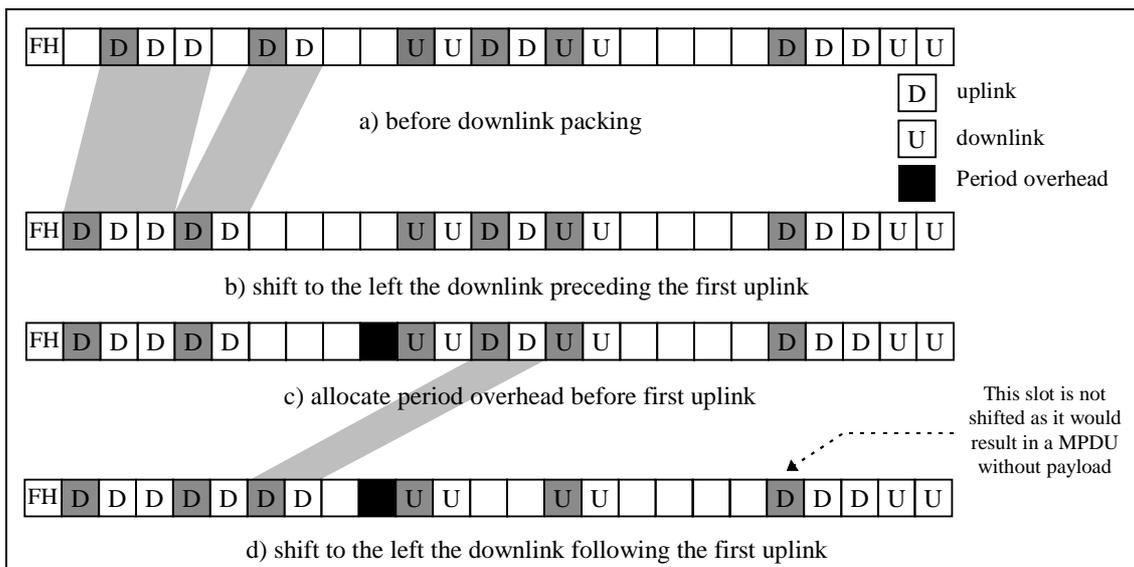
2. As an output of the former sub-step, some slots may be left empty before the first uplink slot. If this number of empty slots is equal to  $E$ , then a comparison is made between  $E$  and  $P$ :

- if  $E \geq P$ , then the last  $P$  slots before the first uplink slot are allocated for the so-called “period overhead” (Figure 13 step c).
- if  $E < P$ , then the last  $P-E$  slots of the last downlink cell train preceding the first uplink slots are re-allocated to the period overhead. If the resulting slot train is less than  $\phi+1$  slots long, then this whole slot train must be deallocated (to avoid keeping an incomplete MPDU).

**Note:** Stealing some slots from the last MPDU may not be the optimal strategy, depending on the traffic class of the corresponding connection. A better (but more complex) approach would be to steal slots from the connection which is the less sensitive to cell loss.

3. In the space left empty between the last downlink allocation and the period overhead, the algorithm tries to pack as many downlink allocations as possible by moving them to the left (Figure 13 step d).

**Note:** When allocations are shifted to the left, a given cell train may be broken if there are not enough free slots available for the whole cell train. In this case, the procedure must ensure that the number of shifted allocations is at least equal to  $\phi+1$  (otherwise it would result into a MPDU with an empty payload or even with an incomplete header).



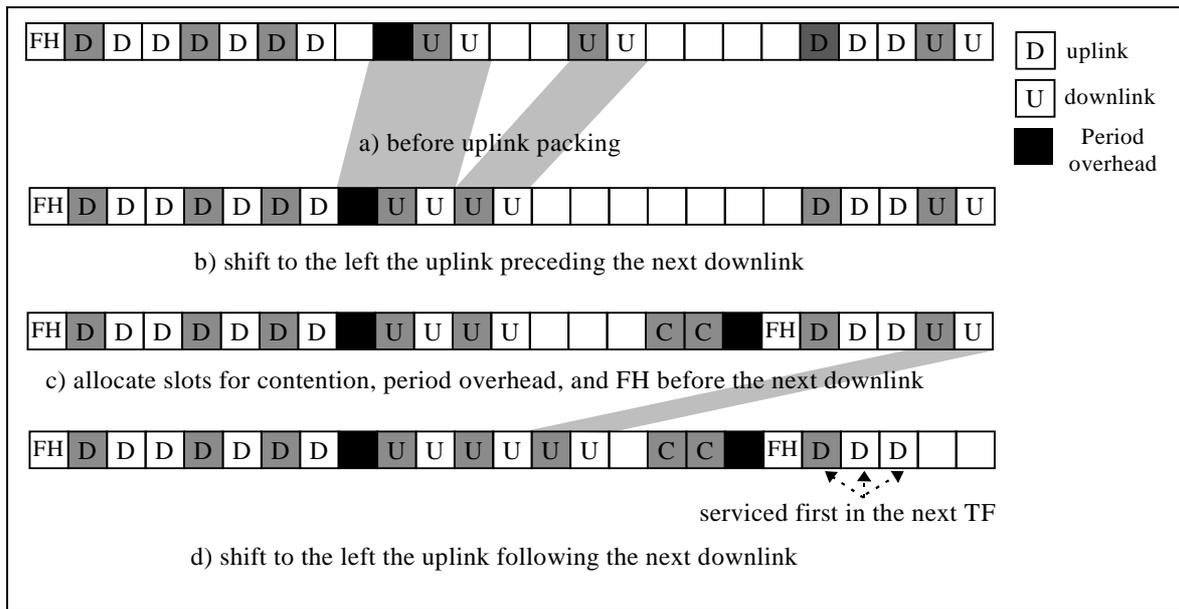
**Figure 13: Packing of downlink allocations**

**Step C:** The purpose of this step is to build the UP interval of the MAC frame. The operation is analogous to step B, except for the following differences:

1. Packing of uplink is performed between the end of the DOWN interval, as produced from step B, and the next slot allocated to a downlink connection.
2.  $P$  in sub-step 2 of step B is replaced by  $LC+P+LFH$ , where  $LC$  is the length of the contention period and  $LFH$  is the length of the FH of the next frame.

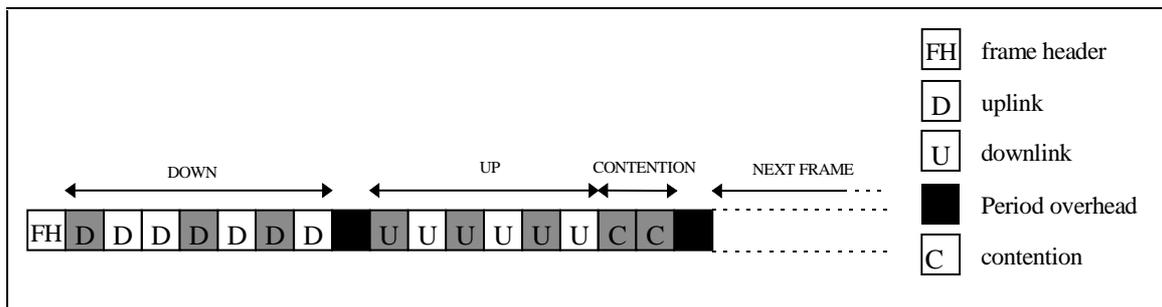
An example is shown in Figure 14. In this figure, the contention period is limited to the minimum for drawing purposes. Concerning allocations that cannot be packed during steps B and C, due to insufficient

space in the DOWN and UP intervals, the corresponding requests are serviced before new requests in the next frame, according to the same algorithm (Figure 14 step d).



**Figure 14: Packing of uplink allocations**

At the end of the procedure, the CONTENTION period is packed right next to the UP period (Figure 15). Packing preserves the “work-conserving” property that was mentioned before, since no slots are left empty.



**Figure 15: Final frame structure**

### 3.3 Discussion on potential enhancements

Three different aspects of the presented scheduling algorithm can clearly be enhanced, at the cost of some complexity increase: the estimation of the deadline for uplink connections, the arrangement of cell trains to form the final MAC frame, and the evaluation of the contention period length.

**Uplink connection deadline:** As illustrated by Figure 7, the deadline for uplink ATM cells is estimated according to the date of reception of the MPDU preceding the reception of a reservation request. This estimator is quite conservative: the real deadline may be later than estimated, potentially resulting in the absence of allocation for this connection, even if the real deadline would have not been exceeded. The proposed enhancement is to carry time stamp information along with the reservation request, so that the scheduler knows exactly when the uplink ATM cell was received for transmission in the MT. Thanks to this

timing information, the estimator, as described in Figure 7 is no longer needed, and the algorithm can operate based on the real arrival times of both uplink and downlink ATM cells.

**Cell train arrangement in the MAC time frame:** As described in steps B and C, the final arrangement of the cell train must respect the frame structure where the downlink transmission always precedes the uplink transmission. This constraint may result in postponing the transmission of some cell trains to a subsequent frame, for instance, in the case where the output of step A is a sequence of alternating DOWN and UP cell trains. As there is some overhead associated with each new frame (the FH period during which the FH MPDU must be built and sent by the AP and then received and analysed by all the MTs), this may lead to situations where a given ATM cell will finally not be allocated for deadline reasons. One way to solve this problem is to allow interleaving of the UP and DOWN periods during the frame, taking into account both the overhead associated to the turn-around time of the physical channel (P slots) and the overhead associated to the frame (the FH period whose duration is LFH slots). The idea, which will not be further investigated here, is to allow period interleaving if and only if it allows to transmit a cell train that would have been aged out if its transmission had been postponed for the next frame. For connection  $C_n$ , this condition is fulfilled if:

$$L_n + P < D_n + \Delta_n < L_n + LFH$$

**Contention Period:** As already mentioned, the contention period is used for unexpected control traffic from the MTs to the AP (for example, when an MT powers up and wants to signal its presence to the AP), or for sending allocation requests by MTs that have no allocated slot to piggyback their requests for data transmission. The length of the contention period is variable to allow adaptability to traffic conditions. The algorithm that determines the contention period length for each frame is part of the random access protocol used for accessing the contention period. In the design of such an algorithm, the objective is to use available information on current traffic conditions and the state of the ongoing collision resolution process (if any) to keep the collision probability and collision resolution delays at the desirable level while using the shortest contention period length possible. With regards to the overall random access protocol used for accessing the contention period, besides the simple slotted-ALOHA, more sophisticated protocols (e.g., tree or stack type [7]) may be used for improved collision resolution performance. Such a protocol, based on a modified stack random access algorithm, has been presented in [18]. The detailed design of the random access protocol used for the contention period in WAND is beyond the scope of this paper, and will be reported in a future publication.

## 4. Scheduling Algorithm Performance Results

In this section, we give simulation results on the performance of PRADOS. We also consider and compare it with a simpler algorithm, in which the delay constraints of the different connections are not taken into account.

The simulation models were built using the OPNET tool [19]. The models use the frame structure defined in section 2, except for the following simplification: the frame header and the contention period have constant lengths. The incoming traffic consists of identical VBR connections. To model the traffic for these connections, we used independent discrete-time batch Markov arrival processes (D-BMAP). D-BMAP, proposed in [20], is the discrete-time analogue of the batch Markov arrival process, introduced by Lucantoni in [21]. The number of connections was equally divided between uplink and downlink. For odd numbers, the extra connection was considered uplink. The system parameters used in the simulations are shown in Table 1.

Channel Characteristics	Connections Characteristics	Overheads
channel capacity = 19.2 Mbps slot duration = $22 \cdot 10^{-6}$ (time needed for one ATM cell)	mean rate = 850 kbps deviation ( $\sigma$ ) = 310 kbps wireless hop CDT = 4.4 msec max. extra delay = 0.44 msec	cell train overhead = 1 slot period overhead = 1 slot frame header length = 5 slots contention period = 6 slots

**Table 1: Simulation parameters**

Finally, to focus on the performance of the scheduling algorithm, we have assumed that allocation requests sent over the contention period are successful in their first transmission attempt, i.e., no collisions.

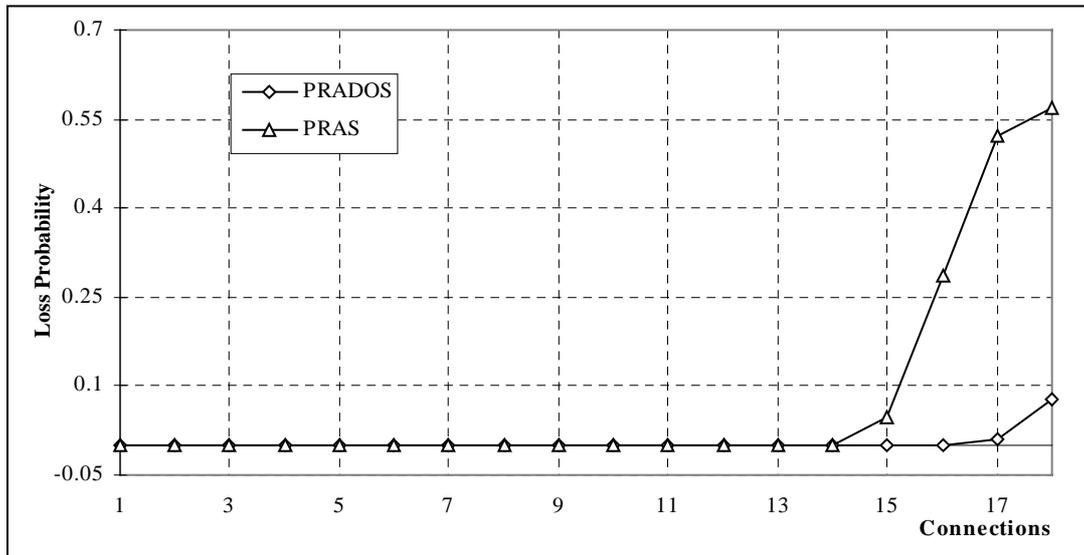
PRADOS was compared with a simpler algorithm called **P**rioritized **R**egulated **A**llocation **S**cheduling (PRAS), which uses the virtual leaky bucket mechanism described in section 3, but does not take into account cell transmission deadlines. More specifically, in PRAS, the first allocation of each connection is made at the end of the downlink or uplink period, depending on the direction of the connection. For example, the first allocation of a downlink connection is placed after the last downlink cell train by shifting all the uplink allocations to the right. Allocations other than the first one for a connection are placed at the end of the connection's cell train.

The performance measures obtained from the simulation model are the following:

- $P_{\text{loss}}$ : This is the ratio of the expired ATM cells (i.e., cells that experience delays higher than the maximum allowed,  $W+\Delta$ ) over the total number of cells generated, and corresponds to the loss probability. Note that, in PRADOS, expired cells are not transmitted, whereas in PRAS expired cells are transmitted over the radio interface.
- Utilization: This is the fraction of the used channel capacity (i.e., the fraction of the total number of slots that are used to carry data traffic).
- Throughput: This is the fraction of the channel capacity used for the transmission of “useful” (i.e., not expired) ATM cells. Clearly,  $\text{Throughput} = \text{Utilization} \cdot (1 - P_{\text{loss}})$
- Mean delay: The mean delay experienced by the non-expired ATM cells. The delay of a cell is defined as the time from generation until transmission completion.
- Mean frame length.

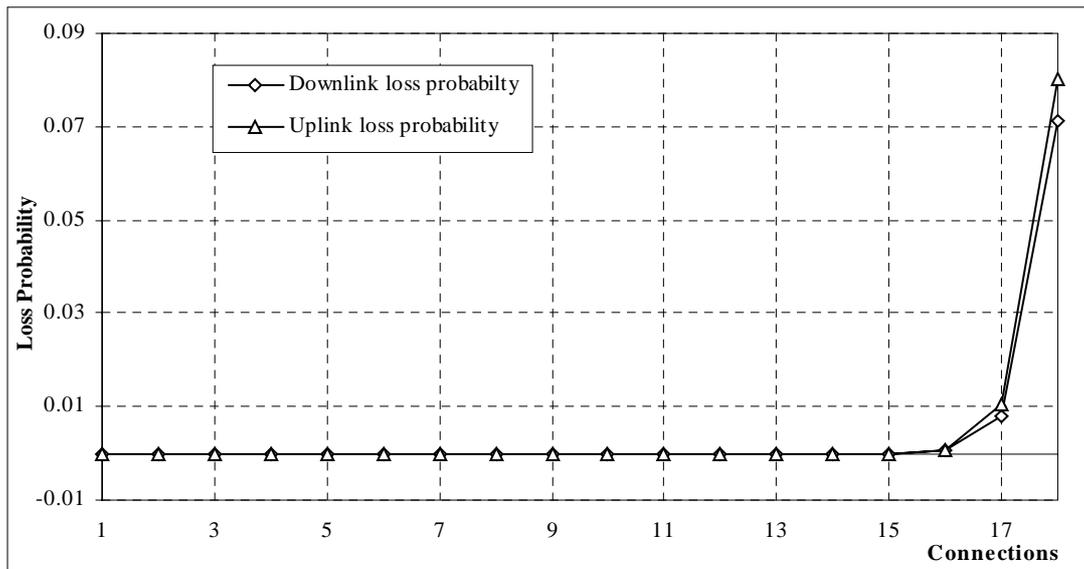
- Mean cell train length: The mean number of slots is a cell train.

Figure 16 plots  $P_{\text{loss}}$  versus the number of connections. Observe that the loss probability remains almost equal to zero for low traffic, i.e., for as many as 14 connections, in both algorithms. However, for heavy traffic the loss probability of PRADOS remains comparatively low, while the loss probability for PRAS follows a more abrupt curve. This can be explained by the waste of network resources due to the transmission of expired ATM cells when PRAS is used.



**Figure 16: Loss probability**

An important objective of the scheduling algorithm is fairness in handling uplink and downlink connections. As shown in Figure 17, the loss probability of uplink and downlink connections is almost identical, indicating fair treatment of uplink and downlink connections by PRADOS.



**Figure 17: Uplink vs downlink loss probability for PRADOS**

Figure 18 gives the mean cell delay (irrespective of direction) versus the number of connections, for both PRADOS and PRAS. As it can be seen from the figure, PRADOS attains better mean delay performance compared to PRAS as the traffic load increases. The reason for this behaviour is twofold: i) since PRADOS transmits only non-expired cells (i.e., the cell delay is upper bounded by  $W+\Delta$ ), the induced cell delay variance is smaller than that under PRAS, and 2) the transmission of expired ATM cells in PRAS wastes resources.

Figures 19 and 20 give the mean cell delay in the downlink and the uplink direction, respectively. Note that, in both algorithms the mean downlink delays are smaller than the uplink ones, which is attributed to the MASCARA frame structure, which places the downlink period before the uplink period. From Figures 19 and 20, we also note that the difference in delays between downlink and uplink ATM cells is smaller under PRADOS compared to PRAS, especially under heavy traffic load.

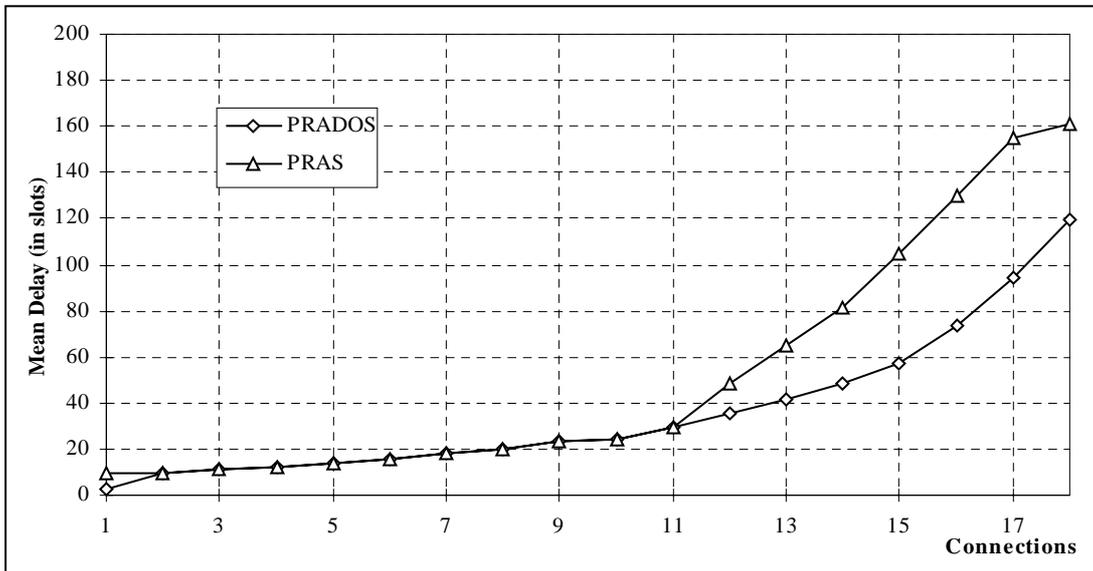


Figure 18: Mean delay

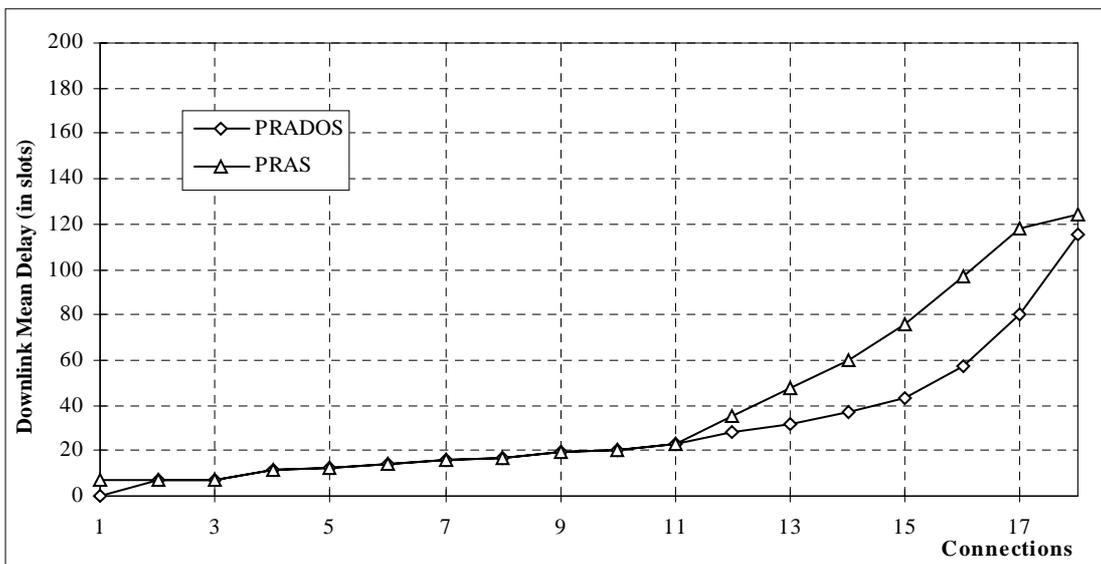
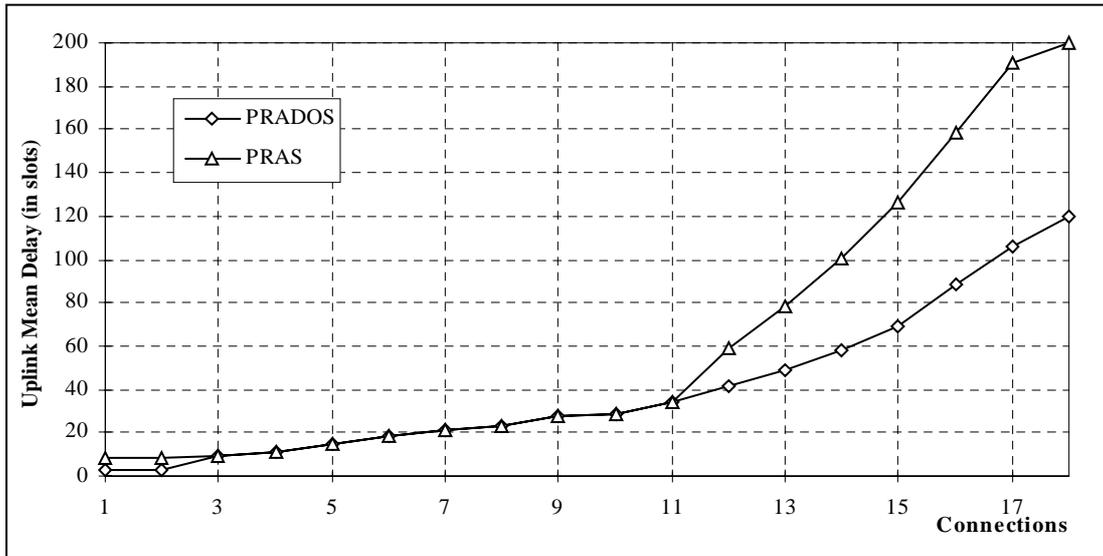
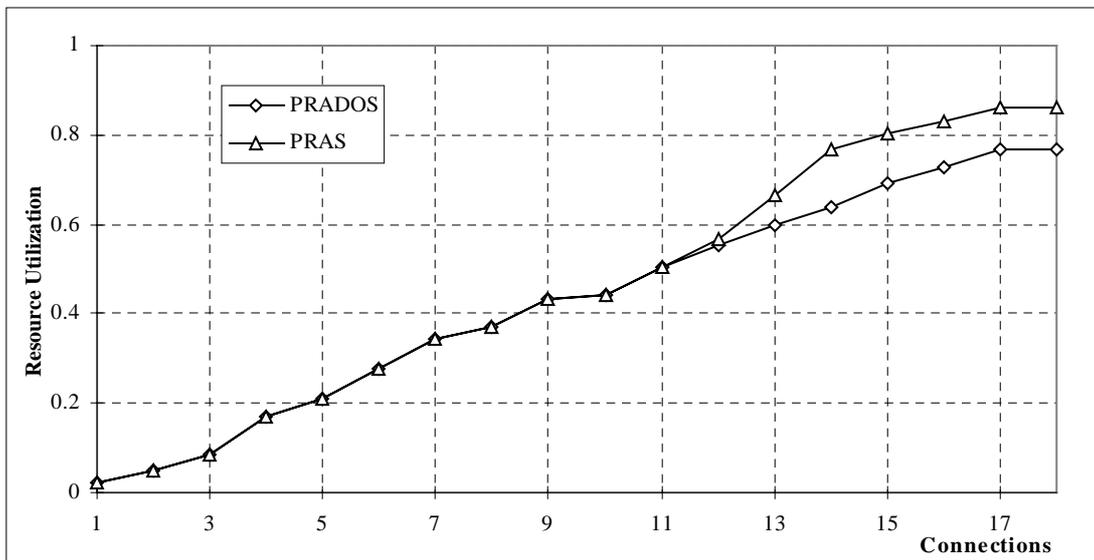


Figure 19: Downlink mean delay

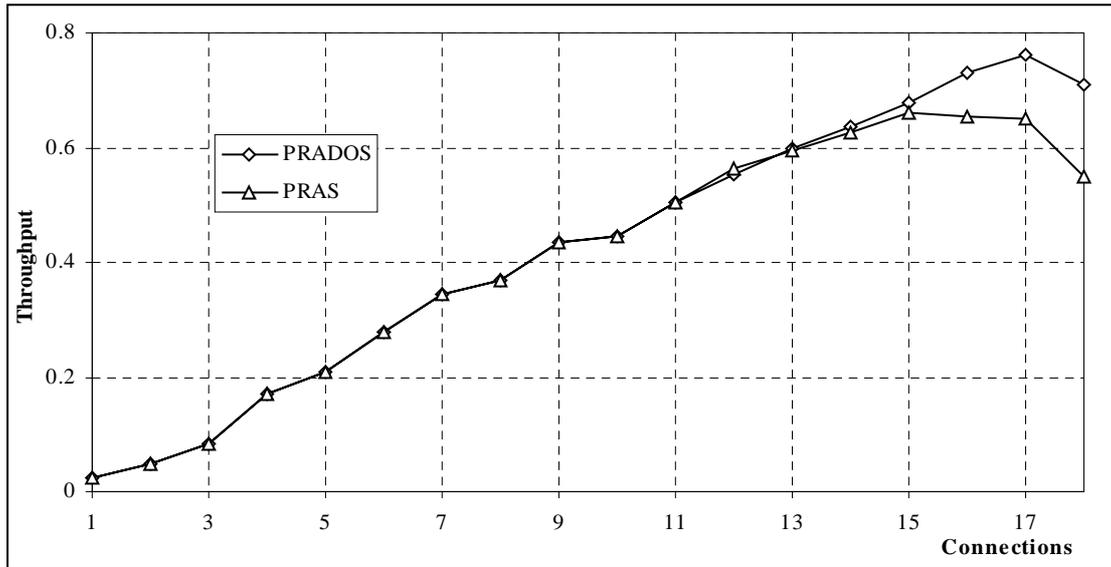


**Figure 20: Uplink mean delay**

As it can be seen in Figure 21, resource utilization seems better in PRAS for heavy traffic; however, a portion of the transmitted ATM cells are actually useless, since they are expired (recall the high loss probability under heavy traffic load, in Figure 16, when PRAS is used). A more interesting measure is the throughput of the two algorithms, which is given in Figure 22. Note that PRADOS attains higher throughput than PRAS under heavy load, reaching a value of 0.75 for 17 connections.

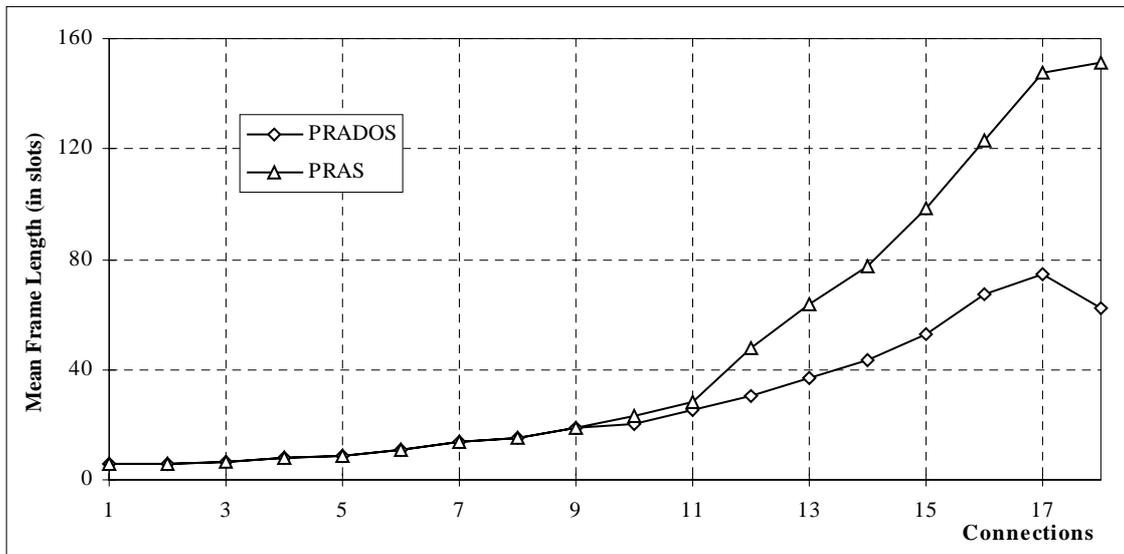


**Figure 21: Resource utilization**

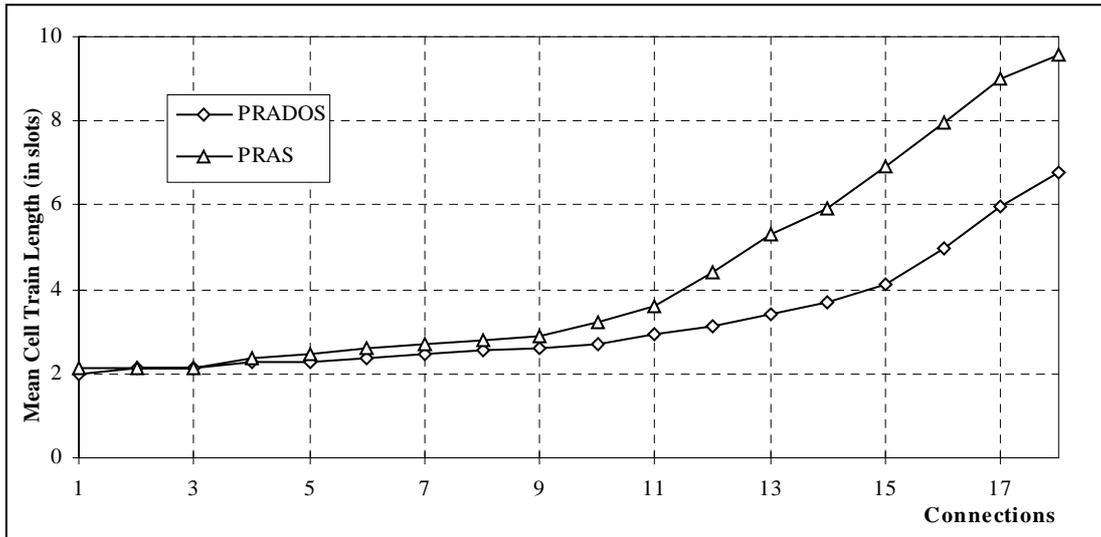


**Figure 22: Throughput**

Finally, Figures 23 and 24 give the mean frame length and the mean cell train length, respectively, for both algorithms. Observe that both lengths have greater values under PRAS, owing to the transmission of expired cells.



**Figure 23: Mean frame length**



**Figure 24: Mean cell train length**

## 5. Conclusion

In this paper, we have given an overview of MASCARA, the MAC protocol designed for ACTS project WAND, and described in detail the traffic scheduling algorithm used in this protocol. MASCARA is a TDMA-based protocol using both reservation and contention for accessing the medium. Most parameters of the protocol, such as time frame length and periods length were left variable and adjustable to traffic conditions, to attain better performance. The scheduling algorithm (PRADOS) focuses on satisfying delay constraints of the various connections. It is based on the intuitive idea that, in order to minimize the fraction of expired ATM cells, each ATM cell should be initially scheduled as close to its deadline as possible.

To evaluate the performance of the scheduling algorithm, we have compared it with a simpler algorithm (PRAS) that does not take into account delay constraints. The obtained simulation results show that the increased complexity of the proposed algorithm is worth while, since the loss and delay performance is significantly improved. Work in progress includes improvement of specific operations of the scheduler, such as deadline estimation and cell train arrangement, as well as dynamic calculation of the contention period length.

## 6. References

- [1] J.D. Gibson, Editor-in-Chief, "The Mobile Communications Handbook", CRC Press, 1996, ISBN 0-8493-8573-3.
- [2] O. Kyas, "ATM Networks", International Thomson Publishing, 1995.
- [3] D.C. Cox, "Wireless Personal Communications: What Is It?", IEEE Personal Communications, Vol. 2, No. 2, pp. 20-35, April 1995.
- [4] H. Hansen et al., "Description of the Handover Algorithm for Wireless ATM Network Demonstrator (WAND)", in Proc. ACTS Mobile Communications Summit, Granada, Spain, Nov. 27-29, 1996.
- [5] G. Pavlidis et al., "Managing a Wireless ATM Network", in Proc. ACTS Mobile Communications Summit, Granada, Spain, Nov. 27-29, 1996.
- [6] D. Bertsekas and R. Gallager, "Data Networks", Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [7] L. Merakos and C. Bisdikian, "Delay Analysis on the n-ary Stack Random-Access Algorithm", IEEE Trans. on Information Theory, Vol. 34, No. 5, pp. 931-942, Sept. 1988.
- [8] P. Fortier, "Handbook of LAN Technology", Second edition, McGraw-Hill Series on Computer Communications, McGraw-Hill, Inc., NY, 1993.
- [9] M.W. Garrett, "A Service Architecture for ATM: From Applications to Signalling", IEEE Network Magazine, Vol. 10, No. 3, pp.6-14, May/June 1996.
- [10] D.F. Bantz and F.J. Bauchot, "Wireless LAN Design Alternatives", IEEE Network Magazine, Vol.8, No.2, March/April 1994.
- [11] S. Nanda et al., "Performance of PRMA: A Packet Voice Protocol for Cellular Systems", IEEE Trans. on Vehicular Technology, Vol. 40, No. 5, pp. 584-598, Aug. 1991.
- [12] F. Lee and L. Merakos, "Voice/Data Channel Access Integration in TDMA Digital Cellular Systems", IEEE Trans. on Vehicular Technology, Vol. 43, No. 4, pp. 986-996, Nov. 1994.
- [13] A.E. Eckberg et al., "Meeting the Challenge: Congestion and Flow Control Strategies for Broadband Information Transport", in Proc. IEEE GLOBECOM'89, pp. 49.3.1-49.3.5, 1989.
- [14] The ATM Forum, "User-Network Interface (UNI) Specification", Version 3.1, September 1989.
- [15] N. Passas and L. Merakos, "A Leaky-Bucket Based Scheduling Technique for Wireless Personal Communication Network", in Proc. ICT'96, pp. 306-312, 1996.
- [16] T. Ling and N. Shroff, "Scheduling Real-Time Traffic in ATM Networks", in Proc. IEEE INFOCOM 96, pp. 2b.4.1-2b.4.8, 1996.
- [17] H. Zhang and S. Keshav, "Comparison of Rate-Based Service Disciplines", in Proc. ACM SIGCOMM'91, pp. 113-121, 1991.
- [18] N. Passas et al., "A Medium Access Control Framework for Wireless ATM Networks", in Proc. International Workshop on Mobile Communications, Thessaloniki, Greece, Sept. 19-21 1996.
- [19] OPNET Modeler, MIL 3, Inc., 3400 International Drive NW, Washington, DC 20008, 1993.
- [20] C. Blondia and O. Casals, "Performance Analysis of Statistical Multiplexing of VBR Sources", in Proc. INFOCOM '92, pp. 828-838, 1992.

[21] D.M. Lucantoni, “New Results on the Single Queue with a Batch Markovian Arrival”, *Process. Stochastic Models*, vol. 7, 1991.

## **7. Acknowledgements**

This work has been performed in the framework of the project ACTS AC085 The Magic WAND, which is partly funded by the European Community and the Swiss BBW (Bundesamt für Bildung und Wissenschaft). The Authors would like to acknowledge the contributions of their colleagues from Nokia Mobile Phones, Tampere University of Technology, VTT, Ascom Tech AG, University of Lancaster, Lucent Technologies, Robert BOSCH GmbH, University of Ulm, Compagnie IBM France, ETH Zurich, INTRACOM Hellenic Telecommunications and University of Athens.