Quality-of-Service Oriented Medium Access Control for Wireless ATM Networks

Nikos Passas, Sarantis Paskalis, Dimitra Vali, and Lazaros Merakos

Communication Networks Laboratory
Department of Informatics
University of Athens
157 71 Athens, Greece
E-mail: passas@di.uoa.gr
Tel: +301 7230172 (x312)
Fax: +301 7219561

Abstract

The Medium Access Control (MAC) protocol and the underlying traffic scheduling algorithm developed within project Magic WAND (Wireless ATM Network Demonstrator) is 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 algorithm is delay oriented to meet the requirements of the various traffic classes defined by the ATM architecture. The results of the simulation of a number of scenarios are presented to assess the performance of the proposed algorithm.

1. Introduction

Broadband and mobile communications are presently the two major drives in the telecommunication industry. Asynchronous Transfer Mode (ATM) is considered the most suitable transport technique for the future Broadband Integrated Services Digital Network (B-ISDN), due to its ability to flexibly support a wide range of services with quality of service (QoS) guarantees. On the other hand, wireless local area networks (LANs) are becoming popular for indoor data communications because of their tetherless feature and their increasing transmission speed. Wireless communications have been developed to a level where offered services can now be extended beyond voice and data. The combination of wireless communications and ATM, especially in local area environments, can provide freedom of mobility with service advantages and QoS guarantees. The main challenge of wireless ATM is to harmonize the development of broadband wireless systems with B-ISDN/ATM and ATM LANs, and offer similar advanced multimedia, multiservice features for the support of time sensitive voice communications, LAN data traffic, video, and desktop multimedia applications to the wireless user [1]. Emerging standards, such as HIPERLAN or IEEE 802.11, have been designed to provide wireless access to corporate networks, but do not incorporate yet the ATM technology over the air [2].

There are several open issues in the development of wireless ATM. Most of them stem from the fact that ATM was designed having reliable fixed links in mind. More precisely, ATM assumes fixed users, plenty and constant bandwidth, allocated dynamically based on users needs, full duplex and point-to-point transmission, very good transmission quality (that is why error detection and error correction techniques are limited), and small physical layer overhead. On the other hand, in a wireless environment users can move inside the covered range, the available bandwidth in the radio interface is limited and can vary based on the quality of the channel, transmission is usually half duplex and point-to-multipoint, due to the lack of available frequencies, transmission quality is usually poor, requiring advanced error detection and error correction techniques, and physical overhead is much larger than in fixed links, basically due to the synchronisation delay between transmitter and receiver [3].

Currently, a number of research activities are focusing on the topic of wireless ATM to resolve its problems (see for example [4-7]). One of these activities is project Magic WAND (Wireless ATM Network Demonstrator) [8], which 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 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,
- *Access Points* (APs), the base stations of the cellular environment,
- 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.

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 should be able to support all (or a useful subset of) ATM services with often conflicting requirements, and guarantee the required QoS for each connection. It should also guarantee fairness and allocate bandwidth efficiently and dynamically. 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 **M**obile **A**ccess **S**cheme based on **C**ontention **A**nd **R**eservation for **A**TM, or MASCARA. We focus on the structure of the protocol and the scheduling of ATM traffic on the radio interface. Aspects, such as support of handover for mobility, are not discussed here, although they definitively impact the protocol; for these aspects the interested reader is referred to [9].

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 describes the basic features of the scheduling algorithm used in MASCARA. Section 4 presents simulation results on the performance of the proposed scheduling algorithm. Finally, Section 5 contains our conclusions.

2. Medium Access Control in Wireless ATM

2.1 Background

In wireless cellular ATM networks, an advanced MAC protocol is required, able to provide support to all ATM traffic classes, as they are defined by ATM standards, together with efficient use of the scarce available radio bandwidth shared by all the MTs in a cell. Additionally, this protocol should be adaptive to frequent variations of channel quality.

MAC protocols can be grouped in general into five classes [10]: fixed assignment, random access, centrally controlled demand assignment, demand assignment with distributed control, and adaptive strategies. Fixed assignment techniques permanently reserve one constant capacity subchannel to each connection for its whole duration and they perform very well with constant bit rate connections, both in terms of service quality and channel efficiency. However, their performance decreases dramatically when they are asked to support many infrequent users with variable rate connections. In such cases, random access protocols

perform better. A typical example of such a protocol is ALOHA, which permits users to transmit at will [11]; whenever a collision occurs, collided packets are retransmitted after some random delay. It is well known that, although ALOHA-type protocols are easy to implement and attain minimum delays under light load, they suffer from long delays and instability under heavy traffic load. Enhancements of ALOHA include collision resolution techniques that increase the maximum achievable stable throughput [12]. Centrally controlled demand assignment protocols reserve a variable portion of bandwidth for each connection, adjustable to its needs. Unlike random access techniques, these protocols are split into two phases: reservation and transmission. In the reservation phase, the user requests from the system the portion of bandwidth required for its transmission need, and the system responds by reserving the bandwidth and informing the user, while in the second phase the actual transmission takes place. Demand assignment protocols are usually complex, but perform well under a wide range of conditions, although the reservation phase results in time and bandwidth consumption. With distributed control, the users by themselves can decide about their transmissions, based on broadcast information. Finally, adaptive schemes combine elements from the above techniques and they aim to support many different types of traffic [13].

The proposed protocols for the radio interface of wireless ATM networks are based on Frequency Division Multiple Access (FDMA), or Code Division Multiple Access (CDMA), or Time Division Multiple Access (TDMA), or combinations of these techniques. In wireless ATM networks, the lack of available frequencies and the requirement for dynamic bandwidth allocation, especially for variable bit rate connections, make the use of FDMA inefficient. On the other hand, CDMA limits the peak bit rate of a connection to a relatively low value, which is a problem for broadband applications (>2Mbps). Accordingly, most protocols in the area use an adaptive TDMA scheme, due to its ability to flexibly accommodate a connection's bit rate needs, by allocating more or fewer time slots, depending on current traffic conditions. Beyond this general choice of a TDMA-based scheme, the MAC protocols proposed in the literature differ by the technique used to build the required adaptivity in the TDMA scheme. The three main techniques used, alone or in combinations, are contention, reservation, and polling.

Contention-based random access protocols are simple and require minimal scheduling. An example is the slotted ALOHA with exponential back-off protocol presented in [6]. Functionality that can be omitted from the MAC layer, such as handover and wireless call admission control, is pushed to the upper layers. These protocols, attain good performance under light traffic, basically due to the short delays when the number of collisions is limited. They also fit well with the statistical multiplexing philosophy of ATM. Nevertheless, their performance is questionable under heavy traffic conditions or when multiple traffic classes must be supported with guaranteed QoS.

Another group of protocols uses reservation techniques, mainly through reservation/allocation cycles, to dynamically allocate the available bandwidth to connections, based on their current needs and traffic load.

A well designed representative protocol of this group can be found in [4]. It is a TDMA Time Division Duplex (TDD) protocol, where time is divided in constant length frames and every frame is subdivided in a request subframe and a data subframe. The request subframe is accessed by MTs, through a simple slotted-ALOHA protocol, in order to declare their transmission needs, while the data subframe is used for user data transmission. The allocation of data slots is performed by the AP based on a scheduling algorithm, and the MTs are informed through broadcast messages. This kind of protocols are more compex and introduce some extra delays, due to the required reservation phase, but on the other hand, they are stable under a wide range of traffic loads and can guarantee a predictable quality of service, which is very important in wireless ATM networks. Their performance depends to a large extent on the scheduling mechanism used for the allocation of the available bandwidth. A number of scheduling algorithms have been proposed recently, which try to separate real-time and non-real-time connections (for example, see [14]). A minimum bandwidth is allocated to non-real-time connections, while real-time connections are served as soon as possible.

A third group of protocols uses adaptive polling to distribute bandwidth among connections (e.g., [15], [16]). A slot is given periodically to each connection, without request, based on its expected traffic. Compared to reservation-based protocols, these protocols are simpler, since there is no reservation phase, but their performance depends on the algorithm that determines the polling period for each connection. If the polling period is shorter than needed, then they might suffer from low utilization, since many slots will be empty. On the other hand, if the polling period is longer than needed, they result in increased delays and poor QoS. The problem becomes more difficult for variable bit rate bursty connections. Several proposals suggest an adaptive algorithm to decide on the polling period of each connection, based on total traffic load, expected traffic for each connection, and required QoS [15].

Finally, to improve performance, a combination of the above schemes is possible; for example a protocol that is based mainly on reservation, but has also a random-access part for urgent traffic. However, attention should be paid in order not to make such a protocol too complex and difficult to implement and operate.

2.2 MASCARA: The Medium Access Control Protocol in WAND

The MAC protocol for the radio interface of WAND is based both on reservation and contention techniques, and it is called **M**obile **A**ccess **S**cheme based on **C**ontention and **R**eservation for **A**TM, or MASCARA. The multiple access technique used in MASCARA for uplink (from the MTs to the AP of their cell) and downlink (from the AP to its MTs) is based on TDMA, where time is divided in variable length time frames, which are further subdivided in time slots. The 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 TDD. Slot allocation is performed dynamically,

with the use of the scheduling algorithm described in the next section, to i) match current user needs and attain high statistical multiplexing gain, and ii) provide the QoS required by the individual connections.

The MASCARA protocol belongs to the MAC layer of each MT and AP, which is located between the ATM layer and the radio physical layer. 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. Data link control is required, as the quality of the radio channel is significantly worse than that of conventional wired media (bit error rate can reach values as high as 10^{-3}). For this purpose, the MAC 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. In any case, a WDLC overhead is required in each individual ATM cell transmitted.

One of the most important components of the MAC layer is the Scheduler, which is responsible for scheduling the traffic transmitted through the wireless medium, i.e., decides on the time an ATM cell will be transmitted. Since in WAND MTs communicate through the AP they are associated with, MASCARA is a hub(AP)-based protocol, and the natural place for the Scheduler is the AP. The task of the Scheduler is to determine how the slots of each time frame are allocated to its associated MTs and to downlink transmissions. A well designed scheduling mechanism should allocate the slots in a way that maintains the agreed QoS to the uplink and downlink ATM connections sharing the radio bandwidth, and, at the same time, attains high bandwidth utilization.

As shown in Figure 2, the MASCARA time frame is divided into a DOWN period for downlink data traffic, an UP period for uplink data traffic, and an uplink CONTENTION period used for MASCARA control information. Each of the three periods has a variable length, depending on the traffic to be carried on the wireless channel. 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 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 UP period, or contained in special "control MPDUs" sent for that purpose in the CONTENTION period. At the end of a frame, the AP constructs the next frame, according to the MASCARA scheduling algorithm presented below, 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 (Figure 2). 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 a number slots. In the WAND system, this overhead consumes only one slot, and we refer to it as the *"period overhead"*.

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 bit error rate, characterizing the wireless media asks for not-too-large data packets to keep the packet error rate at tolerable values. To minimize the physical layer overhead, 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 or AP. 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. This way it is possible to follow the slot-based timing structure, whatever the number of transmitted cells contained in a MPDU. Figure 2 sums up the TDMA frame structure. For a more detailed description of the operation of the MASCARA protocol the interested reader is referred to [17].

3. The Scheduling Algorithm of MASCARA

As already mentioned, the scheduling mechanism is critical for the performance of a reservation-based protocol, such as MASCARA. An arbitrary order of slot allocation from the AP, in accordance with some properties of the MASCARA protocol, such as UP/DOWN period 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 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 values 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 traffic scheduler, located at the AP, that takes into account the delay constraints of individual connections in the allocation of bandwidth.

In this section, we describe the scheduling algorithm for the Scheduler of MASCARA. It is called **P**rioritized **R**egulated **A**llocation **D**elay **O**riented **S**cheduling (**PRADOS**), and has two main objectives: i) traffic regulation based on traffic characteristics, and ii) maintenance of the delay constraints of the connections in the radio interface. At the beginning of each frame, the Scheduler has a number of pending "requests" for slot allocation to service, which are either downlink ATM cells waiting to be transmitted, or uplink reservation requests, piggybacked in the data MPDUs. The algorithm can be separated in two independent actions, which are performed in parallel:

1. specification of how many requests for slot allocation from each active connection will be serviced in the current frame, and
2. determination of the exact location in the frame of the time slot allocated to each serviced request.

For the first action, the algorithm combines priorities with a leaky bucket traffic regulator [18]. It sorts connections based on their service class [19], and assigns priorities to them as shown in Table 1 (the larger the priority number the higher the priority)

Additionally, a token pool 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 [19], as declared and agreed upon at the time of connection setup. 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. A token pool is implemented as a token variable, which is increased by one every time a token is generated, and decreased by one every time a slot is allocated to the corresponding connection. The token variable of a connection is allowed to take negative values when slots are allocated to the connection while its token pool is empty.

The first action is divided in two steps. In the first step, the Scheduler services "conforming" requests, defined as requests that belong to connections whose token pool is non-empty (i.e., positive token variable). Starting from priority class 5 (CBR), and going down to priority class 2 (ABR), the Scheduler services requests from connections, as long as tokens and slots are available. UBR connections have no guaranteed bandwidth, thus no token pool is maintained for them, and they are not serviced during this step. At every priority class, it is very probable to have more than one connections having ATM cells to transmit. In that case, PRADOS gradually allocates one slot at a time to the connection (or connections) that possesses the most tokens (i.e., highest token variable), removing one token from the corresponding pool. The rationale is that the connection with the most tokens has consumed proportionally the least bandwidth compared to its declared one, and thus has higher priority for getting slots allocated. At this state, one or both of the following statements hold:

i) all token pools are empty (i.e., token variables are less than or equal to zero),

ii) all requests have been satisfied.

If only statement (i) holds, the Scheduler proceeds with the second step, which involves service of "non-conforming" requests, i.e., requests from connections with non-positive token variable. It starts again allocating slots beginning from priority class 5 (CBR), and down to priority class 1 (UBR), following the same procedure as described above. For a detailed description of the priority leaky bucket mechanism described above the interested reader is referred to [20].

For the second action (determination of the location in the frame of the slot allocated to each serviced request), 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 [21]. 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"* [22]. Consequently, the final transmission time of an ATM cell will be the earliest possible given the ATM cell's initial ordering. The Sheduler allocates slots gradually and constructs the time frame in such a way, so that to satisfy the wireless hop CDT of each connection. The wireless hop CDT can be evaluated by decomposing end-to-end CDT into CDT for each hop of the ATM connection path. If an allocation causes violation of the deadlines of existing allocations, then this allocation is not performed. Below we briefly describe the operation of the algorithm for this second action. A detailed description can be found in [17].

The operation of the second action can be divided in three steps. The purpose of the first step is to make the initial transmission ordering, based on the deadlines. When a request corresponding to an ATM cell is selected for service, the Scheduler attempts to allocate one slot for its transmission. If the request is the first of the corresponding connection, then the algorithm attempts to make the allocation before and as close to its deadline as possible. If shifting of the existing allocations is required, the algorithm ensures that none of them exceeds its deadline. If this is not possible, the allocation is not performed. If the request is not the first for the corresponding connection, the algorithm tries to make the allocation at the end of the connection's cell train, provided again that, if shifting is required, no allocation exceeds its deadline. An example illustrating this procedure is shown in Figure 3. When all pending requests have been processed, the Scheduler proceeds to the second step.

In the second step, the DOWN period of the frame is built. 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). In the space left empty between the last packed downlink allocation and the first uplink allocation, the algorithm adds the period overhead, and tries to pack as many downlink allocations as possible by moving them to the left (Figure 4).

The purpose of the third step is to build the UP period. The operation is analogous to the second step, but now packing is performed between the end of the DOWN period, as produced from the second step and the first unpacked downlink allocation. In the space left empty between the last packed uplink allocation and the first unpacked downlink allocation, the algorithm adds the CONTENTION period, the required period overhead and the frame header, and tries to pack as many uplink allocations as possible by moving them to the left (Figure 5).

The length of the CONTENTION period depends on the expected traffic, and the specific accessing method used. As already mentioned, the CONTENTION period is used for control messages from the MTs to the AP. These control messages are single slot MPDUs, consisting of only a MPDU header including the control information. Two are the most important types of messages that use this period:

- association requests from MTs performing "power on" or handover, and
- reservation requests from uplink connections having traffic to transmit but no allocated slots to piggyback their requests in (e.g., after connection setup or after an idle period).

Quick transmission of these messages is essential for the performance not only of MASCARA, but of the WAND system in general (e.g., handover delay). On the other hand, since MASCARA control traffic is unexpected, the CONTENTION period should be minimized, as much as possible, to avoid waste of bandwidth. The random access algorithm used for the CONTENTION period is part of the traffic scheduling algorithm and depends on the type of feedback information provided to the contending MTs, which in turn depends on the kind of detection that can be provided by the physical layer. If a collision cannot be reliably detected (i.e., the physical layer cannot differentiate a collided slot from an empty slot), then the only available information to the Scheduler is from the number of successfully received messages. This limits the design choices of the random access algorithms that can be used to the class of ALOHA-type algorithms. In any case, the Scheduler should allocate a CONTENTION period length large enough to attain an acceptably low successful transmission delay for the control messages. Additionally, the MTs can contribute to the collision resolution process by appropriately reducing the probability of transmitting in the next frame in case of repeated collisions (backoff algorithms). On the other hand, if collisions can be detected, more efficient algorithms, based on collision resolution controlled by the AP, can be used. An example is the stack based algorithm presented in [23].

4. Simulation Results and Discussion

In this section, we give simulation results on the performance of PRADOS algorithm. The frame structure used in the simulations corresponds to the one described in section 2.2. For the contention period, a variation of the slotted ALOHA protocol is used. MTs are informed about the length of the contention period through a special field in the Frame Header. Each MT attempts transmission of a single slot MPDU in the contention period for every uplink connection that has pending requests, but no MPDUs in the current frame to piggyback them in. The slot that this control MPDU will be transmitted in is chosen randomly, with equal probability among the slots of the contention period. If two or more MTs attempt transmission in the same contention slot, a collision occurs and none of the contenders will succeed in passing its requests to the Scheduler. The collided control MPDUs will be retransmitted in the next frame, provided that the requests they are carrying are still valid.

The contention period length is variable and is calculated at the beginning of each frame to keep the probability of successful transmission in a contention slot larger than an acceptable minimum. For this calculation, the number of transmission attempts in the contention period is required and has to be estimated, since it is unknown to the Scheduler. In the simulation model, we have used the conservative assumption that this number is equal to the number of currently existing connections that have no reserved allocation in the UP period to piggyback their requests on MPDUs transmitted therein. This overestimates the actual number of contenders, since some of these connections will be idle, and, therefore, will have no requests to transmit. Based on the above assumption, the probability of successful transmission in a contention slot can be readily calculated using binomial statistics. In the simulation, the contention period length of a frame was set to the minimum required to keep the successful transmission probability no smaller than 0.5

For comparison purposes, this section also presents simulation results on the performance of a simpler algorithm, called First Come First Served (FCFS). FCFS constructs frames with the same structure as PRADOS, but the priorities and the delay constraints of the different connections are not taken into account. In FCFS, downlink requests are serviced first. The order of the MPDUs in the downlink period corresponds to the arrival order of the corresponding requests of the connections. The uplink period is built in the same way and placed after the downlink period. It is clear that FCFS does not distinguish connections according to their priorities or their delay constraints. For both algorithms, ATM cells, which could not be transmitted prior to their deadline (i.e., within a specified wireless hop CDT), are dropped and considered lost.

The simulation models were built using the OPNET tool [24]. Three scenarios were considered:

i) identical real-time VBR connections,
ii) a mixture of CBR and real-time VBR connections, and,
iii) a mixture of real-time and non-real-time VBR connections.

Performance is measured in terms of loss probability (i.e., the ratio of the dropped ATM cells over the total number of cells generated) and mean delay. The specific simulation parameters are summarized in Table 2.

VBR sources are modeled by means of a discrete-time Markov process belonging to the class of Discrete-time Batch Markov Arrival Processes (D-BMAP). The traffic generated by each VBR source is approximated by the superposition of a number of identical independent on/off sources, called minisources, each generating at a constant rate [25]. The model used for the CBR sources is simply an ATM cell generator that periodically generates an ATM cell in every 284 slots.

4.1 Real-time VBR connections

In this scenario, only real-time VBR connections are considered. The load in the system increases gradually, by adding one pair of connections (one uplink and one downlink) at a time.

Figure 6 gives the loss probability experienced by uplink and downlink connection, versus the number of connections for both PRADOS and FCFS. Observe that, in FCFS, downlink connections experience practically no losses, while the loss probability for uplink connections is high. PRADOS, on the other hand, is reasonably fair and under heavy load the loss probability for uplink and downlink connections is almost the same. The fairness of PRADOS is mainly due to the limits it sets to both the downlink and uplink period lengths, in the second and third step. FCFS on the other hand, allows the downlink period, which comes first in the frame structure, to expand without constraint, causing the expiration of many uplink ATM cells.

From Figure 7, which plots the mean cell delay versus the number of real-time VBR connections, we observe that in both algorithms uplink connections experience longer delays than downlink connections. This is attributed to the frame structure. The mean delay for PRADOS is higher than that for FCFS. This increase in mean delay can be explained by the greater frame length produced by PRADOS, as a result of the longer contention period.

As we can see in Figure 8, the mean contention period length in PRADOS is a few slots greater than that in FCFS. This is because PRADOS schedules according to the ATM cell deadlines and, therefore, not all uplink connections are serviced in each frame; thus, such connections do not have the chance to piggyback their requests on UP period MPDUs. Consequently, these connections will attempt to pass their requests through the contention period, thus increasing its length. In FCFS, the use of the contention period is used only by connections that had been inactive during the past frame, resulting in shorter frames, i.e., shorter delays.

The use of PRADOS is beneficial in environments where different types of connections are considered, and the prioritized service of different connections results in better performance. This is better shown in the following scenarios.

4.2 CBR and real-time VBR connections

Here we consider 50 CBR connections (25 uplink, 25 downlink) to be active during the whole simulation time, while real-time VBR connections are added gradually to the system in pairs (1 uplink, 1 downlink).

Recall that in PRADOS, CBR connections are treated with the highest priority. In our initial simulations, we have observed that this, in conjunction with a random initial scheduling that spreads CBR allocations in different locations within a frame, can cause small groups of unallocated slots, which reduce the scheduling flexibility of the algorithm and deteriorate performance. One solution to this problem is to group the allocations of identical CBR connections in consecutive slots, so that they appear as one CBR connection

with large bursts. This could be done by synchronizing the cell generation times during call establishment for uplink CBR connections. The simulations presented here use the above synchronization.

Figure 9 shows that, in both algorithms CBR connections experience fewer losses than real-time VBR do. Furthermore, we observe that PRADOS outperforms FCFS in both connection classes, although the improvement is not impressive. The relatively small performance difference between the two algorithms was expected, since both connection classes (CBR and real-time VBR) in this simulation scenario have the same delay constraints. Therefore, PRADOS cannot postpone the transmission of real-time VBR ATM cells, without violating their deadlines, in order to transmit CBR ATM cells sooner.

From Figure 10 we observe that as traffic increases, the mean delay for CBR connections becomes lower, than that of real-time VBR connections, in both algorithms. For PRADOS, this was expected, since it services CBR requests first. Under light traffic, bursts of VBR requests cannot be serviced immediately, and thus, they cause worse delays for CBR connections. Nevertheless, this increased delay for CBR connections under light load conditions does not increase their loss probability, as already seen in Figure 9. Owing to its delay oriented scheduling, as load increases, PRADOS attains lower mean delays for CBR and real-time VBR connections than FCFS does.

4.3 Real-time and non-real-time VBR connections

Under this scenario, 10 real-time VBR connections are always active (5 uplink, 5 downlink) in the system, while pairs of non-real-time VBR connections (1 uplink, 1 downlink) are added gradually.

From the definition of FCFS, it is clear that it does not distinguish the connections according to their delay constraints. Accordingly, it treats real-time and non-real-time connections equally, resulting in almost equal values for the mean delay (Figure 11). PRADOS, on the other hand, services requests from real-time connections faster than those from non-real-time connections, since the latter can tolerate longer delays. This prioritive treatment of PRADOS for the real-time connections results in lower mean delay compared to that of FCFS. The price to pay is an increase in the mean delay for non-real-time connections for PRADOS, compared to FCFS. Nevertheless, this cannot be considered as a drawback of the algorithm, since non-real-time service classes have, in general, loose delay constraints.

Figure 12 plots the loss probabilities for both connection types versus the number of non-real-time connections. Owing to their large CDT, non-real-time connections experience a much lower loss probability, compared to real-time connections, in both allocating schemes. Furthermore, observe that losses for non-real-time ATM cells start only at heavy loads. Comparing the two algorithms, we see in Figure 12 that losses for real-time connections in FCFS are higher than those in PRADOS, whereas the non-

real-time connections experience less losses in FCFS. This is as expected since PRADOS treats real-time connections with greater priority and takes into account their stricter delay constraints.

5. Conclusion

The medium access control for the radio interface of a wireless ATM network is an important system component, since it has to provide both efficient use of the scarce radio bandwidth and maintain QoS guarantees over the wireless hop of ATM connections involving MTs, in a multiservice environment.

After giving some background on the classes of MAC protocols that have been proposed in the literature for use in ATM networks, we have described MASCARA, the MAC protocol being designed in ACTS project Magic WAND, with focus on the scheduling algorithm used for bandwidth allocation. MASCARA is a TDMA-/TDD-based protocol, using both reservation and contention for accessing the medium. In MASCARA, the TDMA frame length, as well as the length of the uplink, downlink and contention periods within a frame, are variable to provide the required adaptivity to changing traffic conditions in a multiservice environment, and attain better performance. The scheduling algorithm (PRADOS) focuses on satisfying the delay constraints of the various connections, to avoid CDT violations.

To evaluate the performance of the proposed scheduling algorithm, we have performed simulations of three different traffic scenarios, combining CBR, real-time VBR, and non-real-time VBR connections. Furthermore, we have compared the performance of PRADOS with that of a simpler algorithm (FCFS), which does not take into account priority classes and delay constraints. The obtained simulation results show that the proposed algorithm is promising and attains better loss and delay performance than that of the FCFS algorithm. Moreover, by using connection prioritization and delay oriented allocation, QoS of real-time connections is improved, with minimal impact on the QoS offered to non-real-time connections.

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 fór Bildung und Wissenschaft). The Authors would like to acknowledge the contributions of their colleagues from Nokia Mobile Phones, Tampere University of Technology, Technical Research Centre of Finland, Ascom Systec AG, University of Lancaster, Lucent Technologies WCND, Robert BOSCH GmbH, University of Ulm, Compagnie IBM France, Eurecom Institute, ETH Zurich, INTRACOM Hellenic Telecommunications, and University of Athens.

References

[1] D. Raychaudhuri and N.D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks", IEEE Journal on Sel. Areas in Commun., Vol. 12, No. 8, pp. 1401-1414, Oct. 1994.

[2] D.C. Cox, "Wireless Personal Communications: What Is It?", IEEE Personal Communications, Vol. 2, No. 2, pp. 20-35, April 1995.

[3] F. Bauchot, "MASCARA: A Wireless ATM MAC Protocol", in Proc. Wireless ATM Workshop, Helsinki, September 1996.

[4] D. Raychaudhuri et al., "WATMnet: A Prototype Wireless ATM System for Multimedia Personal Communication", IEEE Journal on Sel. Areas in Commun., Vol. 15, No. 1, pp. 83-95, Jan. 1997.

[5] M. Naghshineh and A.S. Acampora, "QoS Provisioning in Micro-Cellular Networks Supporting Multiple Classes of Traffic", ACM Wireless Networks, 1996.

[6] J. Porter and A. Hopper, "An ATM-Based Protocol for Wireless LANs", Olivetti Research Ltd. Tech. Rep. 94.2, April 1994, available at ftp://ftp.cam-orl.co.uk/pub/docs/ORL/tr.94.2.ps.Z

[7] A.S. Mahmoud et al., "A Multiple Access Scheme for Wireless Access to a Broadband ATM LAN Based on Polling and Sectored Antennas", IEEE Journal on Sel. Areas in Commun., Vol. 14, No. 4, pp. 596-608, May 1996.

[8] The Magic WAND Wireless ATM Demonstrator is available at http://www.tik.ee.ethz.ch/~wand.

[9] H.Hansen et al., "Description of the Handover Algorithm for the Wireless ATM Network Demonstrator (WAND)", in Proc. ACTS Mobile Communications Summit, Granada, Spain, November 1996.

[10] F.A. Tobagi, "Multiaccess Link Control", Computer Network Architectures and Protocols, P.E. Green, Jr., ed., New York: Plenum Press, 1982.

[11] D. Bertsekas and R. Gallager, "Data Networks", Prentice-Hall, Englewood Cliffs, NJ, 1987.

[12] J.L. Massey, "Collision-Resolution Algorithms and Random-Access Communications", Multi-User Communication Systems (CISM Courses and Lectures Series, No. 265), Springer-Verlag, 1981.

[13] E.Ayanoglu et al., "Wireless ATM: Limits, Challenges, and Proposals", IEEE Personal Communications, Vol. 3, No. 4, pp. 18-34, August 1996.

[14] X. Wu et al., "Dymanic Slot Allocation Multiple Access Protocol for Wireless ATM Networks", in Proc. IEEE International Conference on Communications, Montreal, Canada, June 1997.

[15] C.-S. Chang et al., "Guaranteed Quality-of-Service Wireless Access to ATM Networks", IEEE Journal on Sel. Areas in Commun., Vol. 15, No. 1, January 1997.

[16] A.S. Mahmoud, D.D. Falconer, and S.A. Mahmoud, "A Multiple Access Scheme for Wireless Access to Braodband ATM LAN Based on Polling and Sectored Antennas", IEEE Journal on Sel. Areas in Commun., Vol. 14, No. 4, May 1996.

[17] N. Passas et al., “MAC Protocol and Traffic Scheduling for Wireless ATM Networks”, accepted for publication, ACM Mobile Networks and Applications Journal, special issue on Wireless LANs, 1997.

[18] A.E. Eckberg et al., “Meeting the Chalenge: Congestion and Flow Control Strategies for Broadband Information Transport”, in Proc. IEEE GLOBECOM’89, pp. 49.3.1-49.3.5, 1989.

[19] The ATM Forum, “User-Network Interface (UNI) Specification”, Version 3.1, September 1989.

[20] N. Passas, D. Skyrianoglou, and L. Merakos, “Traffic Scheduling in Wireless ATM Networks”, in Proc. IEEE ATM’97 Workshop, Lisbon, Portugal, May 1997.

[21] 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.

[22] H. Zhang and S. Keshav, “Comparison of Rate-Based Service Disciplines”, in Proc. ACM SIGCOMM’91, pp. 113-121, 1991.

[23] N. Passas et al., “A Medium Access Control Framework for Wireless ATM Networks”, in Proc. International Workshop on Mobile Communications, Thessaloniki, Greece, September 1996.

[24] OPNET Modeler, MIL 3, Inc., 3400 International Drive NW, Washington, DC 20008, 1993.

[25] C. Blondia and O. Casals, “Performance Analysis of Statistical Multiplexing of VBR Sources”, Proc. INFOCOM ’92, pp. 828-838, 1992.

Priority number	Service class
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)

Table 1: Service class priorities

Channel characteristics	Overheads
channel capacity = 20 Mbps slot duration = $20.6 \cdot 10^{-6}$ sec	MPDU overhead = 1 slot period overhead = 1 slot contention period (variable)
Connection Characteristics	
CBR Connections	**VBR Connections**
constant bit rate = 64 kbps wireless hop CDT = 5 msec (242 slots)	mean rate = 256 kbps standard deviation = 128 kbps wireless hop CDT for: real-time = 5 msec (242 slots) non-real-time = 25 msec (1210 slots)

Table 2: Simulation parameters

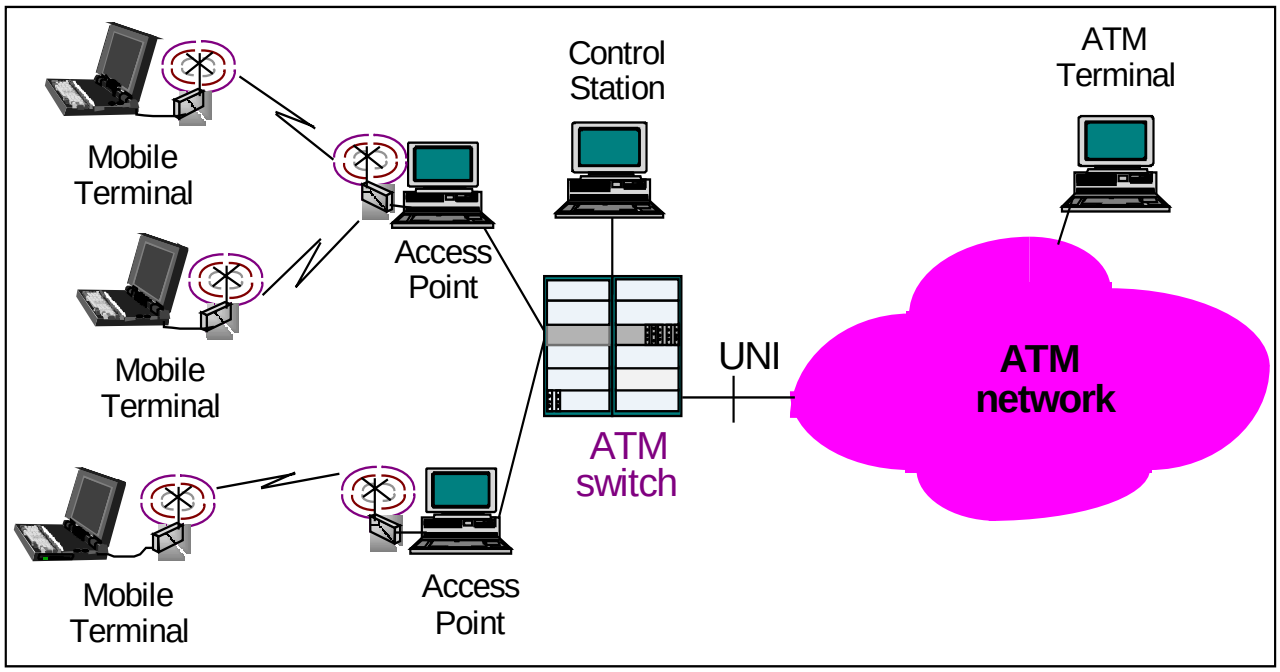

Figure 1: A WAND system

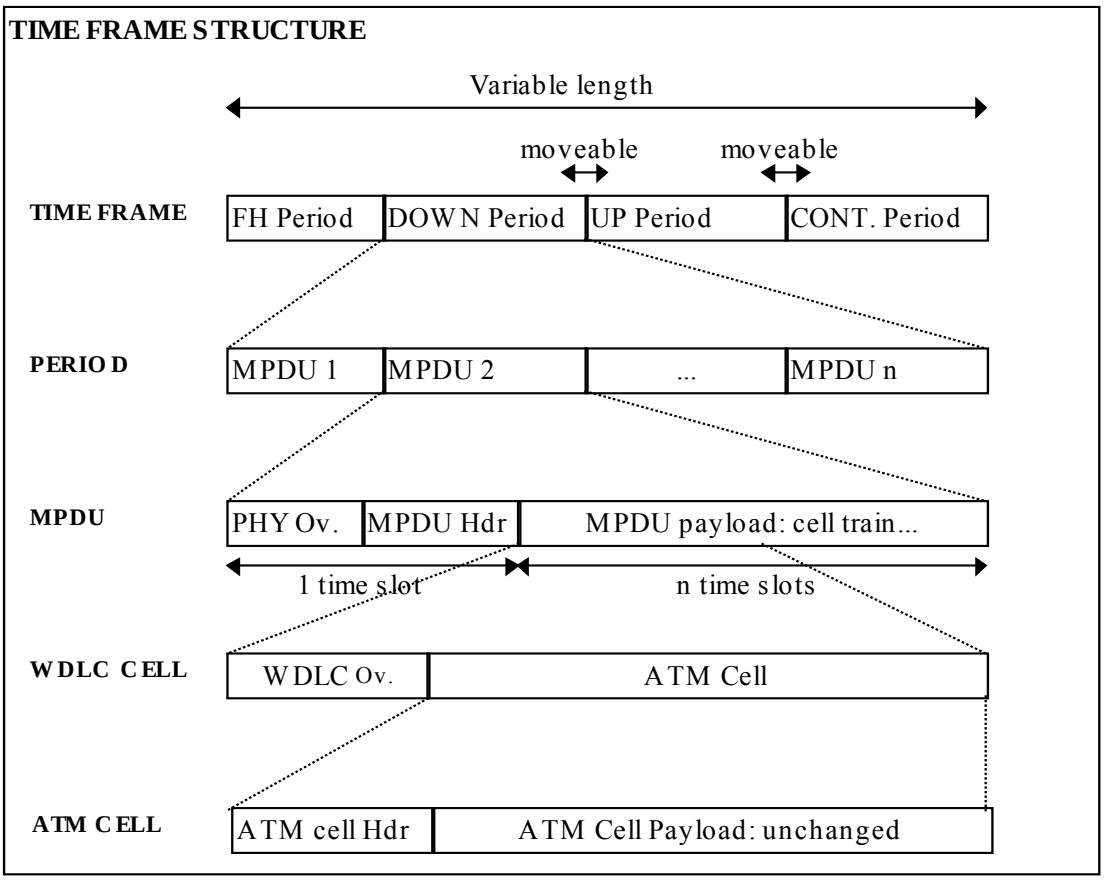

Figure 2: Time Frame Structure

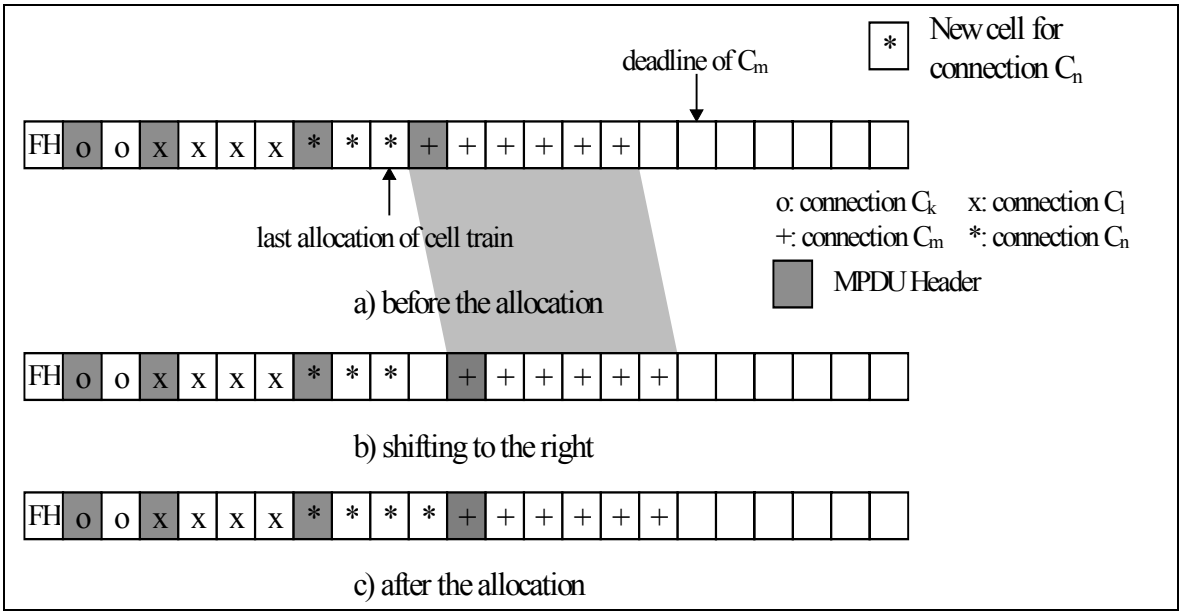

Figure 3: Example of allocation when there are no free slots before the deadline

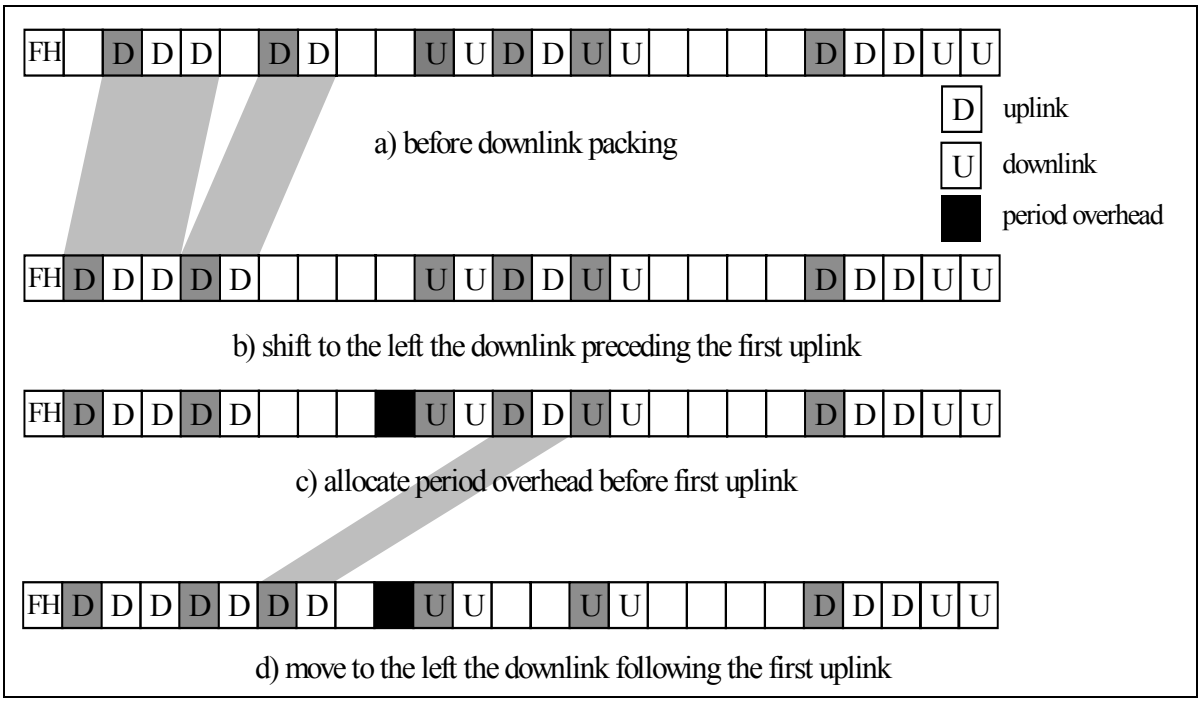

Figure 4: Packing of downlink allocations

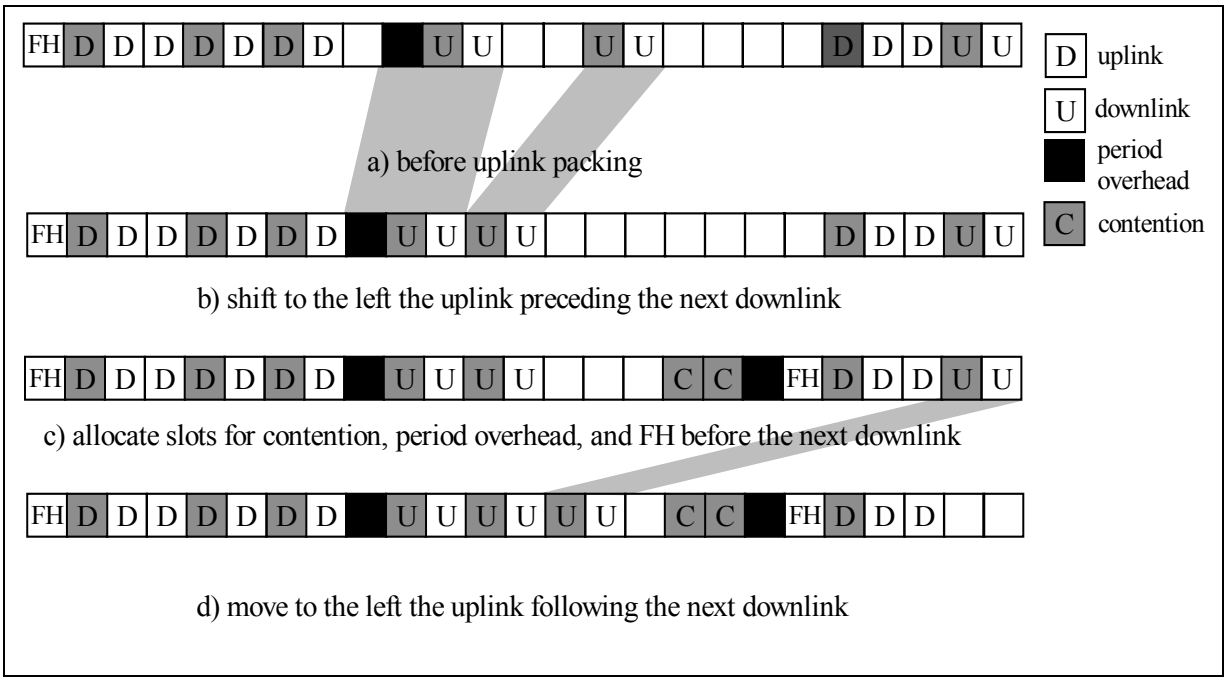

Figure 5: Packing of uplink allocations

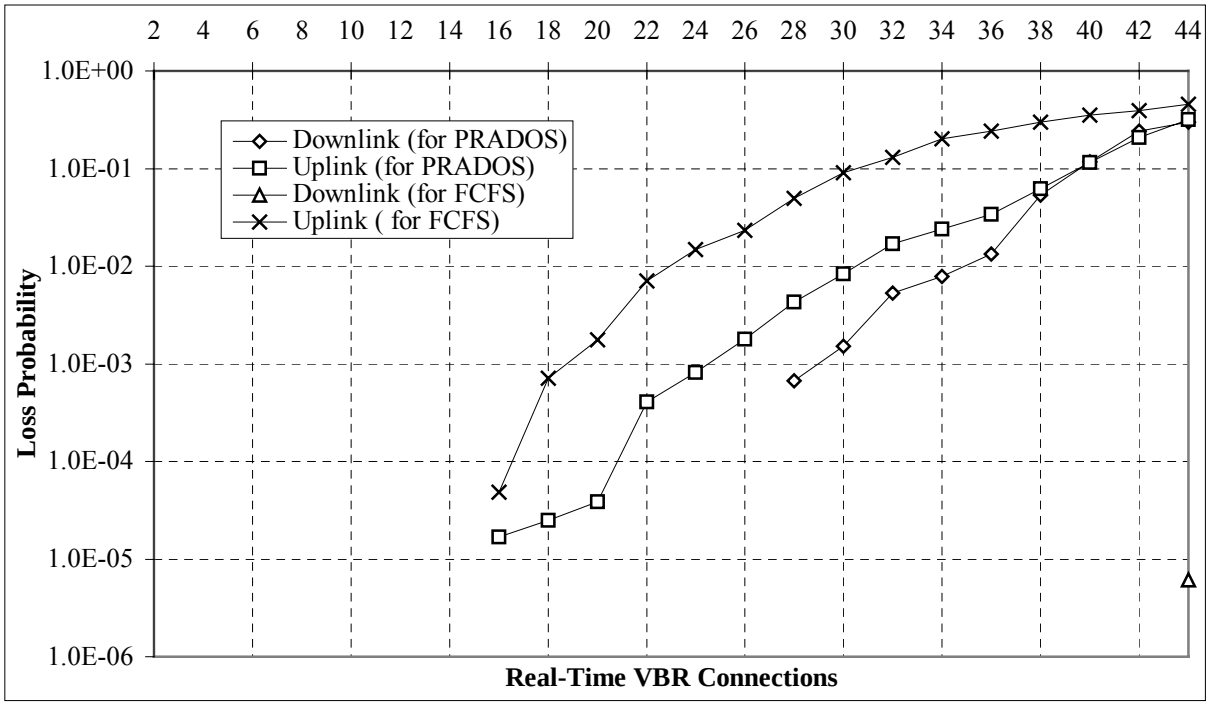

Figure 6: Loss probability for identical real time VBR-connections

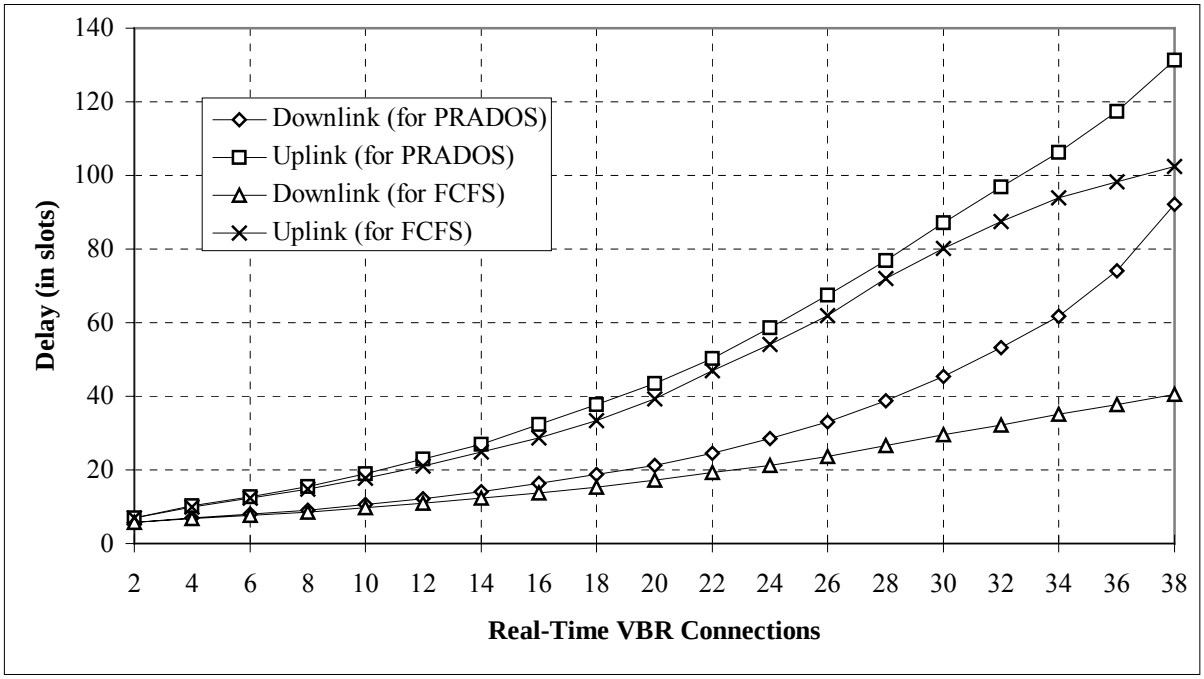

Figure 7: Mean delay for identical real-time VBR connections

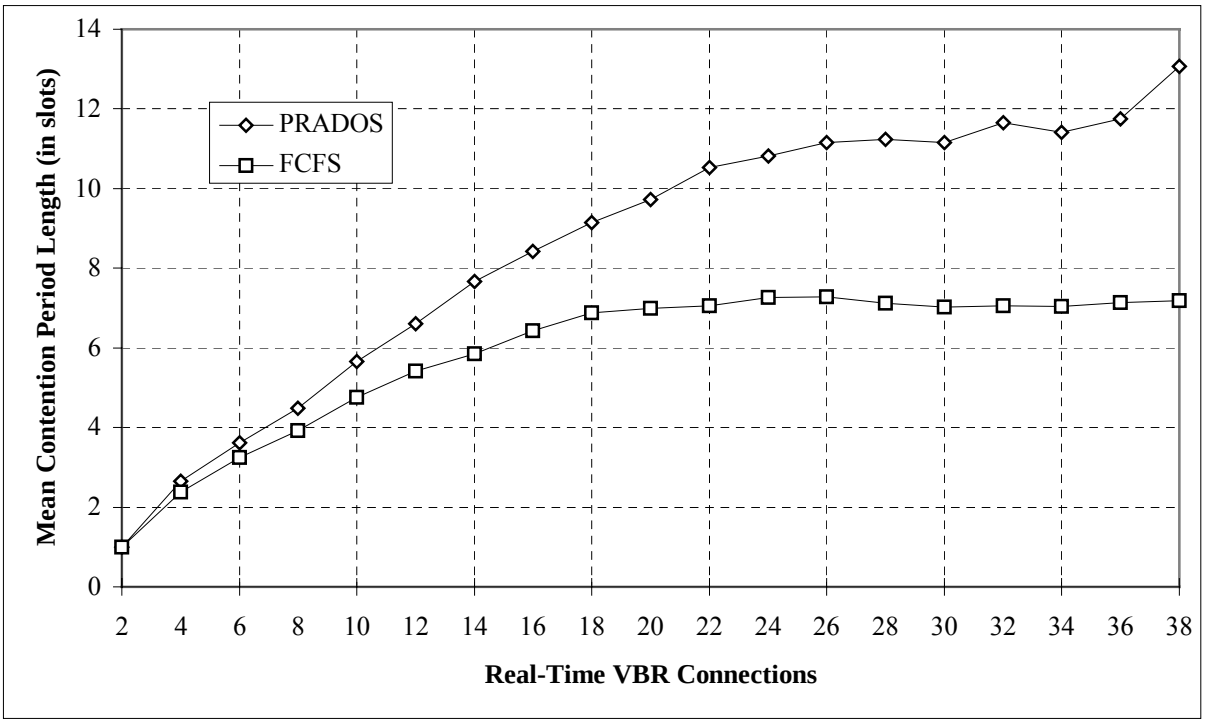

Figure 8: Mean contention period length for identical real-time VBR connections

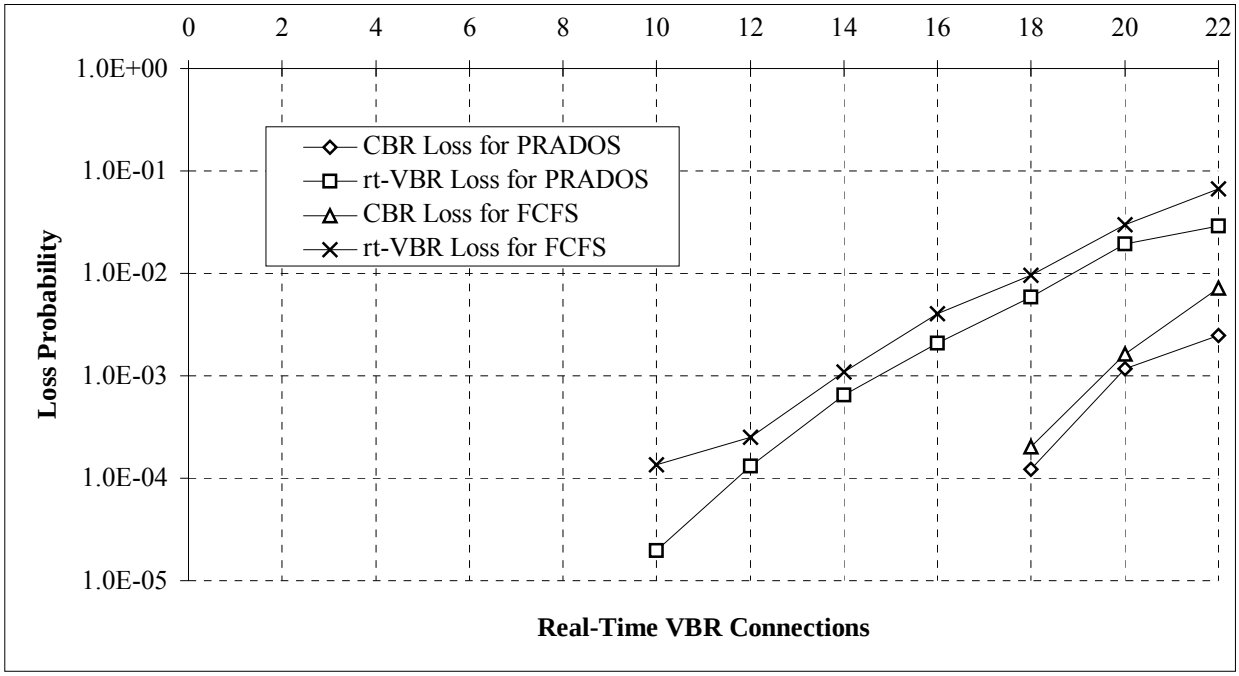

Figure 9: Loss probability for CBR and real-time VBR connections

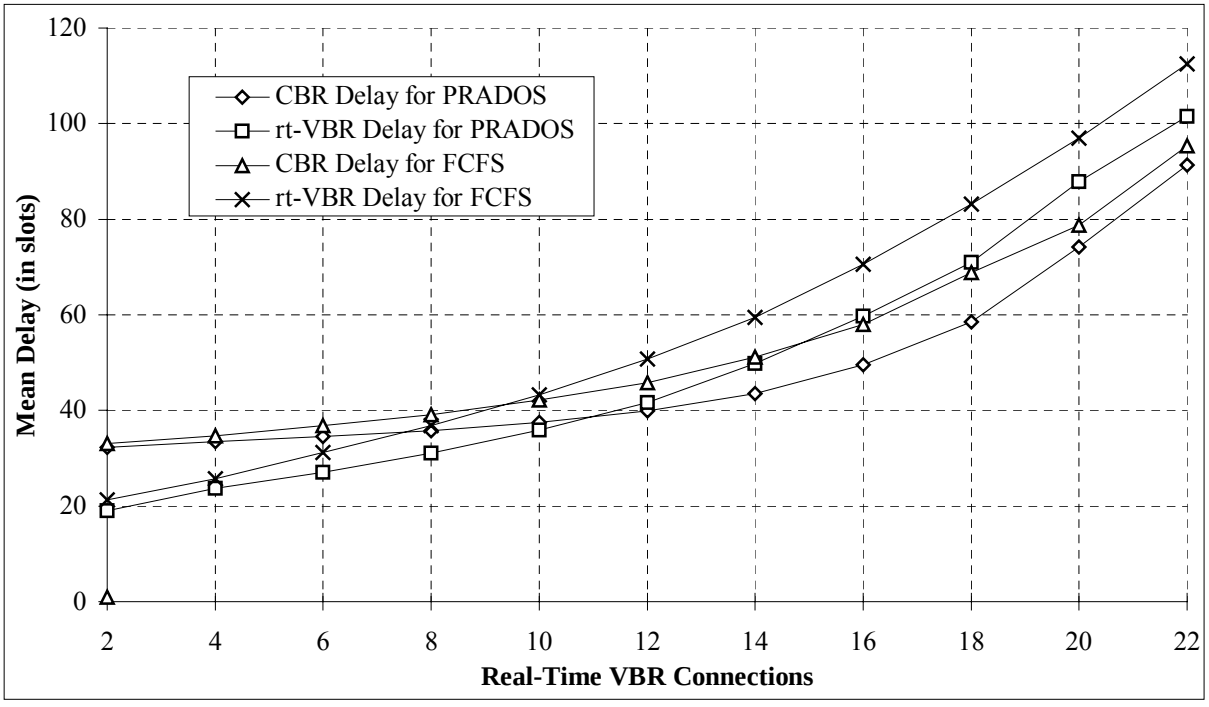

Figure 10: Mean delay for CBR and real-time VBR connections

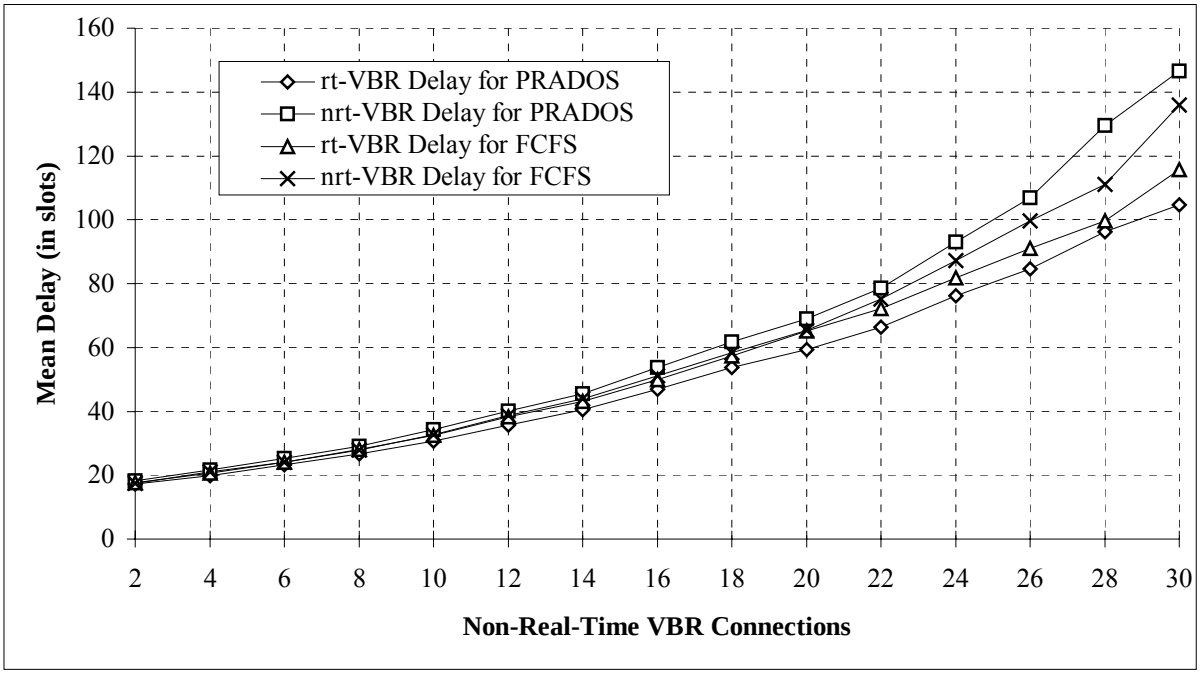

Figure 11: Mean delay for real-time and non-real-time connections

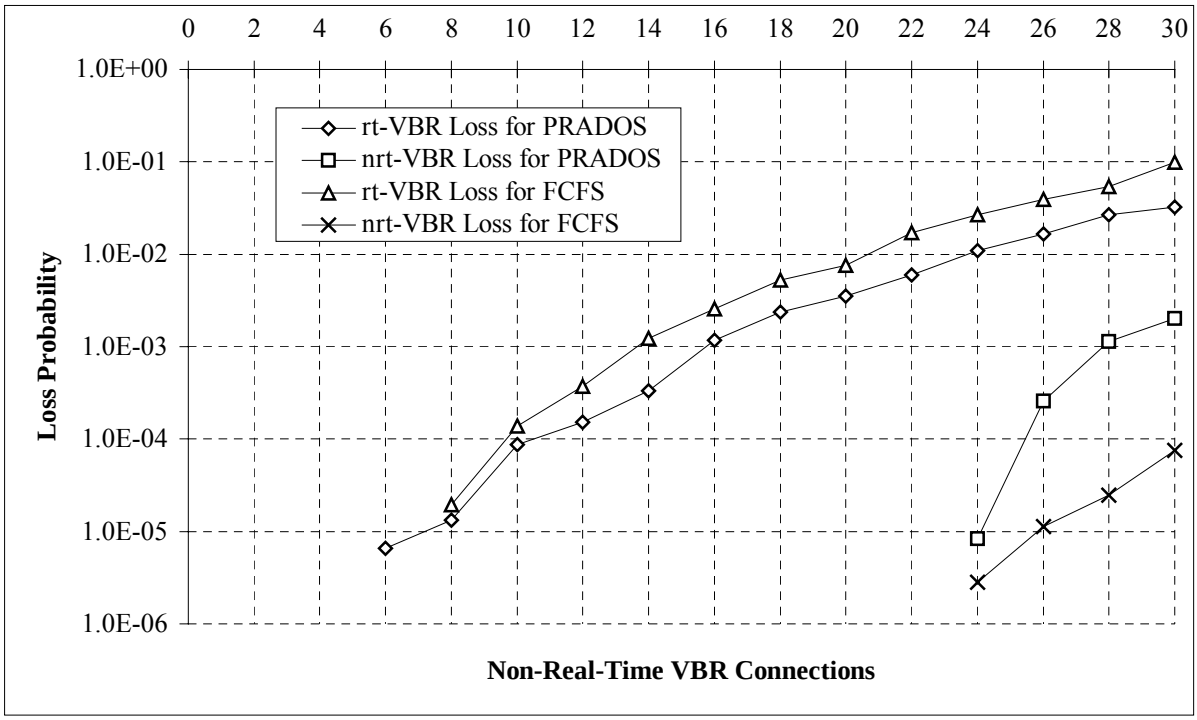

Figure 12: Loss probability for real-time and non-real-time connections