A MAC Protocol for Wireless ATM Systems Supporting the ATM Service Categories

B. Van Houdt[†], C. Blondia[†], O. Casals [‡], J. Garcia [‡], D. Vasquez-Cortizo[†]

[†] *University of Antwerp*
Dept. Math. and Computer Science
Performance Analysis of Telecommunication Systems Research Group
Universiteitsplein, 1, B-2610 Antwerp - Belgium
{blondia,vanhoudt,dcortizo}@uia.ua.ac.be

[‡] *Polytechnic University of Catalunya*
Computer Architecture Department
C/ Jordi Girona, 1-3, E-08034 Barcelona - Spain
{olga,jorge}@ac.upc.es

Abstract

This paper presents a new Medium Access Control (MAC) protocol for broadband wireless ATM Local Area Networks. Quality of Service (QoS) is provided by the support of the different ATM service categories. The system is TDMA with Frequency Duplexing and has a centralized architecture: one Base Station (BS) controls the access to the shared medium for multiple Mobile Stations (MS) per cell. The main issues addressed concern the information exchange between BS and MSs, the bandwidth allocation algorithm for the uplink channel (the scheduler at the BS), the frame structure and the analysis (throughput and delay) of the Splitting algorithms combined with Polling to be used in the uplink contention channel.

1 Introduction

Broadband and mobile communications have become two major issues in the telecommunications community. The standardization of ATM, meant to be the technology for the future Broadband Integrated Services Digital Network (B-ISDN), and the widespread commercial success of wireless standards such as GSM and DECT have both arisen the interest in a technology allowing a wireless (indoor) access to the high-capacity integrated-services wired networks being already deployed.

The importance of wireless broadband systems is also made evident by the number of projects within the European program ACTS concerning the topic (e.g. MEDIAN, Magic WAND [7], SAMBA [10], AWACS, AMUSE), despite the technical difficulties involved: a broadcast channel with Bit Error Probabilities that can be as bad as 10^{-3}-10^{-4}, the limited spectrum availability, the high frequencies involved and the need to provide the different traffic classes with different

Quality of Service (QoS) guarantees.
Within this framework a number of Medium Access Control (MAC) protocols have been proposed such as MASCARA [9], PRMA [4], DSA++ [10], DQRUMA [6] and others [13, 3, 12]. A feature in common (except for [3]) is the centralized TDMA architecture, with a Base Station (Access Point) running a scheduler which rules the access to the broadcast channel in the uplink (the transmissions from the mobile stations to the base station). The transmission needs at the mobiles are usually piggybacked in the data packets that have already been scheduled, but the provision of an uplink contention channel is also necessary not only for the new terminals entering the network but also for those terminals which have remained silent for a long period or have sudden increases in their uplink transmission needs (e.g. VBR traffic).

The work presented in this paper is a new proposal for a wireless broadband MAC protocol whose main contributions concern the scheduler design and the aim at a more efficient uplink contention channel. Efficiency in the uplink contention channel has often been overlooked on the assumption of a poor channel utilization. In this paper we provide analytical results for the calculation of the throughput and delay associated to the requests when a splitting algorithm [1, 11] combined with polling is employed.

The system we are considering in this paper has the following characteristics. Consider a cell in an ATM network with diameter of around 100m, consisting of a base station (BS) serving a finite set of mobile stations (MS) by means of a shared radio channel. The number of MS is not necessarily fixed, as new MSs may enter the cell and others may leave the cell. The BS is connected to an ATM switch which supports mobility, realizing access to the wired ATM network. ATM PDUs arriving in the BS with destination a MS are broadcasted downlink. ATM PDUs originating from a MS share the uplink radio channel using a well defined access protocol. The design and evaluation of such a MAC protocol is the subject of this paper. The netto bit rate of the uplink and downlink channels is of the order of 25 Mbit/s (a gross rate of 50 Mbit/s). The access technique is Time Division Multiplexing Access (TDMA) and Frequency Division Duplex (FDD). The BS of a cell attaches to each MS in that cell an address that will be used by the MAC layer. This MAC address consists of 2 bytes. In addition, the BS maintains for each MS a table containing connection related information: type of service category, traffic contract parameters, etc.
We propose a MAC protocol with centralized control located in the BS. The uplink channel is used by the MSs to inform the BS about its bandwidth needs (through requests) and to transmit ATM PDUs (data cells, RM cells, OAM cells), according to the MAC protocol. The downlink channel is used to send acknowledgments, information about the permission to use the uplink channel and to transmit ATM PDUs (data cells, RM cells, OAM cells). The bandwidth of the uplink channel is allocated among the active MSs on basis of the ATM service category the connections carried by an MS belong to, the traffic contract and the current traffic conditions. The service categories that are considered are CBR, VBR, ABR and UBR. We assume that each MS can carry at most one connection of each service category. In the sequel of this paper ATM PDUs will be called as usual cells.

The paper is structured as follows. The second section describes the information exchange between MS and BS. Section 3 gives a detailed description of the mechanism used to inform the bandwidth needs of the MS. Section 4 presents the structure of the uplink and downlink frame. Section 5 is devoted to the bandwidth allocation algorithm, i.e. the rules that are used to share the available bandwidth among the active stations. Section 6 deals with the specific features of the MAC protocol to support the flow and congestion control schemes of the ABR service category. The most critical part of the MAC protocol, namely the contention resolution scheme, is analytically evaluated in Section 7. The details of the model and the analysis are not presented, but they are used in Section 8, where numerical results illustrate the impact of the different system parameters on the delay and the throughput. Conclusions and directions for future research are given in Section 9.

2 Information Exchange between MS and BS

In this section we describe the information exchange between MS and BS. Assuming a MAC protocol with centralized controller located in the BS, each MS must be able to inform the BS about its bandwidth needs and the BS should be able to inform the MS about the received bandwidth. The information exchange is based on a request/permit mechanism.

2.1 Permits

In order to be allowed to use the uplink channel, the MS has to receive a *permit* from the BS. A permit has a length of 4 bytes and contains the following information

(i) the address of the permit's destination MS (2 bytes)

(iii) the service category of the connection receiving the permit: CBR, VBR, ABR, UBR (2 bits)

(iv) an indication of the instant the MS can send an upstream PDU (i.e. the sequence number of the slot in the next upstream frame that may be used to send the upstream cells) (14 bits)

2.2 Requests

The MS declares its bandwidth needs to the BS by means of requests. There are different ways to send requests, depending on the ATM service category the requested bandwidth will be used for. A request needs 8 bytes and contains the following information:

(i) address of the MS that is issuing the request (2 bytes)

(ii) per type of service category (VBR, ABR, UBR), the number of cells that are waiting in the respective queues (3 times 2 bytes)

There are two different ways to send requests (in the second case a slightly different format is used, see section 4.1):

(i) Piggybacked with upstream packets : when an MS is allowed to transmit a packet, its future bandwidth requirements are added (i.e. piggybacked).

(ii) Using a contention resolution protocol : specific time intervals are used to let all MSs who want to inform the BS about their bandwidth needs compete for access to the medium to send this information.

Depending on the service category, a combination of these mechanisms is used to declare the bandwidth needs of the MS.

3 The Request Mechanism

In this section we describe the mechanisms used to inform the BS about the bandwidth needs of the active MS. The two different ways of sending requests mentioned in Section 2.2 are used to support Quality of Service provisioning in the different ATM service categories.

3.1 Request Mechanism for CBR Traffic

In view of the regular arrival instants of PDUs in the MS of a CBR connection, and in order to reduce the overhead introduced by the request mechanism, a polling scheme is used without explicitly sending requests. The MS does not generate a request for each cell it wants to transmit. Instead, the Permit Distribution Algorithm (see Section 5) generates at regular instants (i.e. according to the Peak Emission Interval agreed at call setup for a CBR connection and maintained in a table in the BS) permits for each MS with a CBR connection.

3.2 Request Mechanism for VBR Traffic

Due to the variability of the cell rate, we can not use the above scheme any longer. In principle a piggybacking scheme is proposed for this type of services as this introduces a minimal overhead. This means that a MS can add a request to each upstream cells (whatever the service category this cell belongs to: e.g. an upstream slot containing an ATM cell belonging to a CBR connection can contain a request for a VBR connection carried by the same MS). However, this scheme fails in case the last upstream cell leaves behind empty buffers and the VBR connection is still active (i.e. will generate a cell in the future). In particular the first cell of a new burst needs a mechanism to inform the BS about its presence. For this we propose a combination of a contention resolution and polling scheme, called the *Identifier Splitting Algorithm with Polling.* In what follows we give a more detailed description of the two parts of this protocol.

3.2.1 The Identifier Splitting Algorithm (ISA)

This protocol is based on the well known tree algorithm [1] and was proposed by Petras in [11]. A contention cycle (CC) consists of a number of consecutive upstream frames during which the contention is solved for all requests that want to make use of this scheme at the beginning of the cycle. Requests that intend to use the contention resolution scheme generated by an MS during a CC have to wait for participation till the start of the next CC. In the first frame of a cycle, a number of contention minislots are available,

which can be used for contention resolution (we say that we start at level 2 of the tree since we have 2^2 minislots). The MS selects a minislot according to its MAC address (2 bytes) (contrary to the coin flipping procedure as in the original tree algorithm): an MS uses the first two bits of its MAC address to decide which minislot it will use. The BS checks which transmissions have been successful and informs the MSs that were involved in the scheme in the next downstream frame using a feedback field (see Section 4, Frame Structure). Two situations are possible:

(i) an MS sending in slot k, $1 \leq k \leq 4$, was successful. In this case the MS will eventually be granted a permit by the BS to send an upstream cell.

(ii) an MS sending in slot k, $1 \leq k \leq 4$, was not successful, i.e. a collision with one or more other MSs occurred. If there where l minislots $0 \leq l \leq 4$, with collisions, then the next (second) frame of the CC provides $2 \times l$ minislots for contention resolution and the involved MS apply the same scheme again, each time the next bit of the MAC address is used to decide which of the two minislots, used to resolve the collision, is used.

Due to the delayed feedback, i.e. an MS can only retransmit the request in the next frame since it must wait for the feedback, the tree is traversed in a breadth-first manner, instead of a depth-first way as proposed in the original tree protocol [1].

3.2.2 The Identifier Splitting Algorithm combined with Polling

One of the advantages of the Identifier Splitting Algorithm as opposed to the original protocol is that as the scheme is being resolved, the BS obtains more and more knowledge about the MSs that are still competing. For example, if the BS notices that the tree at level i (the top of the tree containing 4 minislots is called level 2) contains k collisions and the MAC-addresses are n bits long then the BS knows that the remaining competing MSs can only have $k2^{n-i}$ possible addresses. This follows from the fact that each slot at level i corresponds to 2^{n-i} addresses. Therefor, if the remaining address space is small enough ($\leq N_p$), the contention protocol can decide to switch to polling. The value N_p, that triggers this polling mechanism, is assumed to be predefined.

The BS need not to inform the stations at which level polling starts, as every MS can preform the necessary calculation(s) by itself, using the feedback information (see section 7.1). Also based on the MAC address, each MS that is involved in the polling knows exactly which minislot it is allowed to use to send its address together with an indication to which service category the connection that needs a permit belongs. In this way, the minislots are used collisoin free to inform the BS in a collision free way about the MS needs.

3.2.3 The Influence of Skipping the First Few Levels

Now we consider a variant of the ISA scheme where the first few levels in the scheme are skipped. So instead of starting with just 4 contention slots, we might provide more slots for the first attempt.

At first the starting level is fixed at a predefined value S_l. It is expected that this has a positive impact on the delay. Apart from that, the throughput might improve in case of high loads. Unfortunately as will be shown in the numerical results in Section 8, this results in some extra throughput losses during silent periods. To solve this we will propose a scheme that changes the starting level dynamically, between level S_{min} and S_{max}. Depending on the length of the previous contention cycle we will increase or decrease the starting level S_l. To make this decision, the system load ρ is not taken into account, as this value is hard to measure or predict in real systems.

The selection of the starting level should be such that the delay results for dynamically changing the starting level S_l are comparable with those of the static scheme where S_l is fixed at S_{max}. This means that we want to minimize the extra delay added by the dynamic scheme.

3.3 Request Mechanism for ABR and UBR

Again a piggybacking mechanism is preferable, but when not possible (see the conditions in 3.2) the Identifier Splitting Algorithm (with or without polling) can be used to allow these connections to declare their bandwidth needs.

4 Frames structure

In this section we give a detailed description of the uplink and downlink frame structure. Both frames have the same fixed length.

4.1 Uplink Frame Structure

The uplink frame contains two types of slots, each having a length of 106 bytes. The total number of such slots in a frame is set to 80, resulting in a constant frame length of 8480 bytes.

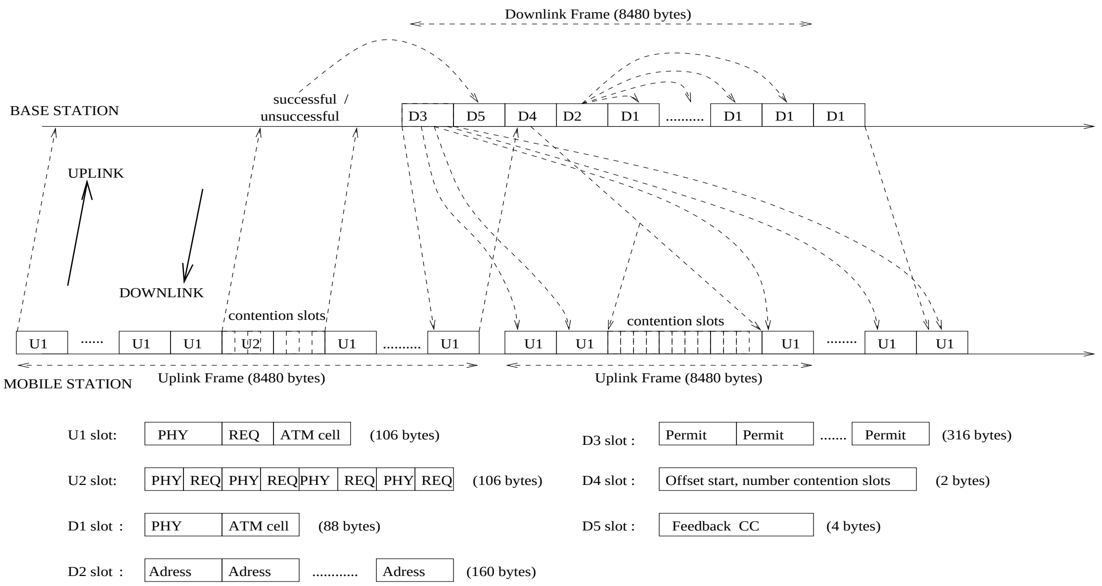

Figure 1: *The Frame Structure*

U1 slot: this slot is used to transmit an uplink ATM cell (53 bytes), together with a piggybacked request (8 bytes). A physical layer overhead of 45 bytes is used for error detection, a safe guard time and sufficient training sequences. This results in a total length of 106 bytes.

U2 slot: a U2 slot is used to allow bursty VBR, ABR and UBR connections to inform the BS about the need for a permit (see Section 3). This type of slot has the same length as a U1 slot and consists of 4 minislots used for contention resolution. Such a minislot will be used by one or possibly more stations during a contention cycle. A minislot consists of the address of the MS using the minislot (2 bytes), an indication of the ATM service category the permit is needed for (VBR, ABR, UBR: 2 bits), while the remaining 194 bits (848 / 4 - 16 - 2) are used to implement a safe guard time, some training sequences and the necessary error control bits. We limit the number of U2 slots to 8, leading to a maximum of 32 minislots (or contention slots) per frame.

4.2 Downlink Frame Structure

The downlink frame contains five kinds of information. The D1 slots contain the downstream ATM cells while the other four types of slots are used for control and feedback information. These slots are grouped together and will be treated together with respect to training sequence and error correction.
D1 slot: this slot contains a downstream ATM cell (53 bytes), accompanied by the necessary physical layer overhead (training sequence, error detection: 35 bytes), resulting in a total of 88 bytes. Each downlink frame contains 80 D1 slots.

D2 slot: this slot is sent before the first D1 slot in a frame and it is used to specify the addresses of the destination MSs of the D1 slots of that frame. Using a D2 slot avoids that each active MS has to listen continuously to the medium in order to detect which D1 slots have its address as destination, leading to an important power consumption reduction. This slot has a length of 80 times 2 bytes.

D3 slot: this slot is used to inform the MS about the permission to transmit a cell in the next upstream frame. This slot contains a variable number of permits, between (80 - 1) and (80 - 8) (80 being the total number of slots in an upstream link and 1 and 8 the lower and upper bound for U2 slots). Each permit requires 4 bytes, hence the length of a D3 slots takes a value between 316 bytes and 288 bytes.

D4 slot: this slot informs the MS which slots in the next upstream frame are declared as U2 slots, i.e. can be used for contention resolution. The offset of the start is specified by means of 13 bits, while the number of slots used for contention can be coded in 3 bits. This results in a total of 2 bytes for a D4 slot.

D5 slot: this slot contains the feedback information for the MS about the result of the contention resolution in the previous uplink frame. For each contention minislot that was available in the previous uplink frame, an indication is given whether there was a collision or not. Since each participating MS knows which minislot it has used, this indication is sufficient for the MS to know if it was successful or not. As the maximum number of minislots in a frame is set to 32, 4 bytes are needed for a D5 slot.

The control and feedback slots (D2, D3, D4, D5) together are protected by an error correction code. Moreover they also contain training sequences. A part of the remaining 958 bytes is used for this purpose, the rest is used for signaling channels (synchronization, paging and others). The total downlink frame length is then 8480 bytes, which is exactly the same as the uplink frame length. Choosing equal lengths solves a number of synchronization problems, in particular with respect to the provided feedback (D5) and permit (D3) information.

5 The Bandwidth Allocation Algorithm

The bandwidth allocation algorithm has to distribute permits among the active connections based on:

(i) the service class the connection belongs to

(ii) the individual contract parameters of the connection

(iii) the current state of the different queues in the MS (i.e. the bandwidth requirement the MS gas for each service category).

In the following sections we describe how the permit distribution is realized (also see [2, 8]).

5.1 CBR connections

The permits for CBR traffic are generated according to 3.1 and put in a "CBR/VBR FIFO" queue. For each MS (with an active CBR connection), the central controller maintains a counter, the CBR Count Down Counter (CBR_CDC), which is set initially to a value $CBR\_CDC(Init)$ equal to the number of slots corresponding to the Peak Emission Interval of the CBR connection. At each slot it is count down by 1 until zero. When the value reaches zero, a permit is generated for that CBR connection and the counter is reset to $CBR\_CDC(Init)$.
The CBR/VBR queue is emptied with the highest priority: when a downstream cell is scheduled for transmission to the MS, the CBR/VBR FIFO queue is checked and if not empty, a permit is added to the passing downstream cell on a FIFO basis.

5.2 VBR connections

The requests that are received according to 3.2 for VBR traffic are converted into permits, and are put into the CBR/VBR FIFO queue at the PCR of that connection, but taking into account the Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS) by using a GCRA algorithm (token pool leaky bucket) (this can be implemented by means of one counter). In the CBR/VBR FIFO queue the permits for VBR traffic compete (among each other and with the CBR permits) on a FIFO basis.

5.3 ABR connections

The requests that are received according to 3.3 for ABR traffic are first stored per MS into an ABR-REQ Counter. For each MS, this counter maintains the number of permits to be granted for ABR cells to that MS. The requests from the ABR-REQ counter are then converted into permits by writing them to the "ABR FIFO" queue at the agreed PCR of that ABR connection. The ABR FIFO queue obtains the second priority (after the CBR/VBR FIFO queue).

5.4 UBR connections

The requests that are received according 3.3 for UBR traffic are first stored into a UBR-REQ Counter, then converted into permits by writing them to the "UBR FIFO" queue at the PCR of that connection (similar to the ABR traffic). The UBR FIFO queue obtains the third priority (after the CBR/VBR FIFO queue and the ABR FIFO queue).

6 Support of ABR service category

This wireless access system is an integral part of the ATM network, and as such, it should cooperate in the congestion and flow control scheme for the ABR service category. The congestion control scheme that is proposed is based on the ERICA algorithm (see [5]).

6.1 Congestion Detection

We have two means of congestion detection of the access system:

(i) Length of request counters : a congested MS can be identified by checking its ABR-REQ Counter and using a threshold. A global congestion can be recognized by checking the ABR FIFO queue and comparing it with a predefined threshold.

(ii) Number of request arrivals : by counting the number of CBR/VBR and ABR requests during a certain period.

Depending on the information that is used, different congestion control mechanisms may be implemented.

6.2 Congestion Control

When using the first congestion detection mode, a binary congestion control seems to be appropriate, where the congested MSs can be identified. When using the second detection method, we can use an explicit rate mechanism. We show how such Explicit Rate mechanism can be implemented.

Assume an observation period of T time slots. Denote the number of VBR request arrivals, resp. ABR request arrivals, during this T time slots a_{VBR}, resp. a_{ABR}. The CBR traffic does not generate requests, however it is possible to compute the number of permits that have to be generated during T slots for CBR traffic. Denoting PEI the Peak Emission Interval of a CBR connection, we have that the corresponding number of arrivals is given by

$$a_{CBR} = \sum_{\text{CBR connections}} \frac{T}{PEI}.$$

The respective input rates are then given by $R_{CBR} = a_{CBR}/T$, $R_{VBR} = a_{VBR}/T$ and $R_{ABR} = a_{ABR}/T$. Let U be the target utilization of the link and R be the total available rate, then the Target ABR Capacity TC_{ABR} is given by

$$TC_{ABR} = \max\{0,\ \ U \times R - (R_{CBR} + R_{VBR})\}$$

and the overload factor O

$$O = \frac{R_{ABR}}{TC_{ABR}}.$$

After each T slots, the value of O is updated according to the above rules ands stored into memory. When an RM cell passes through the BS, the ER field is updated using the last computed ER field as follows. For fairness reasons, first a fair share FS for each MS is computed in the following way

$$FS = \frac{C_{ABR}}{N_a}$$

where N_a denotes the number MSs with an active ABR connection and C_{ABR} denotes the ABR capacity, which can be computed as

$$C_{ABR} = \max\{0,\ \ R - (R_{CBR} + R_{VBR})\}.$$

For each MS with an active ABR connection, we can now compute its share denoted by MS and given by

$$MS = \frac{CCR}{O}.$$

The ER value that is put in the RM cell, ER_{new} is now computed using the above parameters together with the ER field in the incoming RM cell ER_{old}, as follows

$$ER_{new} = \min\{ER_{old}\ \ \min[C_{ABR}\ \ \max(FS, MS)]\}.$$

In order to implement the above congestion control scheme, the MAC controller has to maintain three counters. A first counter keeps track of the time since the last computation of the overload factor was made (i.e. the duration of the observation period of length T). In addition, a counter keeping track of the total number of request arrivals of VBR traffic and CBR permits is needed, together with a counter keeping track of the number of ABR request arrivals during an observation period.

7 Performance Evaluation of the Protocol

7.1 The System Description and Analytical Model

The major performance issue of this protocol is the request mechanism for VBR traffic which uses the contention resolution protocol. Permits for CBR sources are generated periodically. Therefore, the delay experienced by such traffic can be computed by applying the $\sum N_i * D_i/D/1$ queueing model. We have developed an analytical model to compute the delay distribution and the throughput when using the ISA with polling for VBR traffic. In what follows we briefly present the model. A more detailed description of it can be found in a forthcoming paper [14].

The General Setting Let us shortly summarize the main characteristics of the WATM environment in which we are working. As already mentioned the length of the frame is fixed but the number and type of slots is variable. Thus the number of $U2$ slots used for the contention-based channel in the frame will vary. Feedback on these slots is provided, together with other information concerning the structure, at the start of the next downlink frame. Although it is possible to provide feedback during the frame itself, we do not consider this here.

The Address Space For this analysis we assume that each MS has, at most, one connection of each traffic class. We define n as the size of the MAC-addresses (in bits).
When a MS connects to the BS, possibly due to a handover, a MAC address is assigned as follows. The BS generates a set of n bits in a random way. Next it checks whether this address is used by another MS. If not the MS is assigned this address, otherwise this procedure is repeated until a new MAC-address is generated. A similar procedure is used to generate the flow label in IPv6, in order to optimize the performance of the hash tables. In our case we do it to optimize the performance of the ISA protocol. For the analysis we assume that there are 2^n MSs within the observed cell (i.e. all MAC addresses are used).

The Input Traffic We assume that the MSs generate Poisson traffic with a mean of λ requests per frame. As the number of MSs is finite and equals 2^n, the probability mass that lies beyond the value of 2^n is added to that of 2^n to make the distribution finite. Thus if we define the random variable I as the number of requests generated during one frame we have

$$\begin{aligned} P[I=k] &= \frac{\lambda^k}{k!}e^{-\lambda}, k < 2^n \\ P[I=2^n] &= \sum_{k \geq 2^n} \frac{\lambda^k}{k!}e^{-\lambda}. \end{aligned}$$

Notice that we don not need to consider bursty input traffic since we are observing the access channel used by a MS to transmit a first request after a 'long' period of silence (or after a handover). Although in reality there exists a dependency between the addresses that compete during consecutive collision resolution cycles (CCs), we assume that this is not the case. Thus the addresses of the MSs taking part in the scheme at the beginning of a collision resolution cycle are uniformly distributed over the complete address space.

The Number of Slots To make the analytical model tractable, we assume that a frame can allocate enough $U2$ slots to support a full level of the tree. Thus if the tree is resolved at level i we need $i+1-S_l$ frames for that purpose, where S_l is the starting level. Notice that the number of slots used by a CC, containing k participants, is limited to k, whatever the MAC addresses are. This follows from the fact that the number of slots at a level is twice the number of collisions at the previous level and with k elements we have at most $\lfloor k/2 \rfloor$ collisions. As a consequence the impact this assumption has on the performance results is small for low and medium traffic loads. In order to make a fair comparison between the ISA protocol with and without polling we assume that the polling will only be done if all the remaining addresses can be dealt with within one frame.
The size of the remaining address space that triggers the polling mechanism is denoted N_p. To find this value, the BS just needs to count the number of collisions N_C. Depending on the result

of this counting process it switches to polling or not. Let us now calculate the values that trigger the polling as they depend on the level of the tree. Remember that the number of addresses that correspond to a slot at level $i-1$ equals 2^{n-i+1}. Thus if $N_C * 2^{n-i+1}$ is smaller than N_p the BS starts to poll. As a consequence the scheme switches to polling at level i if the number of collisions at level $i-1$ was smaller than or equal to $\lfloor \frac{N_p}{2^{n-i+1}} \rfloor$ (this of course if there were still collisions at level $i-1$).

The Starting Level Suppose that at some point in time the starting level equals S_l and L is the length of this CC. Then the new starting level S_l' obeys the following equation

$$S_l' = \begin{cases} \max(S_l - 1, S_{min}) & L \leq B_l \\ S_l & B_l < L < B_m \\ \min(S_l + 1, S_{max}) & L \geq B_m \end{cases} \tag{1}$$

where B_l and B_m are two predefined values.

7.2 Solving the Analytical Model

Basically this evaluation uses a matrix analytical approach. We observe the system at the start of each CC and characterise the system state by (k, S_l) if there are k competitors in the CC and the starting level equals S_l. To solve this chain we need the transition probabilities (1). Having done that we need to know the probability that a tagged item is successful at a specific level given that we were in state (k, S_l) (2). Next an expression for the expected number of slots at a specific level again conditioned on the same event (3) is derived.

These three computations are rather straightforward for the classical ISA protocol, merely because the events happening in one half of the tree are independent of the other half given the number of competitors in each half. Unfortunately this is not the case if we combine ISA with polling resulting in cumbersome calculations, especially for the expected number of polling slots at each level. More details will be given in a forthcoming paper [14].

8 Numerical Results

In this section we study the impact of the offered traffic load λ, the trigger value N_p, the starting level S_l and related values B_l and B_m, on the mean delay, the delay density function and the throughput of the traffic using the contention resolution scheme. The system parameters are set as follows:

- The number of mobiles is 128, hence $n = 7$.
- The arrival rate λ of the generated traffic varies between 0.1 and 6 per frame.
- The three values studied for N_p are 0, 20 and 40. The case $N_p = 0$ corresponds to ISA without polling.
- The starting level S_l varies from level 2 to 4, corresponding to 4 to 16 minislots (or 1 to 4 slots).
- When studying a system with dynamic starting level, B_l and B_m are set to 1 and 4 respectively. This means that the level decreases if the contention resolution cycle is solved in one frame and that the level increases if four or more frames are needed. The boundary values are set as follows: $S_{min} = 2$ and $S_{max} = 4$.

We study three different scenarios. First we investigate the impact of the polling threshold N_p on the performance of the system. In that case the starting level is kept fixed ($S_l = 2$). Next the influence of this starting level S_l is studied. Finally, the impact of using a dynamic scheme is investigated.

8.1 The Influence of the Polling Threshold

Figures 2 and 3 show the influence of the polling feature of our ISA protocol on the mean delay and the throughput. As expected we observe a tradeoff between the delay and throughput characteristics: the sooner the ISA protocol switches to polling, the shorter the mean delay, but the lower the throughput. Moreover this tradeoff depends upon the value of N_p.

From the figures we observe that all three protocols behave very similar with respect to mean delay and throughput when the arrival rate λ is very small (below 0.3). A similar result can be obtained for very large values of λ (beyond 10). Both these results are intuitively clear. Polling is not an issue in these cases: for λ very small collisions rarely occur and hence ISA solves the collisions; if λ is very large the remaining size of the address space is too large to switch to polling. For the used system parameters, the asymptotic values for the mean delay and throughput are 9 and $\frac{128}{252}$.

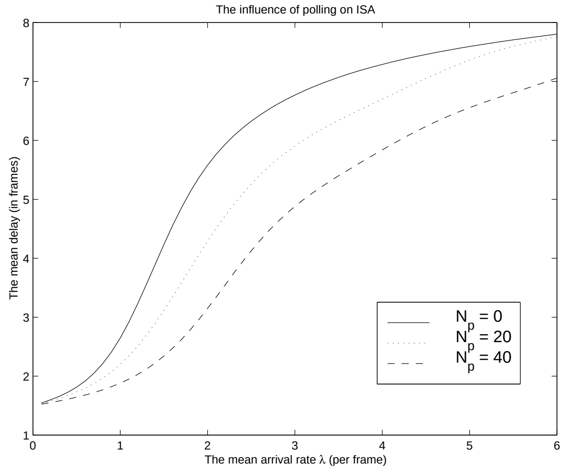

Figure 2: *The impact of Polling on the mean delay*

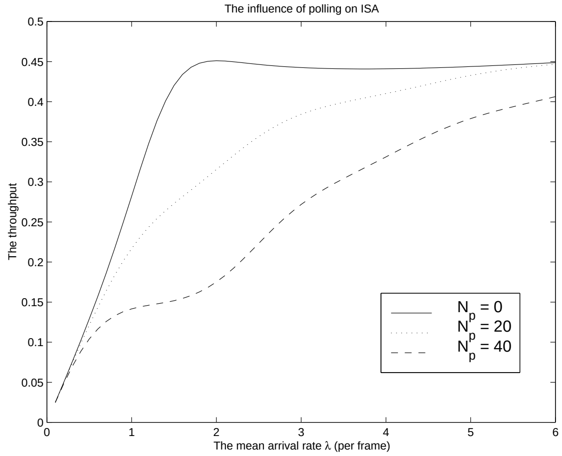

Figure 3: *The impact of Polling on the throughput*

Let us now consider moderate values for λ. Recall that for the polling threshold $N_p = 40$, resp. $N_p = 20$, the protocol will never start polling until level 3, resp. level 4. If the arrival rate increases, the probability that collisions at level 2 (or 3) are introduced increases. In most cases these collisions contain very few participants, and hence occasionally 32 (or 16) polling slots are provided at level 3 (or 4) for very few competitors. Therefore the throughput decreases with increasing value of N_p. If λ is increased even more, polling is postponed in most cases to a later level (as the expected number of collisions at level 2 or 3 becomes larger than one) and will contain more participants. This results in higher throughput values for a fixed value of N_p.

Finally remark that from these figures, the curve that corresponds to $N_p = 40$ is different from the others, merely due to the fact that in this case it is sometimes possible to poll at level 3. Now we investigate whether the conclusion that the mean waiting time decreases due to polling still

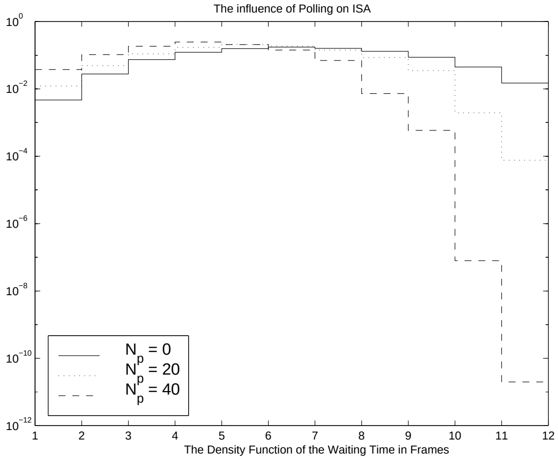

Figure 4: *The ISA with polling delay density function for $\lambda = 3$.*

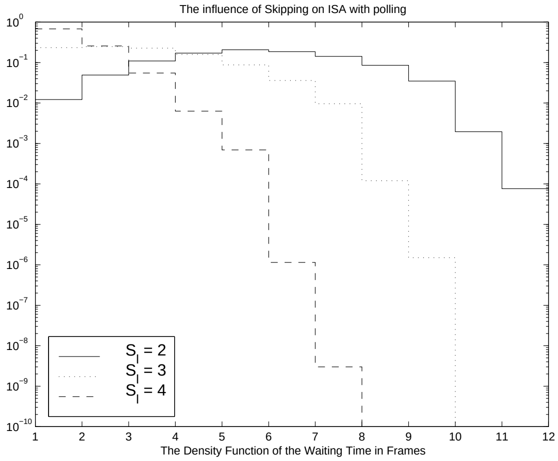

Figure 5: *The impact of Skipping levels on the delay density function with $\lambda = 3$*

holds for the tail of the delay distribution. Figure 4 shows the impact of polling on the delay distribution function for $\lambda = 3$. It illustrates the fact that the main improvement of the delay is located in the tail of the distribution.

8.2 The Influence of Skipping Levels (STATIC)

Figures 6 and 7 illustrate the impact of S_l on the average delay and the throughput. In these figures we have three different types of curves, full, dotted and dashed, corresponding to $S_l = 2, 3$ and 4 respectively. Moreover for each value of S_l the results for $N_p = 0, 20$ and 40 are depicted. For a fixed value of S_l, the upper curve corresponds to $N_p = 0$, the middle curve to $N_p = 20$ and the lowest to $N_p = 40$.

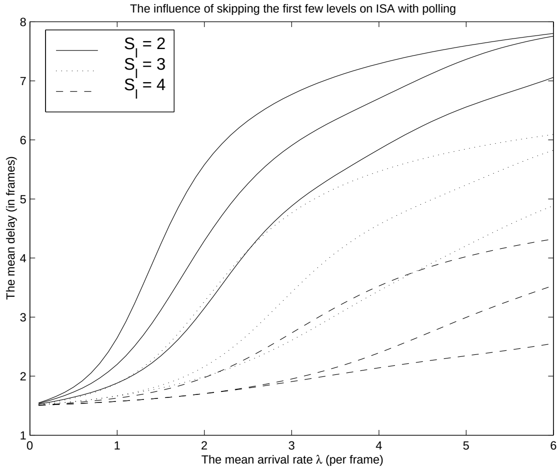

Figure 6: *The influence of skipping on the mean delay.*

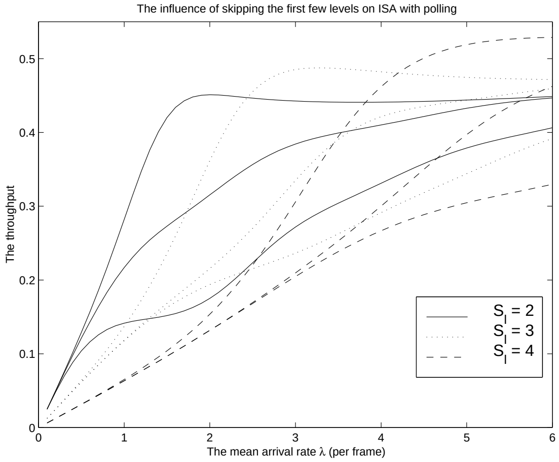

Figure 7: *The influence on the throughput for N_p = 0, 20 and 40.*

Clearly skipping the first levels leads to a decrease of the mean waiting time. The asymptotic value now is $12 - 1.5 \times S_l$. Let us now focus on the impact of polling for variable values of S_l. First Figure 6 shows a faster decrease of the delay due to polling, when the starting level is larger.

This can be seen by observing the area between the curves for $N_p = 0, 20$ and $N_p = 40$. Also notice that the curves converge slower for increasing value of S_l (observe the differences for $\lambda = 6$ in Figure 6).

Secondly, from Figure 7 we conclude that the decrease in the throughput due to polling is smaller for increasing value of S_l. This results from the fact that polling at level 3 (and 4) is no longer possible if $S_l = 3$ (or 4) and thus the number of polling slots provided is smaller but better utilized. Figure 7 represents the throughput results for $N_p = 0, 20$ and 40. We see that for low values of λ skipping levels results in a lower throughput (as most of the S_l slots are wasted). If λ becomes larger this loss is converted in a small gain, due to the fact that the majority of the slots before level S_l contain collisions.

The influence of skipping levels on the delay density function is shown in Figure 5.

8.3 The Influence of Skipping Levels (DYNAMIC)

From the previous sections we may conclude that a higher starting level has a positive impact on the delay especially for larger values of λ. Unfortunately a high price is paid for this in terms of throughput if λ is small. The aim of this section is to show that the dynamic scheme as proposed in section 7.1 solves this problem. That is if λ is small the result should tend to the results for $S_l = 2$ while for λ large the behavior is similar to the one corresponding to $S_l = 4$.

Figures 8 and 9 show that this is the case (for $N_p = 20$), meaning that a system where the levels are skipped dynamically, is able to limit the maximum delay with high throughput.

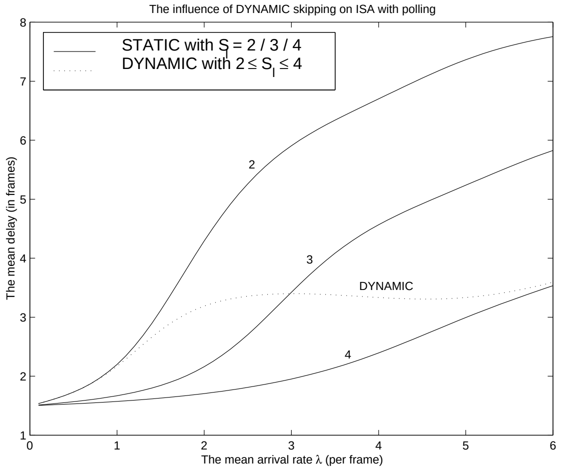

Figure 8: *The influence of dynamic skipping on the delay ($N_p = 20$).*

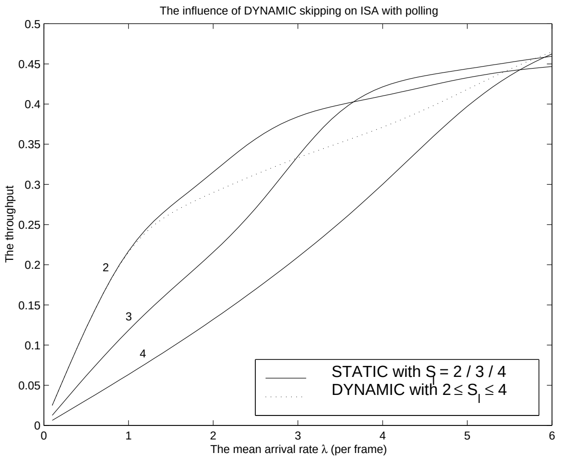

Figure 9: *The influence of dynamic skipping on the throughput for $N_p = 20$.*

9 Conclusions

This paper presents a MAC protocol for a wireless ATM access network supporting the different ATM service categories. The control is centralized and located in the base station. Information exchange between base station and mobile stations is achieved by means of a permit/request mechanism. The controller allocates the available bandwidth to the stations according to the service category the connections belong to. A performance-critical feature of the proposal is the contention resolution protocol used to allow bursty VBR traffic or ABR and UBR traffic to inform the base station about their bandwidth needs. The design of this protocol has an important impact on the performance of the system, in particular delay (for real time VBR services) and throughput (for ABR and VBR services). Using an analytical model, we show that the proposed Identifier Splitting Algorithm combined with Polling, enhanced by a mechanisms of skipping in a dynamical way the first levels, lead to an optimal trade off between low delay and high throughput result.

References

[1] J.I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Trans. Inform. Theory, Vol 25, No 5, pp. 319 - 329*, 1979.

[2] Olga Casals, Jorge Garcia, and Chris Blondia. A medium access protocol for an ATM access network. *Proc. of 5th Int. Conf. on Data Comm. Syst. and their Performance, Raleigh, North Carolina (USA), Oct.*, 1993.

[3] Jean-Pierre Ebert, Ralf Holtkamp, Adam Wolisz, and Louis Ramel. A distributed media access control for wireless ATM environments. *6th WINLAB Workshop on Third Generation Wireless Systems, New Brunswick, NJ, USA, March*, 1997.

[4] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard, and B. Bamamurthi. Packet reservation multiple access for local wireless communications. *IEEE Transactions on Communications, vol. COM-37, pp. 885-890, Aug.*, 1989.

[5] R. Jain, S. Kalyanaramam, R. Goyal, S. Fahmy, and R. Viswanathan. ERICA switch algorithm: A complete description. *ATM Forum document AF-TM-96-1172*, 1996.

[6] Mark J. Karol, Zhao Liu, and Kai Y. Eng. Distributed-queueing request update multiple access (DQRUMA) for wireless packet (ATM) networks. *Wireless Networks, Baltzer Science Publishers, Vol. 1, no III*, 1995.

[7] Jouni Mikkonen, James Aldis, Geert Awater, Andrew Lunn, and David Hutchison. The MAGIC WAND-functional overview. *IEEE Journal on Selected Areas in Communications, Vol. 16, No. 6, August*, 1998.

[8] Frans Panken, Chris Blondia, Olga Casals, and Jorge Garcia. A MAC protocol for an ATM PONs supporting different service cagegories. *Proc. of the 15th ITC, Washington, USA, June 1997, Eds. V. Ramaswami and P. Wirth, Elsevier, Vol 2, pp. 825-834*, 1997.

[9] Nikos Passas, Sarantis Paskalis, Dimitra Vali, and Lazaros Merakos. Quality-of-service-oriented medium access control for wireless ATM networks. *IEEE Communications Magazine, November*, 1997.

[10] Dietmar Petras. Medium access control protocol for wireless, transparent ATM access in MBS. *RACE Mobile Telecommunications Summit, (Cascais, Portugal)*, 1995.

[11] Dietmar Petras and Andreas Kramling. Fast collision resolution in wireless ATM networks. *2nd MATHCOM, Vienna, Austria, Feb*, 1997.

[12] John Porter and Andy Hopper. An ATM based protocol for wireless LANs. *ORL technical report*, 1994.

[13] P.F.M. Smulders and C. Blondia. Application of the asynchronous transfer mode in indoor radio networks. *Proc. of Joint PIMRC and WCN conference 94, The Hague, pp. 839, netherlands*, 1994.

[14] B. Van Houdt and C. Blondia. Performance evaluation of an identifier splitting algorithm with polling for contention resolution in a wireless ATM environment. *in preparation*, 1998.