Performance Evaluation of a MAC Protocol for Wireless ATM Networks Supporting the ATM Service Categories

B. Van Houdt*, C. Blondia*, O. Casals#, J. Garcia# and D. Vazquez*
* University of Antwerp, Dept. Mathematics and Computer Science,
B 2610 Antwerp; Tel : +32-3-8202404, E-mail : {vanhoudt, blondia, dcortizo}@uia.ua.ac.be
Polytechnic University of Catalunya, Computer Architecture Department,
E 08034 Barcelona; Tel : +43-9-3-4016985, E-mail : {olga,jorge}@ac.upc.es

Abstract

This paper presents a Medium Access Control (MAC) protocol for broadband wireless LANs based on the ATM transfer mode, together with the evaluation of its performance in terms of throughput and access delay. Important characteristics of the MAC protocol are the way information between the Mobile Stations (MS) and Base Station (BS) is exchanged and the algorithm used to allocate the bandwidth in order to support the service categories. The performance is heavily influenced by the way the BS is informed about the bandwidth needs of the MSs. In order to obtain an efficient system, a contention resolution scheme based on an Identifier Splitting Algorithm combined with polling is proposed for that purpose, in case no piggybacking can be used. A detailed analytical evaluation, both on cell level and higher layer packet level, is performed, leading to an assessment of the efficiency and the access delay of the system.

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 raised the interest in a technology allowing a wireless (indoor) access to the high-capacity integrated-services wired networks already deployed.

The system we are considering in this paper has the following characteristics. Consider a cell in an ATM network with a diameter of the order of 100m, consisting of a BS serving a finite set of MSs by means of a shared radio channel. The number of MSs is variable, as new MSs may enter the cell and others may leave it. 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 an MS are broadcasted downlink. ATM PDUs originating from an MS share the uplink radio channel using a well defined access protocol. 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 attributes to each MS a MAC address consisting 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.

The wireless MAC protocol considered in this paper was introduced in [5]. A key element of the proposed MAC protocol is the *Identifier Splitting Algorithm with Polling*, the contention resolution mechanism used to let an MS inform the BS about its bandwidth needs when no piggybacking is possible. We define an analytical model which allows the computation of the corresponding delay bounds and throughput results for ATM PDUs which have to rely on this scheme using the results obtained in [4]. In addition we consider traffic generated by higher layers in the MS and propose an analytical model to compute the packet access delay caused by the proposed MAC protocol. We investigate the influence of the traffic characteristics on the efficiency of the protocol and on the delay of upstream packets.

Section 2 introduces the MAC protocol together with the Identifier Splitting Algorithm with Polling. In Section 3, the performance critical part of the protocol, namely the ISA scheme with Polling is evaluated and the impact of system parameters on the delay and the throughput are investigated. Section 4 proposes a queueing model to derive packet level performance measures. In particular, the analytical model leads to an assessment, using the results of Section 3, of the delay a packet of a higher layer experiences due to the access protocol. Also the efficiency to the protocol is evaluated. This model is used in Section 5 to illustrate the influence of the parameters of the packet arrival process on the performance of the protocol. Conclusions are drawn in Section 6.

2 The MAC Protocol Description

2.1 Information Exchange between MS and BS and Frame Structure

In this section we describe the information exchange between an MS and the BS. Assuming there is a MAC protocol with centralized controller located in the BS, each MS must be able to inform the BS about its bandwidth needs (requests) and the BS should be able to inform the MS about the received bandwidth (permits). A more detailed description is found in [5].

2.1.1 Permits

In order to be allowed to use the uplink channel, the MS has to receive a *permit* from the BS. A permit (4 bytes) contains: the address of the permit's destination MS (2 bytes), the service category of the connection receiving the permit (2 bits) and 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) (14 bits).

2.1.2 Requests

The MS declares its bandwidth needs to the BS by means of requests. A request (8 bytes) contains: the address of the MS that is issuing the request (2 bytes) and, 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 mechanisms to send requests: either by piggybacking or by using the contention resolution protocol (in which case a slightly different format is used, see Section 2.2.1). Depending on the service category, a combination of these mechanisms is used to declare the bandwidth needs of the MS.

2.2 Frame Structure

2.2.1 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.

U1 slot (106 bytes): 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.

U2 slot (106 bytes): a U2 slot is used to allow bursty VBR, ABR and UBR connections to inform the BS about the need for a permit. It consists of 4 minislots 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), a queue length indication of 6 bits for this service category, while the remaining 188 bits (848 / 4 - 16 - 8) 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.

2.2.2 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 (88 bytes): this slot contains a downstream ATM cell (53 bytes), accompanied by the necessary physical layer overhead (training sequence, error detection: 35 bytes). Each downlink frame contains 80 D1 slots.

D2 slot (160 bytes): 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, leading to an important power consumption reduction.

D3 slot: this slot is used to inform the MS about the permission to transmit a cell in the next upstream frame. It contains a variable number of permits, between 72 and 80. Each permit requires 4 bytes, hence the length of a D3 slot takes a value between 316 bytes and 288 bytes.

D4 slot (2 bytes): 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 is coded in 3 bits.

D5 slot (4 bytes): 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 whether it was successful or not.

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.

2.3 The Request Mechanism

2.3.1 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 Permit Distribution Algorithm (see Section 2.3) 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.

2.3.2 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. However, this scheme fails in case the last upstream cell leaves behind empty buffers and the VBR connection is still active (i.e. it 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*.

The Identifier Splitting Algorithm (ISA) with Polling The ISA protocol was introduced by Petras in [3]. It is a dynamic collision resolution algorithm that belongs to the class of the tree algorithms but where the splitting is based on the MAC addresses of the MSs instead of the more common coin flip procedure. 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 generated by an MS during a CC intending to use the contention resolution scheme have to wait for participation until the start of the

next CC. In the first frame of a cycle four contention minislots are available (one entire slot), which can be used for contention resolution (we start at level 2 of the tree since we have 2^2 minislots). The 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.

Two situations are possible: either an MS sending in slot k, $1 \leq k \leq 4$, was successful and will eventually be granted a permit by the BS to send an upstream cell, or there will be a new attempt in the next (second) frame of the CC. This frame provides $2 \times l$ minislots if there where l minislots $0 \leq l \leq 4$, with collisions. The involved MSs apply the same scheme again and each time the next bit of the MAC address is used to decide which of the two minislots is used to resolve the collision. Due to the delayed feedback, as 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.

The BS obtains more and more knowledge about the address ranges of the MSs that are still competing. Therefore, 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. Using the feedback information, every MS by itself can perform the necessary calculation(s), to find out whether the polling starts and which minislot to use.

Let us now consider the ISA scheme when a number of levels is skipped (i.e. more than 4 minislots are provided 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 3.2, this results in some extra throughput losses during silent periods. Therefore, we propose a scheme that changes the starting level S_l dynamically, between level S_{min} and S_{max}, depending on the length of the previous contention cycle. 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.

2.3.3 Request Mechanism for ABR and UBR

Again a piggybacking mechanism is preferable, but when not possible (see the conditions in 2.2.2) the Identifier Splitting Algorithm (with or without polling) can be used to allow these connections to declare their bandwidth needs.

2.4 The Bandwidth Allocation Algorithm

The bandwidth allocation algorithm has to distribute permits among the active connections based on: the service class the connection belongs to, the individual contract parameters of the connection and the current state of the different queues in the MS (i.e. the bandwidth requirement the MS has for each service category), see also [2].

CBR connections. The permits for CBR traffic are generated according to 2.2.1 and put in a "CBR/rt-VBR FIFO" queue. For each MS (with an active CBR connection), the central controller maintains a real valued 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 (PEI) of the CBR connection. At each slot it is count down by 1 until zero (or below). When the value reaches zero (or below), a permit is generated for that CBR connection and the counter is incremented by $CBR\_CDC(Init)$. The CBR/rt-VBR queue is emptied with the highest priority: when determining the contents of the D3 slot, the CBR/rt-VBR FIFO queue is checked and if not empty, permits are added to the list of permits in the D3 slot on a FIFO basis.

rt-VBR connections. The requests that are received according to 2.2.2 for rt-VBR traffic are converted into permits, and are put into the CBR/rt-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 more detail, the BS maintains three real valued counters for every rt-VBR connection: a Count Down Counter rt-VBR_CDC, a Request Counter rt-VBR_REQ and a Leaky Bucket Counter rt-VBR_LBC.

The rt-VBR_CDC is initialized at $VBR\_CDC(Init)$ equal to the number of slots corresponding to the Peak Emission Interval (PEI) of the rt-VBR connection. At the end of each slot it is count down by one. When the value reaches zero (or below), the value of the rt-VBR_REQ is checked (we use real values for the counters to support connections with a fractional PEI).

- If the rt-VBR_REQ is zero the Count Down process is put on a hold.
- Otherwise a permit is generated if it is conform to the traffic contract. To check the conformance the rt-VBR_LBC is maintained. If the permit is not conform its generation is postponed until a conform time instance and the Count Down process was put on a hold.

For the analytical model it is assumed that the Maximum Burst Size (MBS) is not exceeded and therefore the influence of the rt-VBR_LBC is not taken into account.

When a permit is generated the rt-VBR_REQ is decreased by one, while the rt-VBR_CDC is increased with $VBR\_CDC(Init)$ and if necessary the Count Down process is reactivated (i.e. starts counting down again).

The rt-VBR_REQ reflects the number of waiting cells at the MS and is updated every time a request arrives. If, during an update, the old value of the rt-VBR_REQ is zero and if the Count Down process is put on a hold, the Count Down Process is reactivated.

In the CBR/rt-VBR FIFO queue the permits for rt-VBR traffic compete (among each other and with the CBR permits) on a FIFO basis.

nrt-VBR connections. For nrt-VBR connections we use the same method as for rt-VBR connections, except that the permits are forwarded to the "nrt-VBR FIFO" permit queue. This queue has the second priority.

ABR and UBR connections. The requests that are received according to 2.2.3 for ABR traffic are first stored per MS into an ABR-REQ Counter maintaining 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 third priority. The requests that are received according to 2.2.3 for UBR traffic are treated in a similar way as the ABR requests, but written to a UBR FIFO queue (fourth priority).

3 Performance Evaluation of the ISA scheme with Polling

3.1 System Description

An analytical model to compute the delay distribution and the throughput when using the ISA with polling has been developed in [4] and is presented briefly in what follows.

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 an MS connects to the BS, possibly due to a handover, a unique MAC address is assigned in a random way similar to the procedure to generate the flow label in IPv6. 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 lying beyond the value of 2^n is added to that of 2^n to make the distribution finite. 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 model more 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. 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.

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

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

where B_l and B_m are two predefined values.

3.2 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, 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 starting level S_l varies from level 2 to 4, corresponding to 4 to 16 minislots (or 1 to 4 slots) and when studying a system with a dynamic starting level, B_l and B_m are set to 1 and 4 respectively. The boundary values are $S_{min} = 2$ and $S_{max} = 4$.

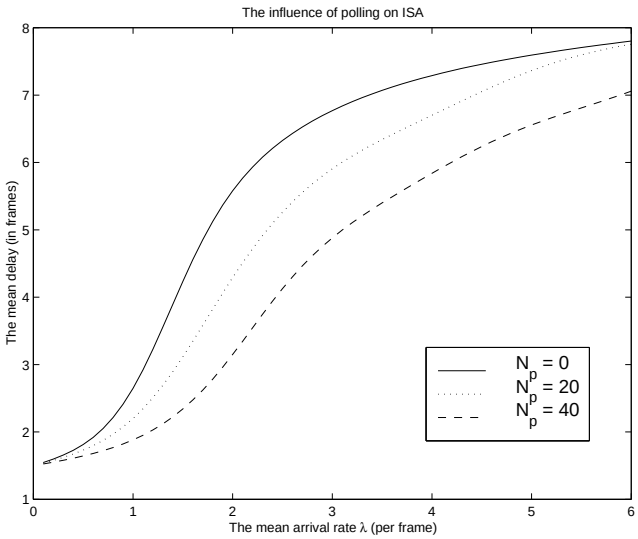

Figure 1: The impact of Polling on the mean delay

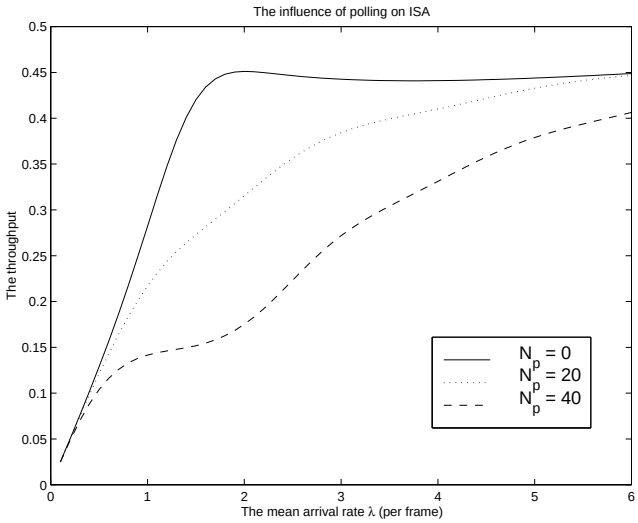

Figure 2: The impact of Polling on the throughput

3.2.1 The Influence of the Polling Threshold

Figures 1 and 2 show the influence of the polling feature of the 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. This tradeoff depends upon the value of N_p.

3.2.2 The Influence of Skipping Levels

Figures 3 and 4 illustrate the impact of S_l on the average delay and the throughput. From these results, 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. Figures 3 and 4 show (for $N_p = 20$) that the dynamic scheme as proposed in Section 2.2.2. solves this problem.

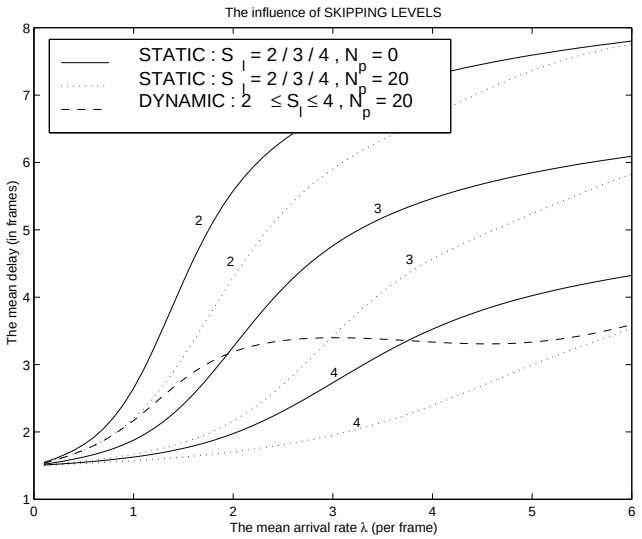

Figure 3: The influence of skipping on the mean delay.

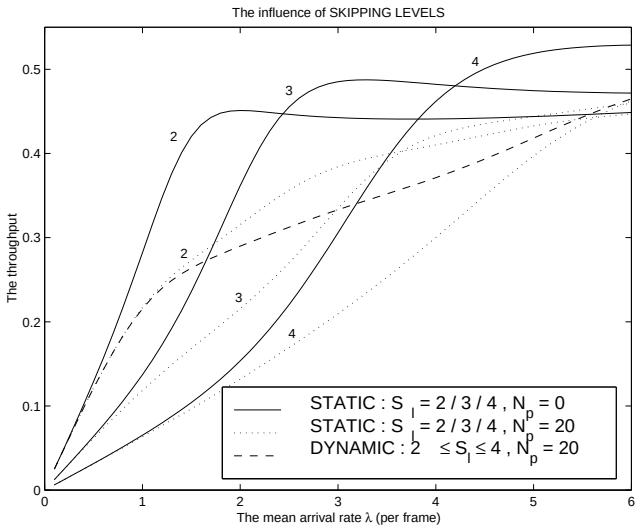

Figure 4: The influence on the throughput for $N_p = 0$ and 20.

4 Packet Level Performance Characteristics

4.1 System Description

We assume that the traffic offered by the higher layer protocols (above ATM/AAL) is in the form of packets. These packets are segmented at the AAL layer in a number of ATM cells. In general this type of traffic is very well suited for a piggybacking scheme, because a single request suffices to notify the BS of the presence of all the ATM PDUs belonging to a packet. Clearly there are two possibilities to transmit this request. First it could be that the packet is generated before the last PDU(s) of the previous packet(s) are transmitted, in which case piggybacking is used. Otherwise the request has to be delivered to the BS using the contention channel, for which the ISA protocol combined with polling is employed. Moreover, the following assumptions are made:

- The real time permit queue, containing the CBR and rt-VBR permits, cannot be congested. This is easily guaranteed by making sure that the sum of the peak cell rates of all the real time connections is less than the link rate. Congestion is allowed in all permit queues other than the CBR/rt-VBR permit queue.
- In the protocol definition the frame length is fixed, while the number of $U1$ ($U2$) slots in such a frame is variable and determined by the ISA protocol. For the analysis, we assume that the number of $U1$ slots is fixed to F, while the number of $U2$ slots is still determined by ISA, resulting in a slightly varying frame length. As frames contain mostly $U1$ slots (the number of $U1$ slots is between 72 and 80), this assumption should hardly have any influence on the results.

The system described above is modeled using a single server discrete time queue. The queue is fed by fixed length packets, although it is easy to incorporate any type of distribution for the packet lengths. The service process of this queue maintains a counter. This counter is incremented by one every time unit and is reset to zero when it reaches a value of F. Clearly it corresponds to the position within one frame in terms of $U1$ slots.

Furthermore, the discrete time process that governs the packet arrivals has a different time scale as the service process has. One time unit for the arrival process corresponds to Q time units for the service process, with Q a divisor of F. Therefore packet arrivals can only occur if the counter value of the service process is divisible by Q. Ideally Q equals one, meaning that arrivals can occur at any time instance.

Finally, the service time of a single packet depends upon: the packet length L, the PCR of the rt-VBR connection, the delay distribution W of the contention channel, the remaining service time of the preceding packets and the counter value of the service process. The first three values are the same for all packets.

For the system to be analytically tractable, the packets generated at the MS must be sufficiently large such that the time in-between the generation of the first and last permit destined for a packet of length L, i.e. $\frac{L-1}{PCR}$, is at least one frame time.

4.2 The Packet Arrival Process

The arrival process in an MS is a discrete-time Markov process belonging to the class of D-MAPs, very similar to the Markov Modulated Bernoulli Processes (MMBP). An m-state arrival process that belongs to this class is characterized by a rate vector $(\beta_1, \ldots, \beta_m)$, that contains the mean arrival rate associated with each of the m states, and an $m \times m$ transition matrix $\mathbf{D}$ that governs the transition probabilities between states. The matrix $\mathbf{D}$ can be written as the sum of two matrices $\mathbf{D}_0$ and $\mathbf{D}_1$:

- $(\mathbf{D}_0)_{i,j}$ equals the probability that no arrival occurs and a transition from state i to state j takes place.
- $(\mathbf{D}_1)_{i,j}$ equals the probability that an arrival does occur and a transition from state i to j takes place.

As opposed to the MMBPs, transitions between states are only allowed at arrival times. Therefore $\mathbf{D}_0$ is a diagonal matrix $diag(1-\beta_1, \ldots, 1-\beta_m)$. Next we denote $(\hat{\mathbf{D}}_1)_{i,j}$ as the probability that a transition occurs from state i to j under the condition that an arrival occured. Thus we have the following relation $(\hat{\mathbf{D}}_1)_{i,j} = \frac{(\mathbf{D}_1)_{i,j}}{\beta_i}$. Finally the $m \times m$ matrices $\mathbf{A}_i^{(n)}$ contain the probabilities of having i arrivals during n time units of the arrival process. By only allowing state transitions at arrival times we get the

following distribution for I, the interarrival time:

$$P[I=k] = \vec{\alpha}_1 \begin{pmatrix} (1-\beta_1)^{k-1}\beta_1 \\ \vdots \\ (1-\beta_m)^{k-1}\beta_m \end{pmatrix}, \qquad (2)$$

where $\vec{\alpha}_1$ is the left stochastic steady state vector of $\hat{\mathbf{D}}_1$. What makes this arrival process interesting is that we can change the correlation between consecutive interarrival periods in a systematic way without changing the distribution I. By definition this correlation equals

$$corr = \frac{E[I_n I_{n+1}] - E[I_n]E[I_{n+1}]}{\sqrt{VAR[I_n]}\sqrt{VAR[I_{n+1}]}}. \qquad (3)$$

The mean value $\mu = E[I_n] = E[I_{n+1}]$ and the variation $\sigma^2 = VAR[I_n] = VAR[I_{n+1}]$ are found in a straightforward way, while $E[I_n I_{n+1}]$ is found using the partial derivatives of the joint generating function $f(z_1, z_2) = \sum_i \sum_j P[I_n = i \cap I_{n+1} = j] z_1^i z_2^j$. Hence,

$$E[I_n I_{n+1}] = \vec{\alpha}_1 \; diag(1/\beta_1^2, \ldots, 1/\beta_m^2) \; \mathbf{D}_1 \begin{pmatrix} 1/\beta_1 \\ \vdots \\ 1/\beta_m \end{pmatrix}.$$

To obtain this result we made use of the following identity $\sum_{i\geq 1} i(\mathbf{D}_0)^{i-1} = diag(1/\beta_1^2, \ldots, 1/\beta_m^2)$.

We now show how to change the correlation in a systematic way without changing the distribution I. We define an infinite set of arrival processes $A(r), r \geq 1$ with the same rate vector $(\beta_1, \ldots, \beta_m)$. The matrices $\mathbf{D}_0$, $\mathbf{D}_1$ and $\hat{\mathbf{D}}_1$ corresponding to the process $A(r)$ are denoted by $\mathbf{D}_{0,r}$, $\mathbf{D}_{1,r}$ and $\hat{\mathbf{D}}_{1,r}$. The matrix $\mathbf{D}_{0,r}$ is the same diagonal matrix for all these processes. $\mathbf{D}_{1,r}$ is defined as

$$(\mathbf{D}_{1,r})_{i,j} = \frac{(\mathbf{D}_{1,1})_{i,j}}{r} \qquad i \neq j$$
$$(\mathbf{D}_{1,r})_{i,i} = \beta_i - \sum_{j\neq i} (\mathbf{D}_{1,r})_{i,j}.$$

Thus, all arrival processes $A(r)$, $r > 1$, are determined by the choice of $A(1)$. It is easy to show that the interarrival distribution $I(r)$ is the same for all the processes $A(r)$. On the other hand, the correlation between successive interarrival periods increases with increasing r.

The sustainable cell rate SCR and the peak cell rate PCR for this arrival process are chosen as $SCR = L/(\mu Q)$ and $PCR = L \max_i \beta_i / Q$, where L equals the packet length and μ is the mean packet interarrival time. Clearly both these values are expressed in time units of the server process.

4.3 The Service Time

The analysis is performed on a packet level, as we are only interested in the service time of packets and not in the service time of individual ATM PDUs.

From the definition of the traffic scheduler in the BS, the permits for the different cells belonging to the same packet are placed in the CBR/rt-VBR permit queue according to the PCR of the connection. From here on the CBR/rt-VBR permit queue is simply called the permit queue, except when stated otherwise. The presence of the permit queue introduces some jitter into the distance between permits belonging to the same packet. As a result their corresponding cells are not exactly transmitted at the PCR.

As we observe the queue on a packet level, we are not interested in the interdeparture times of consecutive cells of a packet, but only in the interdepature time of the first and the last cell of a packet. As we assumed that the permit queue is never congested, we can approximate this interdeparture time by the time in-between the placing of the corresponding permits in the permit queue (which is especially true for longer packets).

To find the service time of a packet, the following two observations must be made:

- Piggybacking is possible if a packet finds a non empty transmission queue upon arrival, otherwise the MS makes use of the contention channel.
- During frame n, the BS schedules the uplink transmissions for frame $n+1$. Therefore once the BS is notified of a cell arrival at the MS, at least a full frame length passes before the actual transmission can occur.

Therefore, we distinguish three scenarios:

Scenario 1: The packet finds the transmission queue empty upon arrival. Piggybacking is no longer an option and the contention channel is used. Once the request is successfully transmitted, at least one frame time will elapse before the first cell is transmitted (see 2). Therefore, the service time S_1 is chosen as follows: $S_1 = R + W + F + \frac{L-1}{PCR}$. The random variable R denotes the remaining time until the counter of the service process reaches zero again, W is the delay suffered on the contention channel (a multiple of F), F is a fixed value that corresponds with one frame, L is the packet length and PCR the peak cell rate of the connection.

Scenario 2: The packet arrives in a non-empty transmission queue but the remaining service time of the preceding packet(s) is smaller than one frame time. Due to the assumption on the packet length L, this scenario can never occur if more than one packet is backlogged at the MS. Taking observation 2 into account, all preceding cells are scheduled for transmission by the BS. Thus the service time S_2 of this packet depends on the remaining service time R_S of the preceding packet and is defined as $S_2 = F - R_S + \frac{L-1}{PCR}$.

Scenario 3: The packet arrives in a non-empty transmission queue and the remaining service time for the packet(s) in front is at least a frame time. Therefore, not all preceding cells have been scheduled for transmission, otherwise the remaining service time would be less than a frame time. Also due to the assumption on the packet length L ($\frac{L-1}{PCR}$ is bigger than F) we define the service time S_3 by $S_3 = \frac{L}{PCR}$.

4.4 Solving the Queueing Model

By observing the system at the time instants O_n that correspond with the transmission epochs of the first cell of packet n, we can describe the system by the vector (N_n, P_n, q_n), where N_n denotes the number of backlogged packets (the one being served is not accounted for), q_n is the phase of the arrival process ($1 \leq q \leq m$) and P_n is the value of the counter associated with the service process at time O_n. To further reduce the state space P_n is rounded to the nearest multiple of Q and therefore can be denoted as a value between 0 and $F/Q - 1$ (again the most accurate results are obtained with $Q = 1$). Furthermore, due to the approximations made in section 4.1 to 4.3, a Markov chain of the M/G/1-type [1] can be obtained. Solving this Markov chain results in the stationary probability vector of the process

at these epochs O_n. Next, we calculate the queue length distribution at the service completion times as follows:

$$P(Q=k) = \sum_{P=0}^{F/Q-1} \sum_{i=0}^{k} \vec{\pi}_i(P) \mathbf{A}_{k-i}^{([\frac{L-1}{PCR}\frac{1}{Q}])} \vec{e},$$

where $\vec{\pi}_i(P)$ is a row vector of length m that contains the stationary probabilities of being in the states $(i,P,j)_{j=1}^{m}$ and $\vec{e}$ is a column vector of size m with elements equal to 1. This is a consequence of the fact that the remaining service time at the observed epochs O_n equals $\frac{L-1}{PCR}$. Clearly, with probability $P(Q=0)$ a packet needs to make use of the uplink contention channel.

Moreover, one can show that for an infinite capacity FCFS stationary discrete time queue with no simultaneous departures or arrivals, both the queue length distribution at the departure times and the arrival times are identical. Thus $P(Q=k)$ is also the probability that a packet finds k customers (packets) in front upon arrival.

5 Numerical Results

In this section we study the influence of the SCR, the PCR and the correlation of the lengths between consecutive interarrival periods on a number of performance measures of an MS holding a single rt-VBR connection. The system parameters for the ISA protocol are set as follows (see [4]): the number of mobile stations considered is 128, the aggregated arrival process of all the MSs on the contention channel is Poisson with a mean of $\lambda = 1$ request per frame, the starting level S_l is static and equal to two, the value N_p that triggers the polling mechanism is 20 and a single instance of the ISA protocol is used.

To find the delay distribution W we refer to [4]. Apart from the piggybacking probability we define the following two performance measures:

$$\begin{aligned} E &= \frac{L/PCR}{P_n L/PCR + P_0(L/PCR + E[W])} \\ D &= P_0 E[W] + P_n E[Q-1 \mid Q>0] L/PCR, \end{aligned}$$

with $P_0 = P(Q=0)$ and $P_n = P(Q>0)$. The first E is a measure for the efficiency of the scheme, the second D is a measure of the delay experienced by packets in the MS.

The system parameter F is fixed at 72. Ideally the parameter Q should be set at 1, meaning that arrivals can occur at any time instance and the frame position P is represented by its true identity (and is not rounded to the nearest multiple of Q). On the other hand, the smaller Q becomes the bigger the block matrices $\mathbf{Q}_{m,n}$ become and the more of them are different from zero making the analytical model less attractive. Therefore, we set $Q=8$ to improve the efficiency of the model. Numerical investigations (not reported here) have shown that the results for smaller values of Q are very well approximated by the model with $Q=8$.

5.1 The Influence of the SCR and the PCR

The packet length L in figure 5 is set to 20, the rate vector β and the transition matrix $\mathbf{D}$ are the following:

$$\beta = \begin{pmatrix} y & xy \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 1-\frac{2xy}{5} & \frac{2xy}{5} \\ \frac{xy}{5} & 1-\frac{xy}{5} \end{pmatrix} \qquad (4)$$

with $1/10 \le y \le 1/200$ and $x = 5/6, 2/3, 1/2$ and $1/2.7272$. When x is fixed and y changes the SCR and the PCR are varied proportionally, thus the ratio $\frac{SCR}{PCR}$ is fixed. However, varying x with a fixed y results in a fixed PCR but a variable SCR.

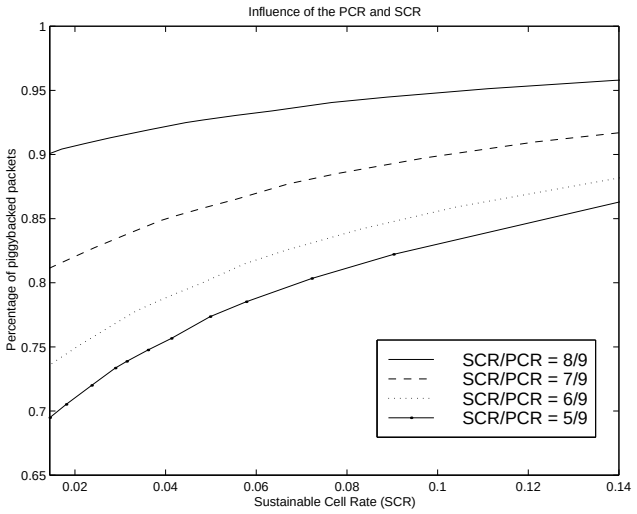

Figure 5: The impact of the SCR and the PCR on $P(Q=0)$

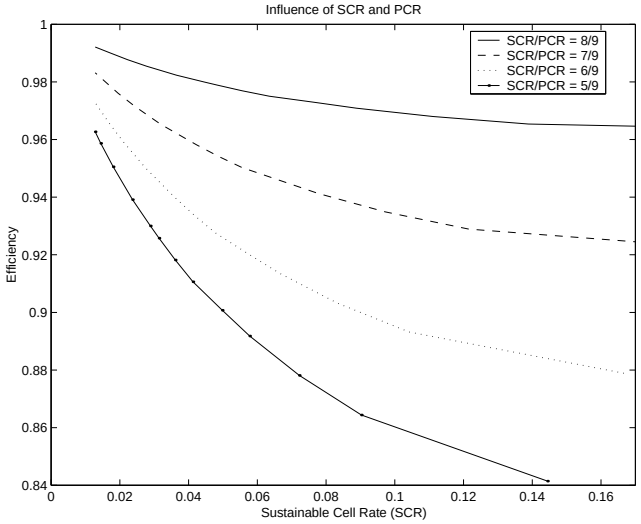

Figure 6: The impact of the SCR and the PCR on E

Figures 5 and 6 show that for a fixed SCR more piggybacking and a better efficiency E is obtained as the ratio $\frac{SCR}{PCR}$ grows. Notice that this ratio is an indication of the burstiness of the traffic source. Secondly, although rt-VBR sources with a higher SCR achieve a higher piggybacking percentage for fixed $\frac{SCR}{PCR}$ ratios (an effect that increases with lower $\frac{SCR}{PCR}$ ratios), their efficiency E is smaller. Figure 7 shows that better delays are achieved as the ratio $\frac{SCR}{PCR}$ decreases, a rather logical result as this ratio is a measure of the load of the scheme. Secondly the delay decreases as the SCR increases, because the workload is offered more gradually by higher bitrate sources.

5.2 The Influence of Correlation

In this section we study the influence of the correlation between the length of consecutive interarrival periods while

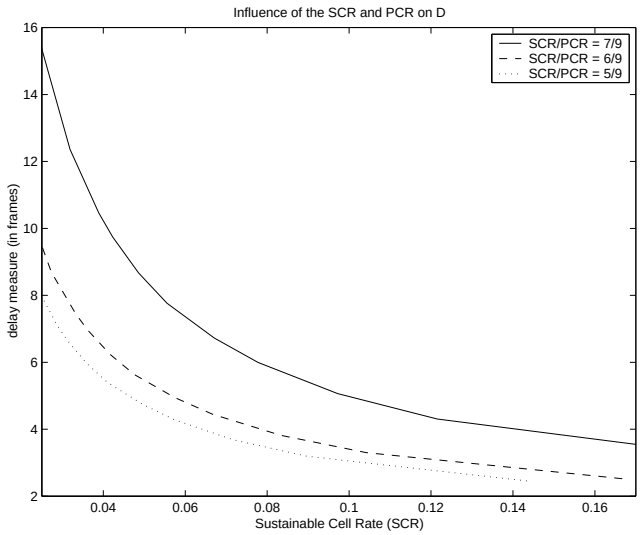

Figure 7: The impact of the SCR and the PCR on D

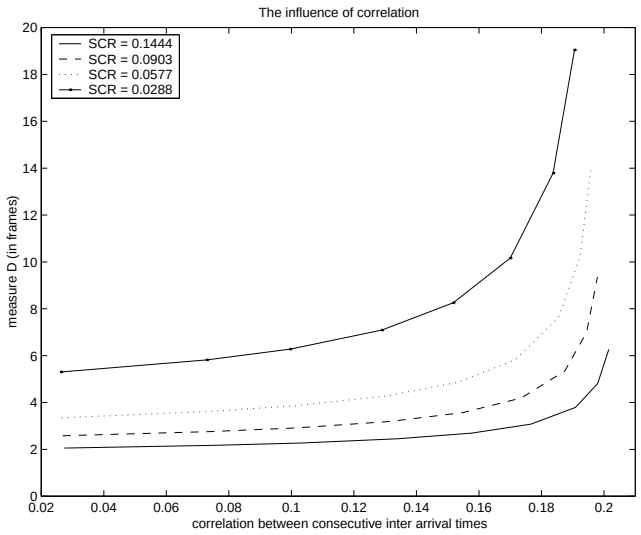

Figure 9: The impact of the correlation on the delay D

keeping the interarrival distribution I fixed. In Section 4.1 we developed a framework that allows us to do so. In this framework the arrival process $A(1)$ is chosen as the one that we used in Section 5.2 with x fixed at $1/2.7272$ thus $\frac{SCR}{PCR} = \frac{5}{9}$. We consider five different values for y resulting in as many different SCRs. Although, in Figure 9 the parameter r is varied from 1 to infinity, the correlation does not go to one when r approaches infinity because of the geometric nature of the arrival scheme.

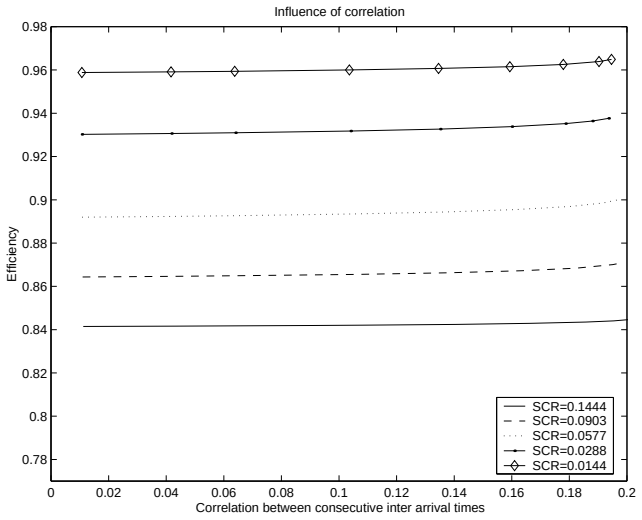

Figure 8: The impact of the correlation on the Effciency E

Figure 8 shows that this type of correlation is less important when studying the piggybacking capabilities or the efficiency of an MS. However, in Figure 9 it is shown that the correlation does have an important impact on the delay. Indeed, more correlation leads to longer delays especially for low bit rate traffic.

6 Conclusions

A MAC protocol for a wireless ATM network using a variant of the ISA algorithm to inform the BS about the bandwidth needs of the MSs was proposed. Since the ISA scheme is a key performance issue of this protocol, we evaluate its performance in detail using an analytical model. We show that the proposed scheme, enhanced by a mechanism of dynamically skipping the first levels, may lead to an optimal trade-off between a low delay and a high throughput. A second analysis is devoted to the evaluation of the influence of the packet arrival process characteristics in the MSs on the efficiency of the protocol and on the delay packets in the MS experience to access the shared medium. In particular, the impact of the following parameters on the packet access delay and the protocol efficiency is investigated: the cell-level ATM traffic parameters, PCR and SCR, and the correlation between packet interarrival times.

References

[1] M.F. Neuts. Structured stochastic matrices of M/G/1 type and their applications. *Marcel Dekker Inc, New York*, 1989.

[2] F. Panken, C. Blondia, O. Casals, and J. Garcia. A MAC protocol for an ATM PONs supporting different service cagegories. *Proc. of the 15th ITC, Washington, USA, June, Vol 2, pp. 825-834*, 1997.

[3] D. Petras and A. Kramling. Fast collision resolution in wireless ATM networks. *2nd MATHCOM, Vienna, Austria, Feb*, 1997.

[4] B. Van Houdt and C. Blondia. Performance evaluation of an identifier splitting algorithm with polling for contention resolution in a wireless ATM environment. *Submitted for publication*, 1999.

[5] B. Van Houdt, C. Blondia, O. Casals, J. Garcia, and D. Vazquez. A MAC protocol for wireless ATM systems, supporting the sevice categories. *Proc. of the 16-th ITC, Edinburgh, UK*, 1999.