

# Packet Level Performance Characteristics of a MAC Protocol for Wireless ATM LANs

B. Van Houdt\*, C. Blondia\*, O. Casals# and J. Garcia#

\* University of Antwerp, Dept. Mathematics and Computer Science,  
B 2610 Antwerp; Tel : +32-3-8202404, E-mail : {vanhoudt, blondia}@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 determines packet level performance measures of a MAC protocol for a wireless ATM local area network. A key characteristic of the MAC protocol is the Identifier Splitting Algorithm with Polling, a contention resolution scheme used to inform the Base Station about the bandwidth needs of a Mobile Station when no piggybacking can be used. We consider higher layer packets that are generated at the Mobile Station and investigate the influence of the traffic characteristics of the packet arrival process on the efficiency of the protocol and on the delay that packets experience to access the shared medium.*

## 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 that supports mobility, realizing access to the wired ATM network. ATM PDUs arriving in the BS with destination on an MS are broadcasted downlink. ATM PDUs originating from an MS share the uplink radio

channel using a well defined access protocol. The net 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 [7]. 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. The corresponding delay bounds and throughput results for ATM PDUs that have to rely on this scheme have been derived analytically in [6]. In this paper we extend this analysis by considering traffic that is offered by higher layer protocols as packets, consisting of a number of back-to-back ATM PDUs. 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. Section 3 proposes a queueing model to derive the performance measures of interest. This model is used in Section 4 to illustrate the influence of the parameters of the packet arrival process on the performance of the protocol. Conclusions are drawn in Section 5.

## 2 The MAC Protocol Description

### 2.1 Information Exchange between MS and BS and Framestructure

In this section we describe the information exchange between an MS and the BS. Assuming a MAC proto-

col 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 [7].

### 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 ways to send requests: either by piggybacking or by using the contention resolution protocol (in the second case a slightly different format is used, see Section 2.1.3). Depending on the service category, a combination of these mechanisms is used to declare the bandwidth needs of the MS.

### 2.1.3 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.

### 2.1.4 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 slots 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.

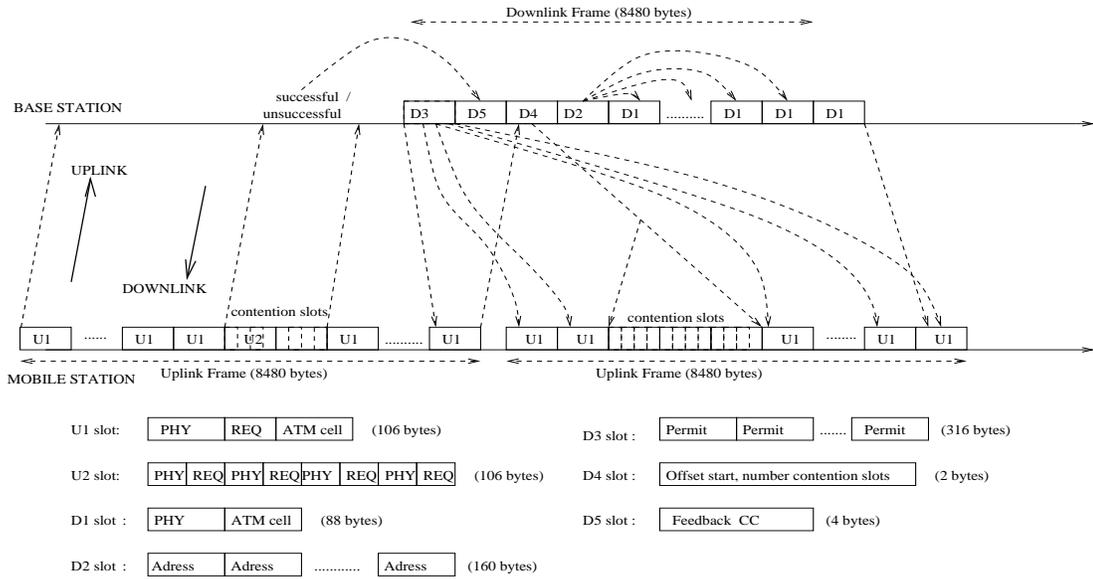


Figure 1: *The Frame Structure*

## 2.2 The Request Mechanism

### 2.2.1 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.

### 2.2.2 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 service as this introduces a minimal overhead. However, this scheme fails in the case where 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 [4]. 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. Also, due to the frame structure the tree is traversed in a breadth-first fashion and has a varying number of contention slots per frame. For more details we refer to [3, 4]. During the ISA procedure, 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. Moreover, the first few levels can be skipped in a static or dynamic way. A detailed description of the protocol is found in [6].

### 2.2.3 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.

## 2.3 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 [1].

**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 counted 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 counted 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 conforms to the traffic contract. To check the conformance the *rt-VBR\_LBC* is maintained. If the permit is not in conformance its generation is postponed until a conform time instance (determined by the *rt-VBR\_LBC* counter) and the Count Down process is put on 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 and the *rt-VBR\_CDC* is increased by  $VBR\_CDC(Init)$  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 was put on 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 System Model

#### 3.1 System Description

We assumed 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 into a number of ATM PDUs. 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 this 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 at zero when it reaches a value of  $F$ . Clearly it corresponds with 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 from the service process. 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.

### 3.2 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  $(\beta_1, \dots, \beta_m)$ , 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  $diag(1 - \beta_1, \dots, 1 - \beta_m)$ . 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 occurred. Thus we have  $(\hat{\mathbf{D}}_1)_{i,j} = (\mathbf{D}_1)_{i,j} / \beta_i$ . 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:

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

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

$$corr = \frac{E[I_n I_{n+1}] - E[I_n]E[I_{n+1}]}{\sqrt{VAR[I_n]} \sqrt{VAR[I_{n+1}]}}. \quad (2)$$

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 by straightforward reasoning.

$E[I_n I_{n+1}]$  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,

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

To obtain this result the identity  $\sum_{i \geq 1} i (\mathbf{D}_0)^{i-1} = \text{diag}(1/\beta_1^2, \dots, 1/\beta_m^2)$  was used.

We now demonstrate how to one can 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  $(\beta_1, \dots, \beta_m)$ . 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

$$(\mathbf{D}_{1,r})_{i,j} = \frac{(\mathbf{D}_{1,1})_{i,j}}{r} \quad i \neq j$$

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

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 follows:  $SCR = L/(\mu Q)$ ,  $PCR = L/O \max_i \beta_i$ . Where  $L$  equals the packet length and  $\mu$  is the mean packet interarrival time. Clearly both the  $PCR$  and the  $SCR$  are expressed in time units of the server process.

### 3.3 The Service Time

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

By 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 interdeparture time of the first and the last cell of a packet. Because we assumed that the permit queue is never congested, we can approximate this interdeparture time by the time between the placing of the corresponding permits in the permit queue; this is especially true for longer packets.

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

1. Piggybacking is possible if a packet finds a non empty transmission queue upon arrival, otherwise the MS makes use of the contention channel.
2. 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:  $S_1 = R + W + F + (L - 1)/PCR$ . 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:  $S_2 = F - R_S + (L - 1)/PCR$ .

**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  $L$ , i.e.,  $\frac{L-1}{PCR}$  is bigger than  $F$ , we define the service time  $S_3$  by  $L/PCR$ .

### 3.4 Solving the Queuing 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  $Q = 1$ . Notice that by observing the system at these epochs  $O_n$  we know that the remaining service time of the currently serviced packet equals  $\frac{L-1}{PCR}$  time units (of the service process). More-

over the distance  $D_n$  between the observation points  $O_n$  and  $O_{n+1}$  equals

$$D_n = S_{n+1} + [A_{n+1} - C_n]^+,$$

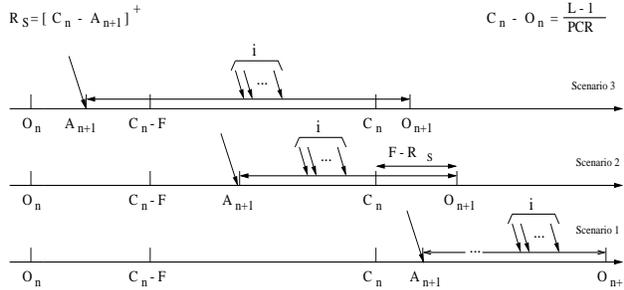
where  $S_{n+1}$  is the service time of packet  $n + 1$ , i.e., the next packet to be serviced,  $A_{n+1}$  is the arrival instance of packet  $n + 1$  and  $C_n$  the service completion time of the currently serviced packet, i.e., packet  $n$ . Finally,  $[x]^+$  denotes  $\max(x, 0)$ . Thus we can make use of the expressions for  $S_1$ ,  $S_2$  and  $S_3$  to determine the transition probabilities.

Assume that we are in state  $S = (N, P, q)$  with  $N \geq 1$ . Then scenario 3 (see Section 3.3) applies for the next packet in line, i.e.,  $D_n = S_3$ , because the next packet has already arrived and due to the assumption on the packet length the current packet has a remaining service time bigger than  $F$ . Therefore, looking at the expression for  $S_3$ , the transition probability  $P(S, S')$  to state  $S' = (N + i - 1, P', q')$ , with  $i \geq 0$ , equals

$$1[P' = (P + [\frac{L}{PCR} \frac{1}{Q}]) \bmod \frac{F}{Q}] \quad (\mathbf{A}_i^{([\frac{L}{PCR} \frac{1}{Q}]})}_{q, q'},$$

where  $[x]$  denotes  $x$  rounded to the nearest integer and  $1[\text{condition}]$  is one if *condition* is true and zero otherwise.

In state  $S = (0, P, q)$  all three scenario's are possible (see Figure 2), depending on the arrival time  $A_{n+1}$  of the next packet. We denote  $S'$  as  $(i, P', q')$ . Looking at the expressions of  $S_1$ ,  $S_2$  and  $S_3$  and keeping in mind that  $\frac{L-1}{PCR}$  is the remaining service time at time  $O_n$  we obtain the expression at the top of the next page, with  $W$  the number of frames needed to successfully transmit a request to the BS using the contention channel.



**Figure 2:** The scenario's for  $N = 0$

By ordering the states  $(N, P, q)$  lexicographically the probabilities  $P(S, S')$  define a stochastic transition probability block matrix  $\mathbf{P} = (\mathbf{Q}_{m,n})$ . The matrices  $\mathbf{Q}_{m,n}$  govern the state  $(P, q)$  transitions when

the queue length changes from  $m$  to  $n$ . Looking at the expressions for  $P(S, S')$ , it is clear that a Markov Chain of the M/G/1-type is obtained. To solve such a Markov chain the algorithm of Ramaswami is used [5] combined with [2] to find the required normalization factor. Notice that due to the periodic nature of the frames, the matrix  $G$ , required to solve this type of Markov process, may become reducible in some rare cases.

Having calculated the stationary probability vector of the process at these epochs we calculate the queue-length distribution  $X$  at the service completion times as follows:

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

where  $\bar{\pi}_i(P)$  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  $P(X = 0)$  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  $P(X = k)$  is also the probability that a packet finds  $k$  customers (packets) in front upon arrival.

## 4 Numerical Results

In this section we study the influence of the SCR, the PCR, the variation of the interarrival times 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 [6]): 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 [6]. Apart from the piggybacking probability we define the following two performance measures:

$$E = \frac{L/PCR}{P_n L/PCR + P_0(L/PCR + E[W])}$$

$$P(S, S') =$$

$$1[P' = (P + \lfloor \frac{L}{PCR} \frac{1}{Q} \rfloor) \bmod \frac{F}{Q}] \sum_{s=1}^{\lfloor \frac{L-1}{PCR} - F \rfloor \frac{1}{Q}} \left( \mathbf{D}_0^{s-1} \mathbf{D}_1 \mathbf{A}_i^{\lfloor \frac{L}{PCR} \frac{1}{Q} \rfloor - s} \right)_{q, q'} +$$

$$\sum_{s=\lfloor \frac{L-1}{PCR} - F \rfloor \frac{1}{Q} + 1}^{\lfloor \frac{L-1}{PCR} \frac{1}{Q} \rfloor} \left( \mathbf{D}_0^{s-1} \mathbf{D}_1 \mathbf{A}_i^{(F/Q)} \right)_{q, q'} 1[P' = P + F/Q + s \bmod \frac{F}{Q}] +$$

$$1[P' = 0] \sum_{s > \lfloor \frac{L-1}{PCR} \frac{1}{Q} \rfloor} \left( \mathbf{D}_0^{s-1} \mathbf{D}_1 \sum_x P[W = x] \mathbf{A}_i^{\lfloor \frac{F+xF}{Q} \rfloor + F/Q - (P+s) \bmod F/Q} \right)_{q, q'}$$

$$D = P_0 E[W] + P_n E[X - 1 | X > 0] L / PCR,$$

with  $P_0 = P(X = 0)$  and  $P_n = P(X > 0)$ . 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  $Q = 8$ .

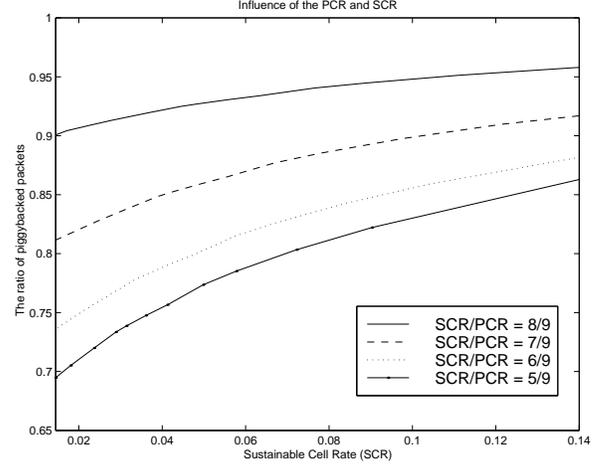
#### 4.1 The Influence of the SCR and the PCR

The packet length  $L$  in figure 3 is set at 20, the rate vector  $\beta$  and the transition matrix  $\mathbf{D}$  are the following:

$$\beta = \begin{pmatrix} y & xy \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 1 - \frac{2xy}{5} & \frac{2xy}{5} \\ \frac{xy}{5} & 1 - \frac{xy}{5} \end{pmatrix} \quad (3)$$

with  $1/10 \leq y \leq 1/200$  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. When we change  $x$  with  $y$  fixed we get a fixed PCR but a variable SCR.

Figures 3 and 4 show that for a fixed SCR more piggybacking and a better efficiency  $E$  is realized as the ratio  $\frac{SCR}{PCR}$  grows. Notice that this ratio is an indication for 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



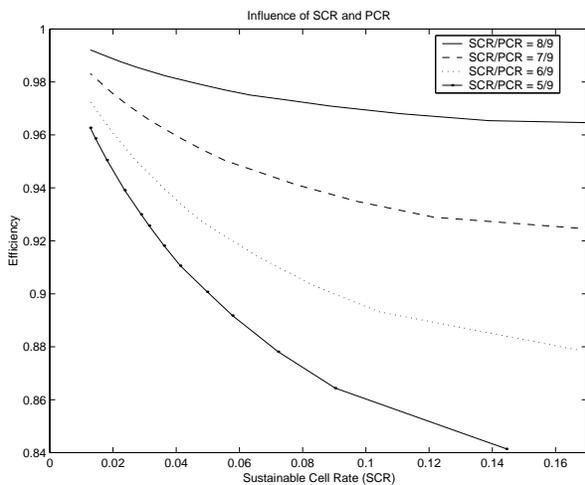
**Figure 3:** The impact of the SCR and the PCR on  $P(X = 0)$

(an effect that increases with lower  $\frac{SCR}{PCR}$  ratios), their efficiency  $E$  is smaller.

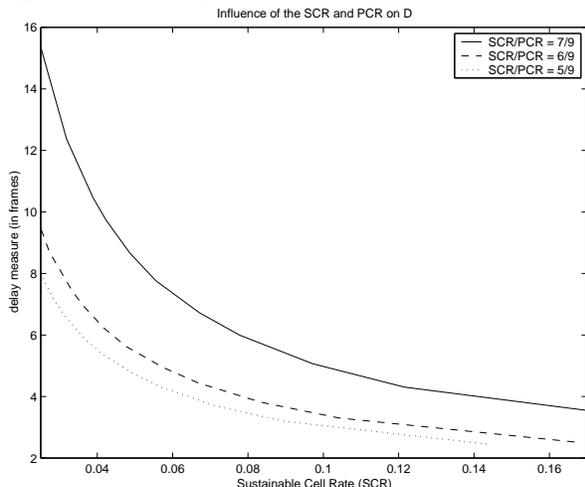
Figure 5 shows that better delays are achieved as the ratio  $\frac{SCR}{PCR}$  decreases, a rather logical result as this ratio is a measure for the load of the scheme. Secondly the delay decreases as the SCR increases, because the workload is offered more gradually by higher bitrate sources (because due to the traffic scheduler in the BS the amount of work is offered to the server in multiples of  $L/PCR$ ).

#### 4.2 The Influence of the Variation of the Interarrival Times

In this section we keep the PCR and the SCR fixed and study the influence of any additional variation of the packet interarrival times that is not determined by the SCR and the PCR. Again the packet length  $L$  is fixed at 20, the rate vector  $\beta = (y, y/2 + x, y/2 - x)$



**Figure 4:** *The impact of the SCR and the PCR on  $E$*



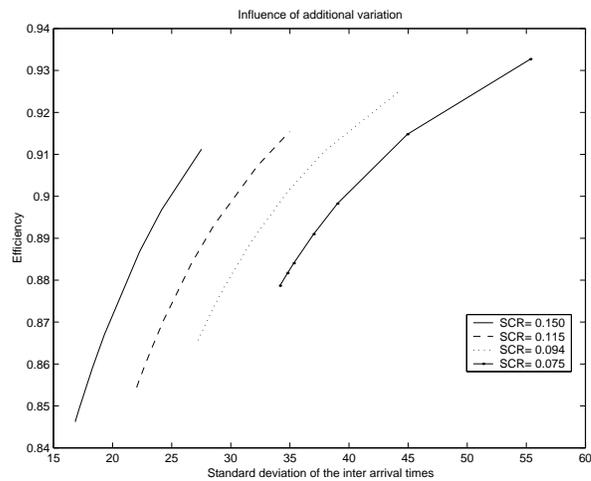
**Figure 5:** *The impact of the SCR and the PCR on  $D$*

and the transition matrix  $\mathbf{D}$  is the following:

$$\mathbf{D} = \begin{pmatrix} 1 - \frac{y}{50} & \frac{y}{100} & \frac{y}{100} \\ \frac{y}{200} & 1 - \frac{y}{100} & \frac{y}{200} \\ \frac{y}{200} & \frac{y}{200} & 1 - \frac{y}{100} \end{pmatrix} \quad (4)$$

with  $\frac{1}{y} = 10, 13, 16, 20$  and  $x$  between 0 and  $\frac{y}{2}$ . We can change the variation while keeping the SCR and the PCR fixed by changing  $x$  with  $y$  fixed. On the other hand if  $x$  is fixed and  $y$  is increased the SCR becomes bigger but the ratio  $\frac{SCR}{PCR}$  remains the same, i.e., 0.6.

Figure 6 shows that the additional variation that is not caused by the choice of the SCR or the PCR has an important influence on the results. The bigger the additional variation becomes the better the efficiency  $E$ . Higher the additional variation also results in a higher piggybacking percentage (this figure is not



**Figure 6:** *The impact of the standard deviation of the interarrival times on  $E$*

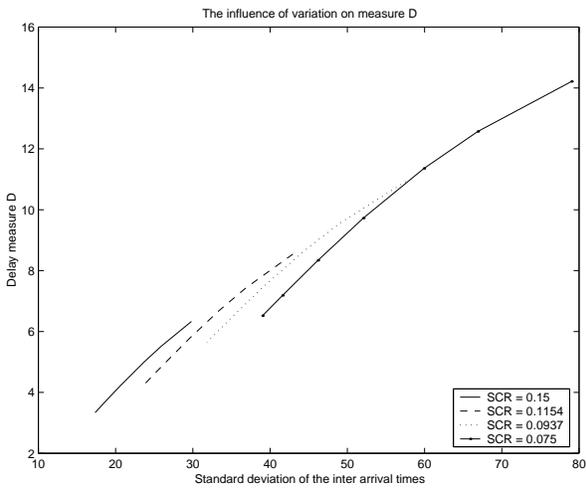
included). Therefore, the SCR and the PCR do not give a sufficient indication as to the piggybacking capabilities or the efficiency of an MS on a packet level. Figure 7 shows that the delay increases with increasing variation, this because more variation means more bursty input traffic and thus longer queues.

A combination of the following observations might explain these results. Typically having a higher variation with a fixed average and peak rate means that more traffic is sent at a high rate ( $\geq SCR$ ). Secondly, having large or exceptionally large interarrival periods makes little difference as both are very unlikely to result in piggybacking. On the other hand, to compensate for a single exceptionally large interarrival period, as opposed to a large period, we need much more short interarrival periods (e.g. the SCR is fixed).

### 4.3 The Influence of Correlation

In this section we study the influence of the correlation between the length of consecutive interarrival periods while keeping the interarrival distribution  $I$  fixed. In Section 3.2 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 4.1 with  $x$  fixed at  $1/2.7272$  thus  $\frac{SCR}{PCR} = \frac{5}{9}$ . We consider five different values for  $y$  resulting in as much different  $SCR$ s. In Figure 8 the parameter  $r$  is varied from 1 to infinity, notice that the correlation does not go to one if  $r$  approaches infinity because of the geometric nature of the arrival scheme.

Figure 8 shows that this type of correlation is less important when studying the piggybacking capabilities or the efficiency of an MS (the figure that shows



**Figure 7:** The impact of the standard deviation of the interarrival times on  $D$

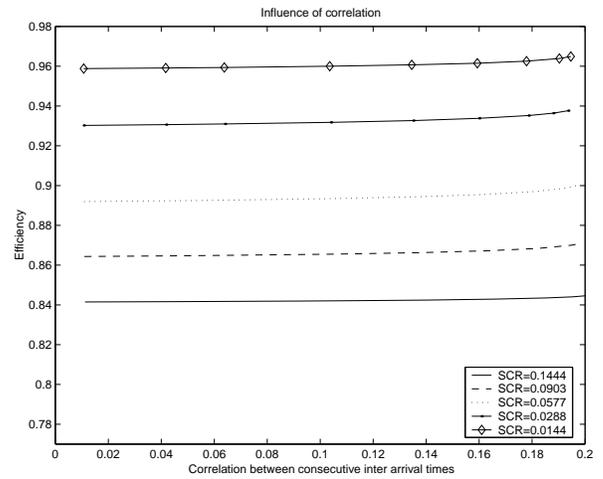
the impact of the correlation on  $P(X = 0)$  is not shown). However, in Figure 9 it is shown that the correlation does have an important impact on the delay. Indeed, more correlation leads to longer delays especially for low bit rate traffic.

## 5 Conclusions

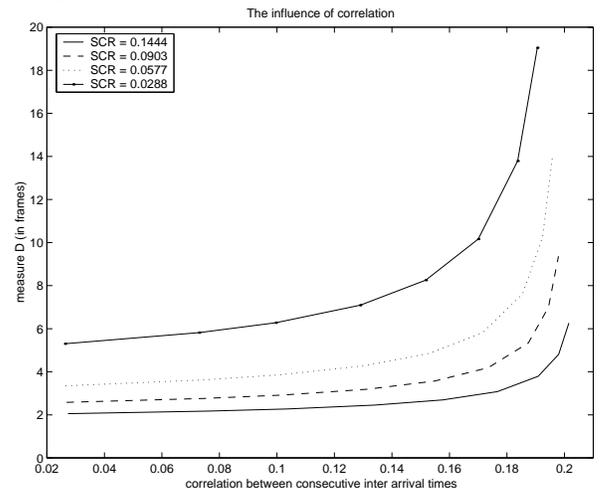
In this paper we have proposed a MAC protocol for a wireless ATM network which uses a variant of the ISA protocol to inform the BS about the bandwidth needs of the MSs. We evaluate the influence of the traffic characteristics of the packet arrival process in the MSs on the efficiency of the protocol and on the delay packets experience in the MS to access the shared medium.

## References

- [1] O. Casals, J. Garcia, and C. 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*.
- [2] M.F. Neuts. Structured stochastic matrices of M/G/1 type and their applications. *Marcel Dekker Inc, New York, 1989*.
- [3] D. Petras. Medium access control protocol for wireless, transparent ATM access in MBS. *RACE Mobile Telecommunications Summit, (Cascais, Portugal), 1995*.
- [4] D. Petras and A. Kramling. Fast collision resolu-



**Figure 8:** The impact of the correlation on the Efficiency  $E$



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

tion in wireless ATM networks. *2nd MATHCOM, Vienna, Austria, Feb, 1997*.

- [5] V. Ramaswami. A stable recursion for the steady state vector in markov chains of M/G/1 type. *Comm. Statist. - Stochastic Models, Vol 4, No 1, p183-188, 1988*.
- [6] 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*.
- [7] B. Van Houdt, C. Blondia, O. Casals, J. Garcia, and D. Vazquez. A MAC protocol for wireless ATM systems, supporting the service categories. *Proc. of the 16-th ITC, Edinburgh, UK, 1999*.