# Performance Evaluation of the Identifier Splitting Algorithm with Polling in Wireless ATM Networks[1]

B. Van Houdt, C. Blondia

*University of Antwerp*
*Department of Mathematics and Computer Science*
*Performance Analysis of Telecommunication Systems Research Group*
*Universiteitsplein, 1, B-2610 Antwerp - Belgium*
{vanhoudt,blondia}@uia.ua.ac.be
http://win-www.uia.ac.be/u/pats/

**Abstract**

This paper presents a performance analysis of the Identifier Splitting Algorithm combined with polling, a contention resolution scheme used to inform the Base Station about the bandwidth needs of the Mobile Station in a wireless ATM network. An analytical model leads to the evaluation of performance parameters which determine the throughput and the access delay of the algorithm for different system parameters. This analysis is used to investigate the influence of the system parameters on the performance, from which guidelines for parameter tuning can be derived.

## 1 Introduction

The development of powerful high performance portable computers and other mobile devices such as palmtops, have motivated the increasing interest in wireless communication systems, in particular for LANs (e.g. in an office environment). This evolution has to be combined with the trend towards high capacity, service integration and Quality of Service (QoS) provisioning, currently supported in fixed networks by the Asynchronous Transfer Mode (ATM). A seamless connection between these wireless LANs and the fixed network requires the definition of an ATM based transport architecture for an integrated services wireless network.

In a wireless network, the broadcast nature of the radio channel requires the introduction of a Medium Access Control (MAC) layer, in order to coordinate the access to the shared

[1]This work was supported in part by Vlaams Actieprogramma Informatietechnologie under project IWT/980272 "Multiservice Network Technologies".

radio channel. A MAC protocol should not only avoid collisions and distribute the available bandwidth in an efficient way, but it is also a key component in the support of QoS provisioning.

In this paper we study the performance of a contention resolution scheme used to inform a Base Station about the bandwidth requirements of the Mobile Stations. This scheme is a tree algorithm know as the Identifier Splitting Algorithm [1, 2]. This algorithm combined with a polling scheme forms an important component of a MAC protocol for a wireless ATM network defined in [3]. We evaluate parameters which determine the throughput and access delay of the protocol for different system parameters.

The paper is organized as follows. A short description of the wireless ATM MAC protocol is given in Section 2. The Identifier Splitting Algorithm is treated in more detail in Section 3 and its performance is evaluated in Section 4. This analysis is used in Section 5 to investigate the influence of the system parameters on the performance, in particular on the throughput and the access delay. The conclusions and future work are summarized in Section 6.

## 2 General Protocol Description

In this paper, we consider the following class of WATM systems. A wide variety of Medium Access Control (MAC) protocols for WATM belongs to this class of MAC protocols, examples are DSA++ [4, 5], D$^2$MA [6], EC-MAC [7] and [3]. Consider a cellular network with a centralized architecture, i.e., the area covered by the wireless ATM network is subdivided into a set of geographically distinct cells each with a diameter of approximately 100m (slight overlaps are allowed to facilitate the handovers from one cell to the next). Each cell contains a base station (BS) serving a finite set of mobile stations (MS). This BS is connected to an ATM switch, which supports mobility, realizing seamless access to the wired network (see Figure 1).

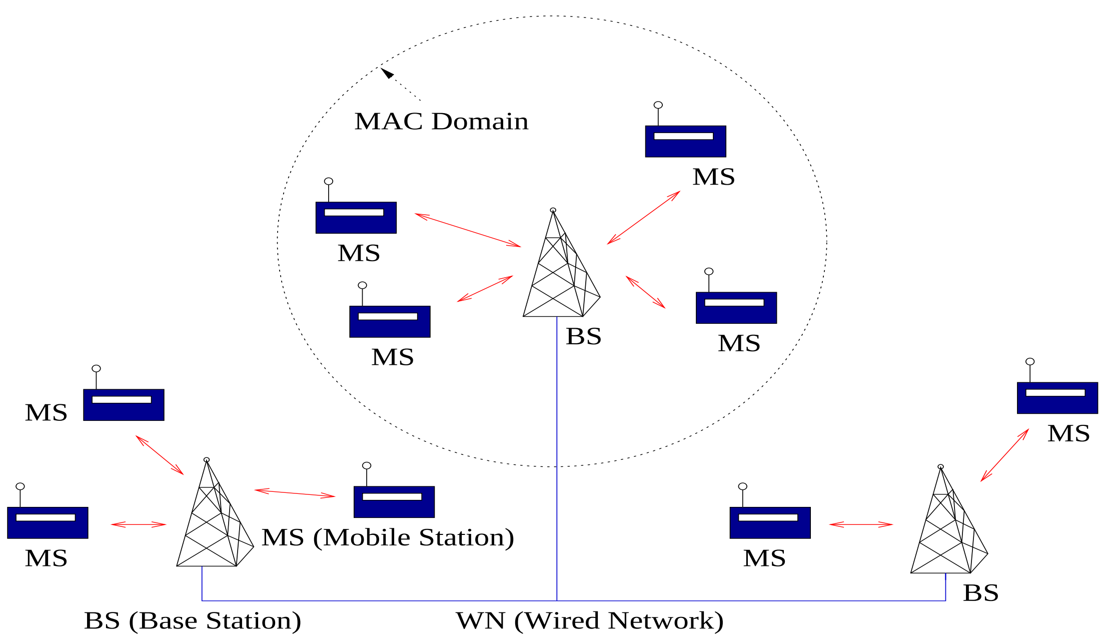


**Figure 1:** *Reference configuration of the system*

Two logically distinct communication channels (uplink and downlink) are used to support the information exchange between the BS and the MSs. ATM cells arriving at the BS are broadcasted downlink, while upstream ATM cells must share the radio medium using a MAC protocol. The BS controls the access to the shared radio channel (uplink). The access technique is Time Division Multiple Access (TDMA) combined with Frequency Division Duplex (FDD) to separate the uplink and downlink channels.

Traffic on both the uplink and downlink channel is grouped into fixed length frames (of approximately 1-2 ms length) to reduce the battery consumption. The uplink and downlink frames are synchronized in time, i.e., the header of a downlink frame is immediately followed by the start of an uplink frame (after a negligible round trip time that is captured within the guard times, see Figure 2). Each uplink frame consists of a (variable length) contentionless

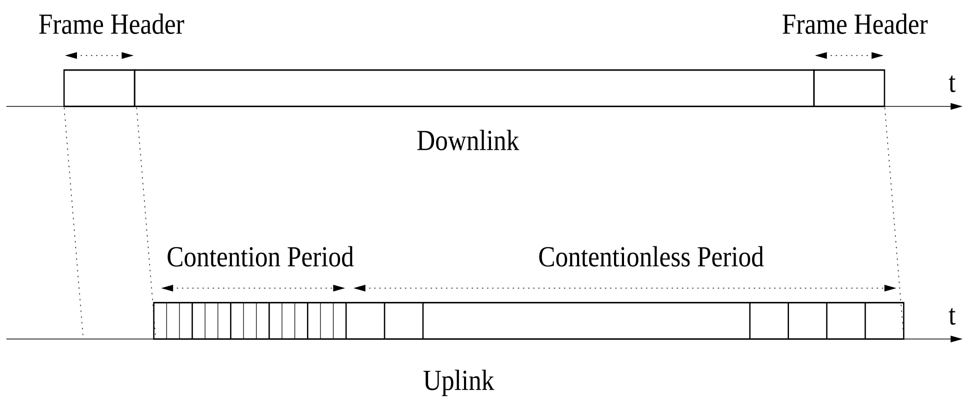


**Figure 2:** *Frame Structure*

and a (variable length) contention period, where the length of the contentionless period dominates that of the contention period. An MS is allowed to transmit in the contentionless period after receiving a permit from the BS. To obtain these permits the MSs must inform the BS about their bandwidth needs using requests. Whenever an MS forwards an ATM cell to the BS a request is piggybacked to the ATM cell. When an ATM cell that is generated in an MS finds the transmission queue empty (in that MS), it uses the contention period to inform the BS about its presence (i.e., it uses the contention period to sent a request), as piggybacking is no longer an option.

The contentionless period in an uplink frame contains a number of fixed length slots. These slots are large enough to carry a single ATM cell, a request and the physical overhead needed to guarantee a safe guard time, some training sequences and error detection codes. The slots forming the contention period have the same size but they can be subdivided into $m$ minislots (as requests tend to be a lot smaller than ATM cells). Realistic values for $m$ are 1, 2, 3 and possibly 4.

Each downlink frame starts with a frame header in which the required feedback on the contention period of the previous uplink frame is given. This informs the MSs participating in the contention period whether there was a collision or the request has been successfully received. The frame header also contains the permits for the contentionless period in the next uplink frame. Finally a unique $n$-bit MAC address is assigned by the BS to every MS.

Thus the main idea is that every MS oscillates between a state in which the bandwidth requirements are piggybacked with upstream ATM cells and a state in which the MS needs to access the contention channel to inform the BS about its needs (especially those MSs that hold bursty VBR connections). Therefore, the delay experienced on the contention channel has a major impact on the delay performance of the MAC protocol. The contention resolution scheme used in this paper is the Identfier Splitting Algorithm (ISA) combined with polling [1, 2, 3, 8]. The aim of the paper is to evaluate the performance for different system parameters. The main difference with some prior evaluations [2, 8] is indicated in the next few sections.

In the remaining part of the paper a minislot is simply called a contention slot or slot, except when stated otherwise.

## 3 The Identifier Splitting Algorithm

The Identifier Splitting Algorithm is based on the well known tree algorithm [9] and was proposed by Petras [1, 2]. A contention cycle (CC) consists of a number of consecutive upstream frames during which the contention is solved for all requests present in the MSs at the beginning of the cycle. The system is gated, in the sense that any request generated by an MS during a CC that wants to use the contention channel, is blocked until the start of the next CC.
In the first frame of a cycle a single contention slot is available. We refer to this slot as level 0 of the tree. Any MS having a request ready at the start of this CC, makes use of this slot. Next the BS checks whether a successful transmission occurred in this slot and informs the MS(s) that were involved in the scheme accordingly in the next downstream frame using a feedback field. Two situations are possible:

(i) An MS sending its request in this slot was successful. In this case the MS returns to the piggybacked state.

(ii) The transmission was not successful, i.e., a collision occurred. In this case, the next level (level one) of the CC provides 2 contention slots. Based on the first bit of their MAC addresses, as opposed to the classical coin flip, the MSs that are involved split up into two distinct sets. An MS belonging to the first set uses the first slot to attempt a retransmission, while the second slot is used by the MSs belonging to the second set.

This process of generating two slots in the next level for each slot in which a collision occurred, is repeated level after level, each time using the next bit of the MAC address in case of a collision. Thus during level $i$ of a CC, two MSs can only collide if their MAC addresses have the same $i$ first bits. Therefore, provided that the address that uniquely identify an MS, is $n$ bits long, all collisions are always resolved in $n+1$ levels. Also notice that for every level the number of contention slots equals twice the number of collisions of the previous level. To clarify all this, Figure 2 shows an example of a CC with 6 participants. In this figure **CO** refers to a collision, **SU** to a success and **EM** to an empty slot. The MAC addresses of the successful MSs are added to the corresponding slot.

Normally every level of the tree corresponds to a single frame, except when the number of slots at level $i$ is bigger than some predefined value $L$. This parameter $L$ defines the maximum number of contention slots that we allow in a single frame. Thus, if a certain

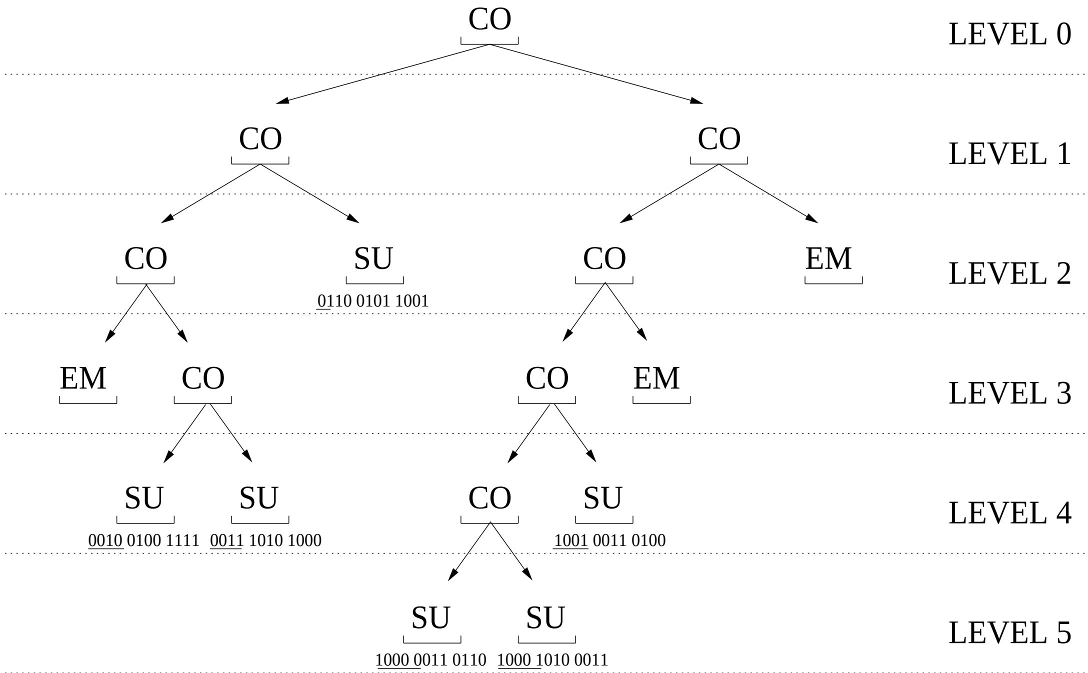


**Figure 3:** *Demonstrating ISA*

level of the tree requires $x = mL + j$, with $1 \leq j \leq L$, slots then $m + 1$ frames are required for this level.

It is the presence of this parameter $L$ that differentiates this analysis from some prior evaluations [2, 8], where the influence of $L$ was neglected (assuming that $i$ frames are required to represent $i$ levels). Apart from the influence of this parameter $L$ on the delay, we are particularly interested in its interaction with the polling threshold $N_p$ and the starting level $S_l$ (both these parameters are defined in the next two subsections).

### 3.1 The Identifier Splitting Algorithm Combined with Polling

One of the attractive features of the Identifier Splitting Algorithm, apart from its dynamic nature, 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 $i$ (see Figure 3) contains $k$ collisions and the MAC-addresses are $n$ bits long then the BS concludes 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. This information can be used by the BS in an attempt to improve the performance characteristics. The basic idea here is that when the size of the remaining MAC address space $Y$ becomes smaller than some predefined value, say $N_p$, the protocol switches to polling. Polling, in this context, means that one slot is provided for each address in the remaining address space. Depending on the relationship between $L$ and $Y \leq N_p$, one or multiple frames are required to support polling. In this analysis we mainly focus on the interaction between the two protocol parameters $L$ and $N_p$. This is again an important distinction with a prior analysis ([8]), where it was assumed that the remaining address space was always polled in a single frame (this because the $L$ value was not taken into account).

### 3.2 Skipping the First Few Levels

In the previous sections the contention period of the first frame of a CC consisted of a single contention slot (level zero of the tree). Now we drop this condition: instead of starting with just one contention slot in the first frame, we provide more than one slot during the first frame of a CC.
The starting level is said to be $S_l$, with $0 \leq S_l \leq n$, if the first frame of the CC contains $2^{S_l}$ contention slots (if $2^{S_l} \geq L$ a number of frames are required to support the starting level $S_l$). An MS taking part in the contention cycle selects one of these $2^{S_l}$ slots based on the first $S_l$ bits of its $n$-bit MAC address. Again, we are mainly interested in the interaction between the protocol parameters $L$ and $S_l$.

## 4 Performance Analysis

As a reminder, let us summarize the following important protocol parameters:

- $n$ : the length of the MAC addresses (in bits).
- $L$ : the maximum number of contention slots allowed in one frame.
- $N_p$ : the value that triggers the polling mechanism.
- $S_l$ : the starting level.

In this section we calculate the following expected values:

- $E[F \mid X = k]$: the expected length of a CC (expressed in frames) with $k \geq 2$ participants,
- $E[S \mid X = k]$: the expected number of contention slots that a CC with $k \geq 2$ participants requires.

The value $E[F]$ is strongly related with the delay experienced by the protocol, whereas $E[S]$ is related with the throughput of the protocol. Also, notice that $E[S \mid X = k]$ does not depend upon the value of $L$. Moreover, in the special case of $L = 1$ both expected values ($E[F]$ and $E[S]$) are identical. Therefore, it is sufficient to set up a scheme to calculate $E[F \mid X = k]$ for any value of $L$.

Consider the following collection of sets: $A_j^{(i)}$, $1 \leq j \leq 2^i$, $0 \leq i \leq n$, is the set of all MAC addresses for which the value of the $i$ first bits equals $j-1$. Each of these sets is referred to as a *virtual* slot at level $i$ (all together there are $2^{n+1}-1$ *virtual* slots). Assuming that a CC has $k$ participants, we state that a collision occurs in the *virtual* slot $A_j^{(i)}$ if 2 or more of the $k$ MAC addresses of the participating MSs are a part of the set $A_j^{(i)}$, otherwise we state that the *virtual* slot is collision free. Notice that every slot of a CC corresponds with exactly one *virtual* slot; e.g., in Figure 3 the second slot at level 3 corresponds to *virtual* slot $A_2^3$, while the third slot (at level 3) corresponds to *virtual* slot $A_5^3$. If a *virtual* slot contains a collision then there exists a corresponding slot in the CC and this slot must hold a collision.

Next, the analysis is divided into three parts: in the first part $N_p$ and $S_l$ are both equal to zero, in the second part $N_p \geq 0$ but $S_l$ is still zero, finally, in the last case $N_p \geq 0$ and $S_l \geq 0$.

### 4.1 Part 1: $N_p = S_l = 0$

Define $p(i, l_1, l_2, k)$ to be the probability that at level $i$, a specific collection of $l_1$ *virtual* slots is collision free and at level $i+1$, there are exactly $l_2$ collision free *virtual* slots, given that we had $k$ contenders in the CC. Notice that the number of collisions at level $i$ might be smaller than $2^i - l_1$, thus other *virtual* slots that do not belong to the specific collection of size $l_1$ might also be collision free. Define $C_r^n$ as the number of different possible combinations of $r$ from $n$ different items. A reasoning based on the Inclusion-Exclusion Principle allows us to state the following:

$$\begin{aligned} p(i, l_1, l_2, k) &= \frac{1}{C_k^{2^n}} \sum_{j=0}^{l_1} 2^{(n-i)j} C_j^{l_1} C_s^{2^{i+1}-2l_1} \sum_{j'=0}^{s} 2^{(n-i-1)j'} C_{j'}^s C_{k-j-j'}^{2^n - l_1 2^{n-i} - s 2^{n-i-1}} \\ &\quad - \sum_{x=1}^{2^{i+1}-2l_1-s} C_s^{s+x} p(i, l_1, l_2 + x, k), \end{aligned} \tag{1}$$

with $s = l_2 - 2l_1$ and with $p(i, l_1, l_2, k) = 0$ for $l_2 < 2l_1$. Also, for $l_2 = 2^{i+1}$ this reduces to

$$p(i, l_1, 2^{i+1}, k) = \frac{1}{C_k^{2^n}} \sum_{j=0}^{l_1} 2^{(n-i)j} C_j^{l_1} 2^{(n-i-1)(k-j)} C_{k-j}^{2^{i+1}-2l_1}. \tag{2}$$

Next, we define $s(i, l_1, l_2, k)$ as the probability of having exactly $l_1$ collision free *virtual* slots at level $i$ and exactly $l_2$ collisions free *virtual* slots at level $i+1$, given that we had $k$ contenders in the CC. We have the following relationship between $p(i, l_1, l_2, k)$ and $s(i, l_1, l_2, k)$:

$$s(i, 2^i, l_2, k) = p(i, 2^i, l_2, k), \tag{3}$$

$$s(i, l_1, l_2, k) = C_{l_1}^{2^i} p(i, l_1, l_2, k) - \sum_{x=1}^{2^i - l_1} C_{l_1}^{l_1+x} s(i, l_1 + x, l_2, k). \tag{4}$$

As a consequence of the previous we also have $s(i, l_1, l_2, k) = 0$ for $l_2 < 2l_1$. In the next subsection we will make use of these probabilities to obtain $E[F \mid X = k]$ for $N_p \geq 0$.

### 4.2 Part 2: $0 \leq N_p < 2^n$ and $S_l = 0$

Define the random variable $F_i$ as the number of frames required to support level $i$ of the tree, then

$$E[F \mid X = k] = \sum_{i=0}^{n} E[F_i \mid X = k]. \tag{5}$$

With $k \geq 2$ and $N_p < 2^n$, we have $F_0 = 1$ and $F_1 = 1$, resp. 2, if $L \geq 2$, resp. $L = 1$. Therefore, we can focus on $E[F_i \mid X = k]$ with $i \geq 2$.

We separate the following three events $E_1^{(i)}, E_2^{(i)}$ and $E_3^{(i)}$:

- $E_1^{(i)}$: the CC is resolved within the first $i-2$ levels (with or without polling) or polling takes place at level $i-1$.
- $E_2^{(i)}$: the CC is resolved (without polling) at level $i-1$ or polling takes place at level $i$.

- $E_3^{(i)}$: the CC is not resolved within the first $i-1$ levels and polling does not occur at level $i$.

Thus, $E[F_i] = P(E_1^{(i)})E[F_i \mid E_1^{(i)}] + P(E_2^{(i)})E[F_i \mid E_2^{(i)}] + P(E_3^{(i)})E[F_i \mid E_3^{(i)}]$. Given that the first event $E_1^{(i)}$ occurs, the expected number of frames $F_i$ at level $i$ equals zero. The two other expressions are found as follows.

Define $C_i$ as the number of collisions at level $i$. Suppose that $C_i = N_c$, then the size of the remaining address space is $N_c 2^{n-i}$. Thus, at level $i+1$ we have no polling when $N_c > N_p/2^{n-i}$. Also, having $N_c > N_p/2^{n-i}$ is equivalent to having $N_c > \lfloor N_p/2^{n-i} \rfloor$ for $N_c$ an integer value. Hence, polling does not occur at level $i+1$ if $C_i \geq 1 + \lfloor N_p/2^{n-i} \rfloor$. We denote $1 + \lfloor N_p/2^{n-i} \rfloor$ as $c_i$.

Hence, we can rewrite the previously mentioned events as: $E_1^{(i)} = C_{i-2} < c_{i-2}$, $E_2^{(i)} = C_{i-2} \geq c_{i-2} \cap C_{i-1} < c_{i-1}$ and $E_3^{(i)} = C_{i-2} \geq c_{i-1} \cap C_{i-1} \geq c_{i-1}$. We already mentioned that $E[F_i \mid X = k \cap E_1^{(i)}]$ is zero. Also,

$$P(E_2^{(i)} \mid X = k)E[F_i \mid X = k \cap E_2^{(i)}] = \sum_{l_1 \geq c_{i-2}} \sum_{l_2 < c_{i-1}} s(i-2, 2^{i-2} - l_1, 2^{i-1} - l_2, k) \left\lceil \frac{2^{n-i+1} l_2}{L} \right\rceil,$$

and finally

$$P(E_3^{(i)} \mid X = k)E[F_i \mid X = k \cap E_3^{(i)}] = \sum_{l_1 \geq c_{i-2}} \sum_{l_2 \geq c_{i-1}} s(i-2, 2^{i-2} - l_1, 2^{i-1} - l_2, k) \left\lceil \frac{2 l_2}{L} \right\rceil,$$

where $s(i, l_1, l_2, k)$ was found in the previous subsection.

### 4.3 Part 3: $0 \leq N_p < 2^n$ and $S_l \geq 0$

To avoid any confusion with the previous we define $F_i(S_l)$ as the number of frames required to support level $i$ knowing that the starting level is $S_l$. Clearly, for $x < S_l$ and $y > S_l + 1$

$$\begin{aligned} E[F_x(S_l) \mid X = k] &= 0, & (6)\\ E[F_{S_l}(S_l) \mid X = k] &= \left\lceil \frac{2^{S_l}}{L} \right\rceil, & (7)\\ E[F_y(S_l) \mid X = k] &= E[F_y(0) \mid X = k], & (8) \end{aligned}$$

where $E[F_i(0) \mid X = k]$ was found in part 2. Thus, only the expected number of frames to support level $S_l + 1$ remains to be determined. Again, we separate three events:

- $E_1(S_l)$: the CC is solved at level $S_l$,
- $E_2(S_l)$: polling occurs at level $S_l + 1$,
- $E_3(S_l)$: the CC is not solved at level $S_l$ nor does polling occur at level $S_l + 1$.

Making use of the values $c_i$ defined in part 2, we can rewrite these events as $C_{S_l} = 0$, $C_{S_l} > 0 \cap C_{S_l} < c_{S_l}$ and $C_{S_l} \geq c_{S_l}$. Hence,

$$E[F_{S_l+1}(S_l) \mid X = k] =$$
$$\sum_{l_1} \left( \sum_{l_2 < c_{S_l}} s(S_l - 1, l_1, 2^{S_l} - l_2, k) \left\lceil \frac{2^{n-S_l} l_2}{L} \right\rceil + \sum_{l_2 \geq c_{S_l}} s(S_l - 1, l_1, 2^{S_l} - l_2, k) \left\lceil \frac{2 l_2}{L} \right\rceil \right),$$

for $S_l > 0$. The results for $S_l = 0$ are obtained using part 2 of the analysis.

# 5 Impact of the System Parameters on the Performance

All the results presented in this section were obtained using the package Mathematica and are therefore exact. Due to the time required to perform rational calculations we considered MAC addresses with $n = 7$ bits, although $n = 8 - 10$ bits might be somewhat more realistic. The number of participants (MSs) in the CC therefore varies from $k = 2$ to 128 (sometimes we only show the results for $k \leq 60$ because no significant differences were observed for $k \geq 60$). For the number of contention slots allowed in one frame we considered $L = 4s$ with $4 \leq s \leq 16$ or $L = 128$. The trigger value $N_p$ is also a multiple of 4 between 16 and 64.

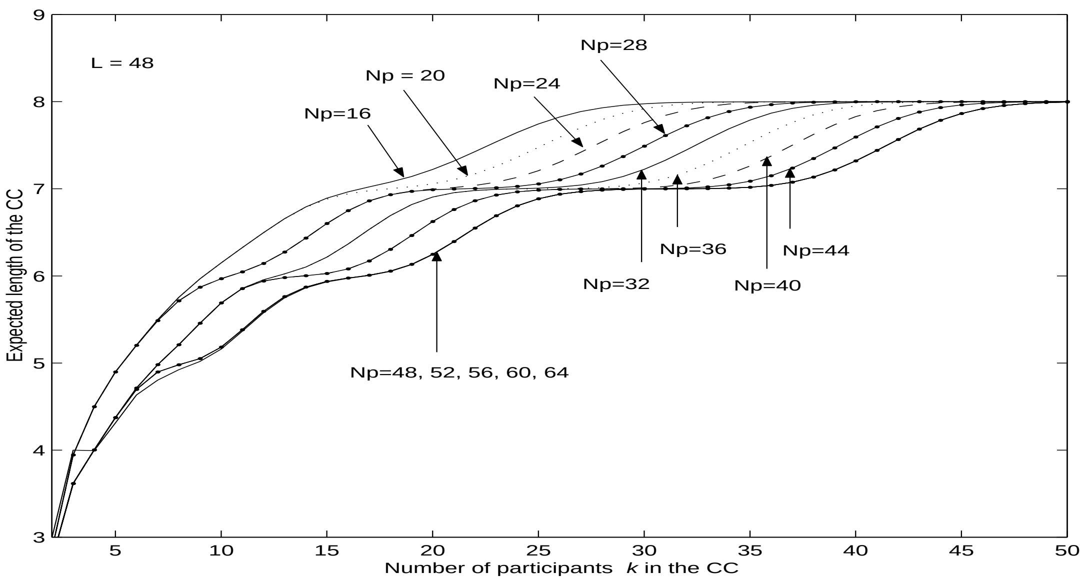


**Figure 4:** *The interaction between $L$ and $N_p$ with $L = 48$*

## 5.1 Tuning the Trigger Value $N_p$

Figure 4 ($L = 48$) shows that the expected length of the CC (in frames) decreases as $N_p$ increases for $N_p \leq 48$. Indeed as long as $N_p \leq L$ polling only lasts one frame and therefore it always results in a delay improvement. More surprisingly, all the curves are almost identical when $N_p = 48, 52, 56$ and 60. To understand this let us compare the cases $N_p = 48$ and $N_p = 52$. Both these cases behave identical except when, at some level $i < 6$,

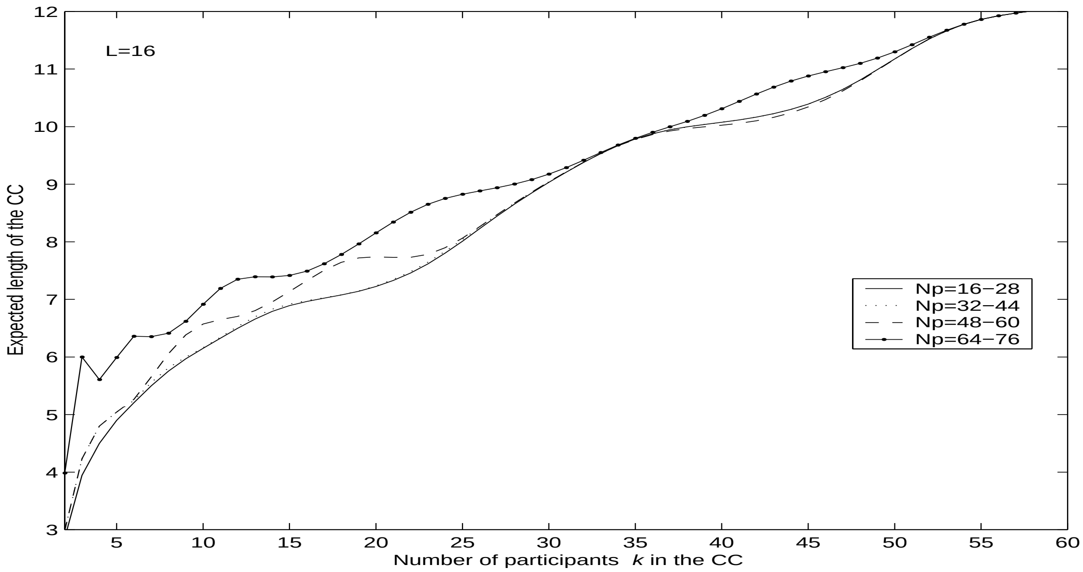


**Figure 5:** *The interaction between $L$ and $N_p$ with $L = 16$*

the size of the remaining address space $Y$ is larger than 48 but smaller than (or equal to) 52. In such a case we switch to polling if $N_p = 52$, i.e., $Y$ contention slots are included in the next two frames. Thus, the remaining length of the CC is two frames. When $N_p = 48$, it is very likely that the remaining length of the CC is also two frames. Indeed, the first frame to come contains level $i+1$ of the tree (only one frame is required to support level $i+1$ as $L = 48$ and $i < 6$) and the second frame to come is most likely used to poll the remaining contenders after level $i+1$ (as it is highly probable that the size of the remaining address space will drop below 48). Finally, increasing $N_p$ even more ($N_p = 64$) results in a somewhat larger delay for small values of $k$. Therefore, choosing $N_p > L$ might not be that useful.

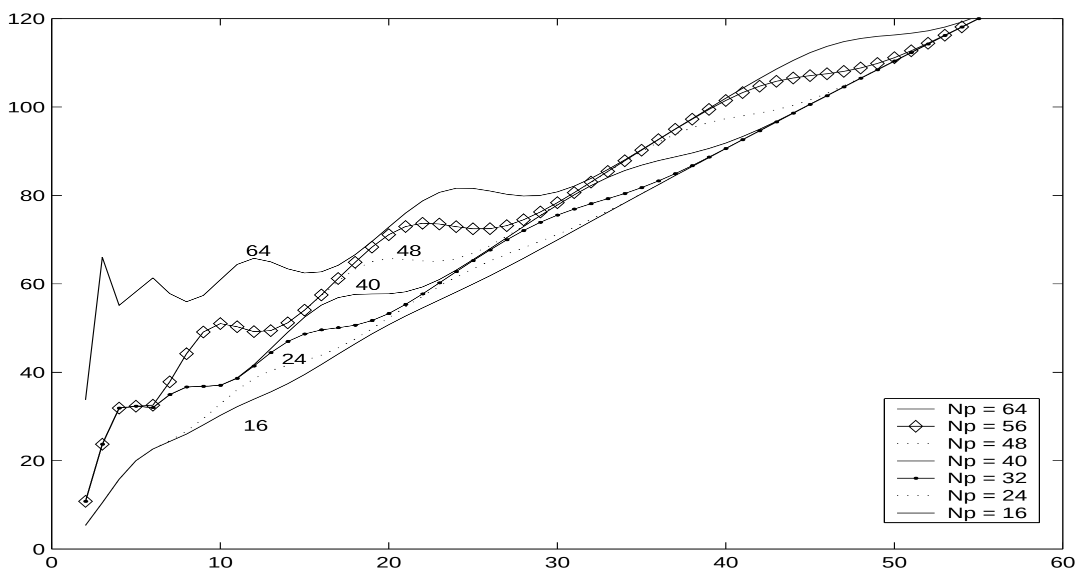


**Figure 6:** *$E[S]$ for different values of $N_p$*

Figure 5 shows the results for $L = 16$. It confirms that there is no use in choosing a polling threshold $N_p > L$ when we look at the expected delay. Moreover, the difference between two values of $N_p$ is only significant if there is a multiple of $L$ in between.

In general, with respect to the expected delay of the scheme, we conclude that the optimal choice for $N_p$ is $L$. There is one exception to this rule: setting $N_p = 2^n$ with $n$ small, e.g., $n < 8$, might result in a better delay: especially if $k$ becomes large, i.e., if the contention channel is highly loaded. For example, in a system with $N_p = 128$ and $L = 16$ (as shown in Figure 5) the length of the CC would be fixed and equal to 8 frames. The main disadvantage of choosing $N_p = 2^n$ is the low throughput that is obtained, leaving less slots available for contention free transmissions.

The throughput of a CC with $k$ participants can be defined as $k/E[S \mid X = k]$. Figure 6 shows that the expected number of slots in a CC (given that we have $k$ contenders) always increases when $N_p$ increases. Moreover, as $N_p$ approaches zero, $E[S \mid X = k]$ behaves more and more linearly.

Combining Figures 4, 5 and 6 we can conclude that $N_p$ should always be chosen smaller than or equal to $L$. The closer we choose $N_p$ to $L$ the better the mean delay but the worse the throughput becomes.

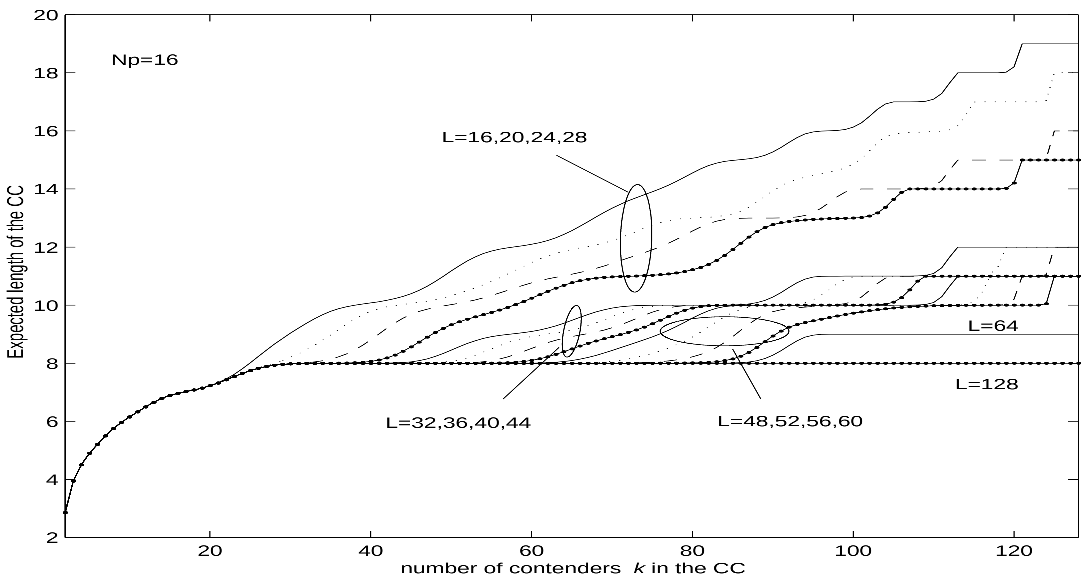


**Figure 7:** *$E[F]$ for different values of $L$*

## 5.2 The influence of the parameter $L$

In this section we investigate the influence of the maximum number $L$ of contention slots allowed in one frame. Figure 7 shows $E[F]$ for different values of $L \geq N_p = 16$. A number of conclusions can be drawn from this figure. Clearly, the less contention slots we allow in one frame the larger the delay becomes. Moreover, the delay improvements that we get when we increase $L$ are the most significant if there is a power of 2 in between. In the previous section we saw that different choices for $N_p$ only resulted in a significant difference if there is a multiple of $L$ in between. Because $N_p = 16$, a small power of two, it is tempting to believe that the difference between two choices of $L$ are the most significant if there is a multiple of $N_p$ in between. Numerical experiments have shown that this is generally not the case. Moreover, even if there is no power of two in between different choices of $L$, we still get a relevant impact on the mean delay.

Different values $L_1$ and $L_2$ (both bigger than $N_p$) do result in identical results when the number of contenders $k$ is smaller than $\min(L_1, L_2)$, this follows from the fact that any level that is part of a CC with $k$ contenders never requires more than $k$ slots. Thus, even if we do not take $L$ into account we can still get good approximate results for low and medium load situations, validating our approach presented in [8].

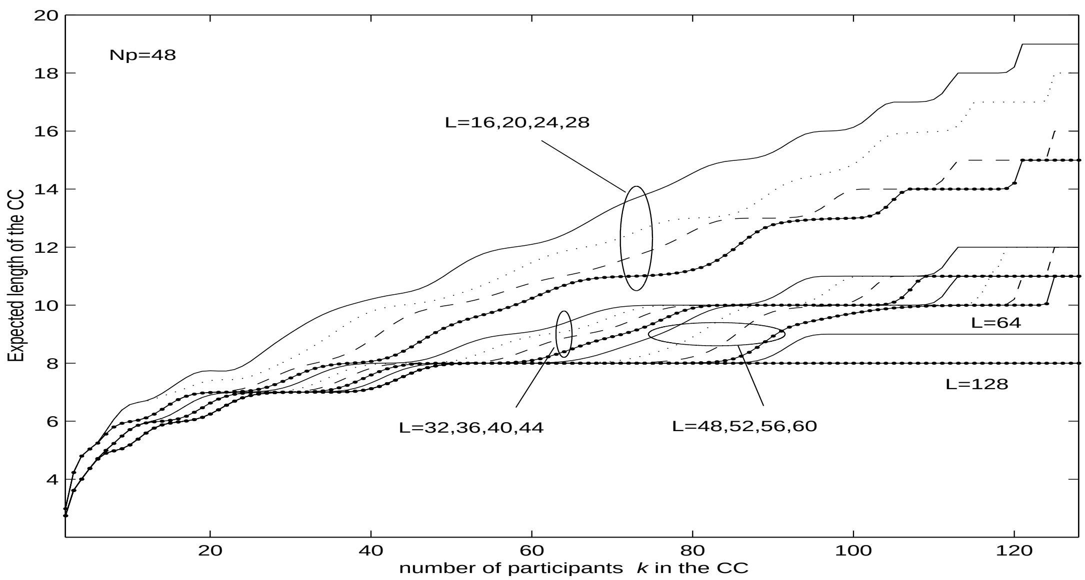


**Figure 8:** *$E[F]$ for different values of $L$*

Although we already demonstated that there is little use in choosing $N_p > L$, we also include Figure 8 for reasons of completeness. The main purpose of Figure 8 is to demonstrate that different values $L_1$ and $L_2$ do not coincide for $k$ smaller than $\min(L_1, L_2)$ when $N_p > \min(L_1, L_2)$.

### 5.3 Selecting the starting level $S_l$

In Figure 9 the influence of the starting level $S_l$ on $E[F]$ is shown ($L = 16$ and $N_p = 0$). For $S_l \leq 4$ the delay decreases for all values of $k$ when increasing the starting level $S_l$. Moreover the improvement that we get by increasing $S_l$ by one is close to one frame. For $S_l > 4$ we still have a delay improvement for large values of $k$ (a more significant one compared to $S_l \leq 4$) but a price is paid for smaller values of $k$. Note that for $S_l = 7$ we obtain a pure polling scheme. In general, looking from the delay perspective, we get the best results with $S_l = \log_2(L)$ if the contention channel has a low to medium load. For high loads a larger value for $S_l$ might be considered.

In Figure 10 the throughput results are shown for different values of $S_l$. Notice that these results are independent of $L$. It demonstrates that when the contention channel carries a low or medium load, increasing $S_l$ always increases the number of slots a CC requires. On the other hand if the load is high, better results are obtained for high values of $S_l$.

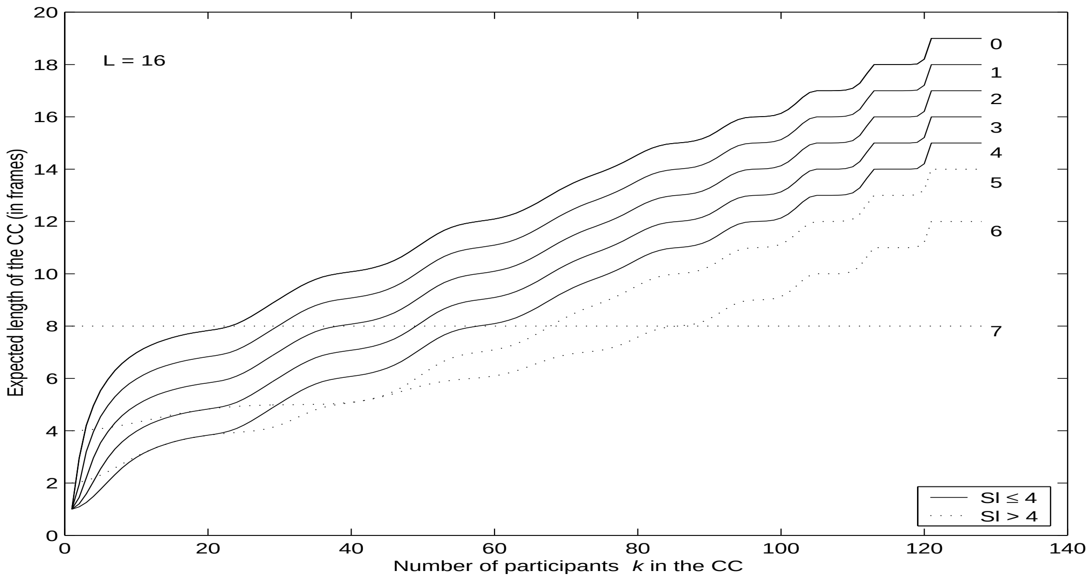


**Figure 9:** *$E[F]$ for different values of $S_l$*

### 5.4 Stability Issues

In this section we investigate the influence of the protocol parameter $L$ and the trigger value $N_p$ on the stability of the scheme under Poissonian input traffic. We define the drift $D[k]$ of the protocol as $\min(2^n, \lambda E[F \mid X = k]) - k$, where $\lambda$ is the expected number of arrivals per frame. A positive $D[k]$ implies that a CC with $k$ contenders is generally followed by a CC with more contenders, a negative value indicates an expected decrease in the number of contenders in the CC. Finally, when the number of contenders $k$ is such that $D[k] = 0$ the number of contenders is expected to remain the same, therefore we refer to

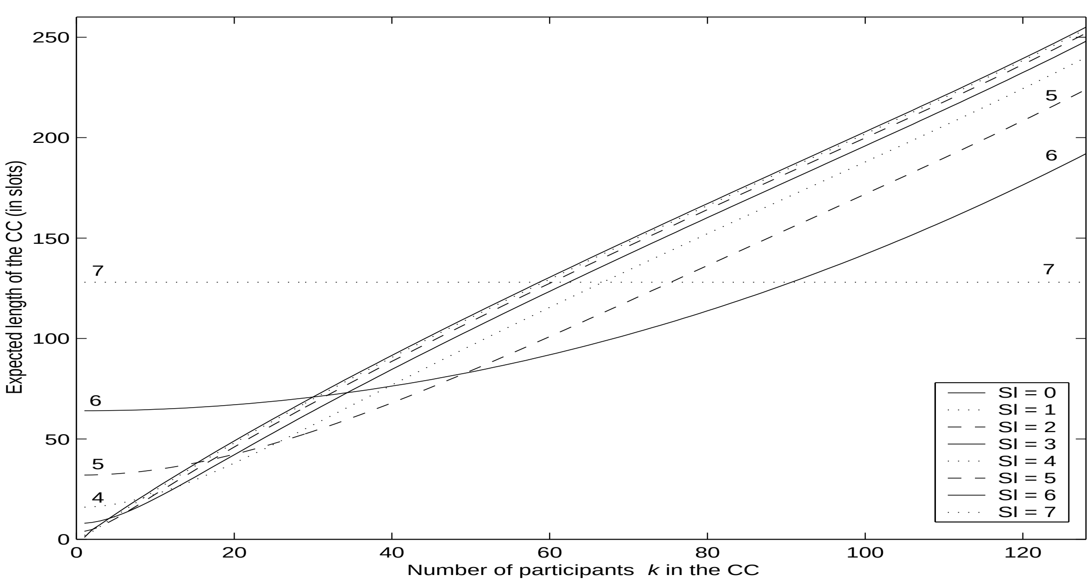


**Figure 10:** *$E[S]$ for different values of $S_l$*

these points as stability points. The scheme is expected to operate around these stability points for the majority of time.

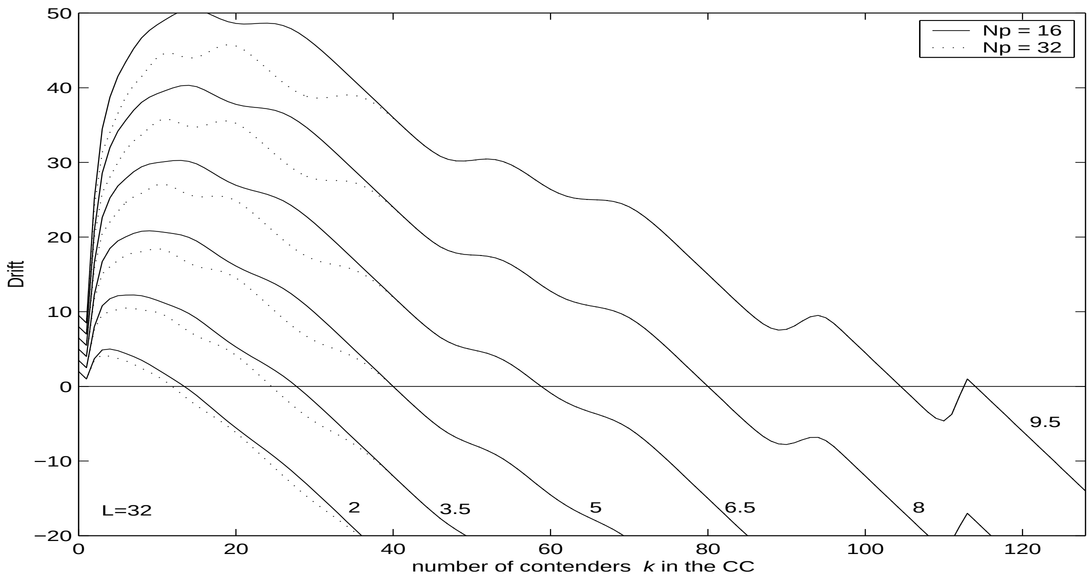


**Figure 11:** *Stability points for Poissonian input traffic*

Figure 11 shows the drift for $\lambda = 2, 3.5, 5, 6.5, 8$ and 9.5, with $L = 32$ and $N_p = 16$ or 32. With the exception of $\lambda = 9.5$ all the curves have a unique stability point. The curve with $\lambda = 9.5$ was included on purpose to show that in some rare cases the unique stability point might split into two hardly separated stability points (this is due to the oscillations in the $E[F]$ curves). Still these split stability points are not expected to endanger the general stability of the protocol.

Comparing the results for $N_p = 16$ and $N_p = 32$, we see that the stability point of the protocol remains the same for $\lambda \geq 5$ as opposed to $\lambda < 5$ where we get a smaller stability point for $N_p = 32$. Thus, the delay improvement that we get by increasing $N_p(\leq L)$ is the most significant for systems with low to medium loads.

Finally, it should be clear that the introduction of $L$ does not affect the stability of the protocol, though numerical experiments did show that the stability points might shift somewhat to the right when we decrease $L$ in systems with a high load.

### 5.5 Summary of the Best Parameter Settings

From the numerical examples the following conclusions were drawn. The polling threshold $N_p$ should not be chosen higher that $L$, the maximum number of contention slots allowed in one frame (the exception to this rule is addressed in Section 4.1). When selecting an appropriate value for $N_p$ a tradeoff has to be made between the delay and throughput characteristics where a better delay is obtained for larger values of $N_p$ ($\leq L$). If the load of the contention channel is low (or medium) the starting level $S_l$ should not be chosen larger than $\log_2(L)$. For $S_l \leq \log_2(L)$ we get a similar tradeoff as with the trigger value $N_p$. For highly loaded systems it might still be useful to select $S_l$ bigger than $\log_2(L)$ as this might result in better delay and throughput characteristics. We have shown that the influence of the parameter $L$ can be neglected for low to medium loads provided that $L$ is sufficiently large, thereby validating the approach presented in [8]. Finally, we showed that the stability of the ISA protocol is not endangered by introducing the polling threshold $N_p$ or the protocol parameter $L$.

## 6 Conclusions and Future Work

In this paper the performance analysis of the Identifier Splitting Algorithm combined with Polling (ISAP) was presented. The ISAP scheme is a random access protocol used in a wireless ATM environment to inform the Base Station about the bandwidth needs of the Mobile Stations.

The advantages of this algorithm can be summerized as follows. By using the MAC address to resolve collisions, the ISAP algorithm guarantees an upper bound on the maximum delay (like a pure polling scheme). Also, the ISAP scheme has good stability properties and can easily be configured to work efficient under both low and high load conditions (as opposed to slotted ALOHA). Finally, because of its breadth-first structure ISAP has no problems in dealing with the delayed feedback environment that is typically found in a wireless ATM access network.

Currently, we are working on a comparison between the ISAP scheme and FS-ALOHA [10]. FS-ALOHA is a contention resolution algorithm that has been shown to perform significantly better than slotted ALOHA without imposing a high degree of complexity. Meanwhile, we have also generalized many of our results to the $Q$-ary ISAP scheme [11], where $Q$ slots are provided to resolve a collision instead of 2.

## References


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

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

[3] 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 16th ITC, Edinburgh, UK*, June 1999.

[4] B. Walke, D. Petras, and D. Plassmann. Wireless ATM: Air interface and network protocols of the mobile broadband system. *IEEE Personal Communications Magazine, Aug.*, 1996.

[5] D. Petras and A. Kramling. Wireless ATM: Performance evaluation of a DSA++ MAC protocol with fast collision resolution by a probing algorithm. *Int. Journal of Wireless Information Networks*, 1997.

[6] M. Listanti, F. Mascitelli, and A. Mobilia. D2MA: A distributed access protocol for wireless ATM networks. *INFOCOM*, 1997.

[7] K.M. Sivalingam, M.B. Srivastava, P. Agrawal, and J.C. Chen. Low-power access protocols based on scheduling for wireless and mobile ATM networks. *Proc. IEEE Int. Conf. on Universal Personal Communications (ICUPC), (San Diego, CA), pp. 429-433, Oct.*, 1997.

[8] B. Van Houdt and C. Blondia. Analysis of an identifier splitting algorithm combined with polling for contention resolution in a wireless ATM access network. *Submitted to JSAC*, 1999.

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

[10] D. Vazquez-Cortizo, J. Garcia, C. Blondia, and B. Van Houdt. FIFO by sets ALOHA (FS-ALOHA): a collision resolution algorithm for the contention channel in wireless ATM systems. *Performance Evaluation, 36-37, p401-427*, 1999.

[11] B. Van Houdt and C. Blondia. Analysis of a q-ary identifier splitting algorithm combined with polling for contention resolution in a wireless ATM access network. *To appear in the Proc. of the ITC Spec. Seminar on Mobile Systems and Mobility*, 2000.