

Analysis of a Q-ary Identifier Splitting Algorithm Combined with Polling for Contention Resolution in a Wireless ATM Access Network

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`

Abstract

In this paper we generalize a prior performance analysis [5] of the binary Identifier Splitting Algorithm (ISA) with polling to the Q -ary case (to resolve a collision Q slots are provided instead of two). The analysis combines a number of queueing theoretical techniques with a fair amount of combinatorics. The ISA algorithm is a contention resolution scheme that operates on the contention channel of a wireless ATM access network. It has a tree structure and is used to inform a Base Station about the bandwidth requirements of the Mobile Stations. We evaluate the influence of the splitting factor Q on the throughput and access delay of the protocol and its interaction with the other protocol parameters.

1 Introduction

In this paper, the following class of WATM access systems is considered. A wide variety of Medium Access Control (MAC) protocols for WATM belongs to this class of MAC protocols, examples are DSA++ [8, 4], E-MAC, D²MA, [7] and many more. 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.

Two logically distinct communication channels (uplink and downlink) are used to support the information exchange between the BS and the MSs. ATM cells arriving

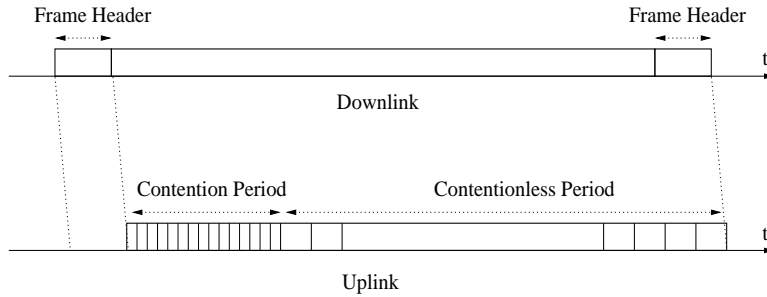


Figure 1: *Frame Structure*

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 used 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 1). Each uplink frame consists of a (variable length) contentionless 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 send 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 much 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 whether 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 -digit MAC address is assigned by the BS to every MS (we make use of Q -ary digits).

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 studied in this paper is the Identifier Splitting Algorithm (ISA) combined with polling [2, 3, 5, 7]. In this paper the performance analysis presented in [5] for the binary case is generalized to the Q -ary case. This generalization causes some changes in the analysis, still the main lines of reasoning can be retained. For the throughput analysis however a different approach from [5] was chosen.

In the remaining part of the paper a minislot is simply called a contention slot or slot, except when stated otherwise. The paper is organized as follows. The Identifier Splitting Algorithm is treated in more detail in Section 2 and its performance is evaluated in Section 3. This analysis is used in Section 4 to investigate the influence of the system parameters on the performance, in particular on the throughput and the access delay. The conclusions which follow from this investigation are summarized in Section 5.

2 The Identifier Splitting Algorithm

The Identifier Splitting Algorithm is based on the well known tree algorithm [1] and was proposed by Petras [2, 3]. 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 succeeded. 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 Q contention slots. Based on the first digit of their MAC addresses, as opposed to the classical coin flip, the MSs that are involved split up into Q distinct sets. An MS belonging to the first set uses the first slot to attempt a retransmission, the second set uses the second slot and so on.

The process of generating Q slots in the next level for each slot in which a collision occurred, is repeated level after level, each time using the next digit of the Q -ary MAC address in case of a collision. Thus during the i -th level of a CC, two MSs can only collide if their MAC addresses have the same i first digits. Therefore, provided that the address that uniquely identifies an MS is n digits long, collisions are always resolved at level n . Also notice that for every level the number of contention slots equals Q times the number of collisions of the previous level. Figure 2 shows an example of a CC for $Q = 2$ 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.

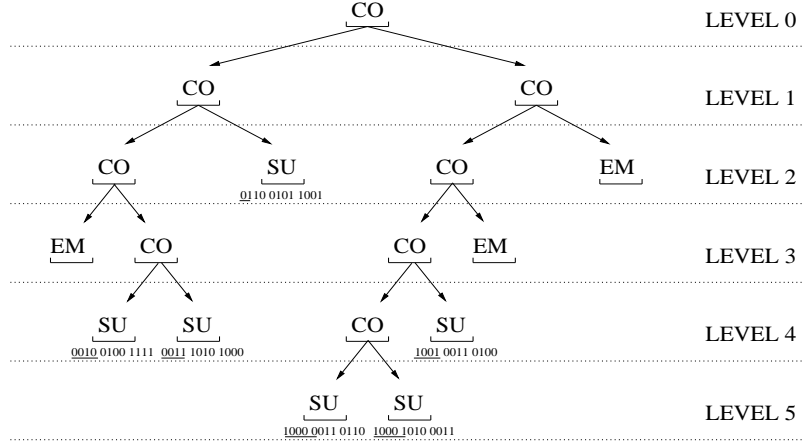


Figure 2: *Demonstrating ISA*

A level of the tree always corresponds to a single frame, except when the number of slots at level i is larger 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 level of the tree requires $x = mL + j$ slots with $1 \leq j \leq L$, slots then $m + 1$ frames are required for this level.

2.1 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 2) contains k collisions and the MAC-addresses are n digits long, then the BS concludes that the remaining competing MSs can only have kQ^{n-i} possible addresses. This follows from the fact that each slot at level i corresponds to Q^{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. In this paper N_p is assumed to be smaller than or equal to L . Moreover, in [6] we have shown that the binary ISA protocol results in a smaller throughput and worse delays if N_p is chosen bigger than L (we did not investigate $Q > 2$ in [6]).

2.2 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 Q^{S_l} contention slots. As with N_p , Q^{S_l} is assumed to be

smaller than L . During the first frame, an MS taking part in the contention cycle selects one of the Q^{S_l} slots based on the first S_l digits of its n -digit MAC address. Although it is possible to evaluate the Q -ary splitting algorithm with a dynamic starting level (see [5]), we do not consider a dynamic starting level S_l in this paper.

3 Performance Analysis

Let n be the size of the MAC-addresses (in digits). The number of MSs located within the reach of the BS is assumed to be Q^n , i.e., all MAC addresses are utilized.

Furthermore, the aggregate traffic generated by all MSs on the uplink contention channel is assumed to have a Poisson distribution with a mean of λ requests per frame. As the number of MSs is finite and equals Q^n , the number of requests generated during a CC should never exceed Q^n . Therefore we drop at random some of the arrivals if this value is exceeded (for $x > Q^n$ arrivals, we drop $x - Q^n$ arrivals). Notice that the fact that we drop these arrivals at random (instead of dropping the last $x - Q^n$ arrivals) should hardly have any influence on the numerical examples presented, as the probability of having more than Q^n arrivals during a CC is negligible. Random dropping assures that the requests arrive in a uniform way during a CC. Hence, define the random variable I_i as the number of requests generated during a CC consisting of i frames, then $P[I_i = k] = \frac{(\lambda i)^k}{k!} e^{-\lambda i}$ for $k < Q^n$ and $P[I_i = Q^n] = \sum_{k \geq Q^n} \frac{(\lambda i)^k}{k!} e^{-\lambda i}$. Notice that we do not need to consider bursty input traffic since we are observing the access channel used by an MS that wants to transmit a first request after a period of silence. In real-life systems the following holds with respect to the number of MSs participating, and their addresses.

- MSs that were successful during the last frame of a CC, will never participate in the next CC.
- Participating MSs, regardless of the frame in which they were successful, are less likely to take part in the next CC as opposed to those that did not participate at all.

To keep the model analytically tractable, both these remarks are ignored. Thus the addresses of the MSs taking part in the scheme at the beginning of a CC are uniformly distributed over the complete address space and their number is distributed according to a Poisson distribution, where the mean depends on the length of the previous CC. Finally, we assume that each level of the CC corresponds with a single frame. Therefore, we cannot use the model to study a system in which the contention channel is highly loaded.

The following random variables will be used in the sequel of this section: X_c , resp. X_a , denotes the number of contenders or participants in a CC for the ISA protocol without resp. with polling. R_c , resp. R_a , denotes the level at which the CC is resolved (i.e. the number of frames needed minus one) and this again for the ISA scheme without resp. with polling. $C_i^{(c)}$ and $C_i^{(a)}$, denotes the number of collisions at level i for both protocols. These variables range from 0 to Q^i . P_a denotes the level at which we poll for the ISA scheme with polling. If the scheme is solved without polling we let P_a be equal to $n + 1$. Furthermore, we use the symbol C_r^n to denote the number of

different possible combinations of r from n different items. In the next two subsections we demonstrate how the delay and throughput can be calculated if $S_l = 0$. The results for $S_l > 0$ can be obtained from those with $S_l = 0$. The procedure required to obtain the results for a higher starting level S_l is very similar for both the binary and the Q -ary case and therefore all details on this procedure are omitted.

3.1 The Delay Analysis

(A) We start by studying $P[R_a \leq i \mid X_a = k]$. Two cases can be considered: first, the CC might be solved before level i or at level i due to polling, secondly, it might be solved at level i without a switch to polling. Thus we get $P[R_a \leq i \mid X_a = k] = P[R_a \leq i-1 \cup P_a \leq i \mid X_a = k] + P[R_a = i \cap P_a > i \mid X_a = k]$. The first probability is discussed in **(A1)**, the second in **(A2)**.

(A1) We calculate the complementary probability mass. Let $u_i = \lfloor N_p/Q^{n-i+1} \rfloor$. By definition of the polling mechanism we have $P[R_a \geq i \cap P_a > i \mid X_a = k] = P[C_{i-1}^{(a)} > u_i \mid X_a = k]$. The right-hand side is found using the following relationship

$$P[C_{i-1}^{(a)} = u_i + x \mid X_a = k] = P[C_{i-1}^{(c)} = u_i + x \mid X_c = k] \quad (1)$$

for $x \geq 1$, but not necessarily for $x \leq 0$. This observation motivates us to study $C_i^{(c)}$ conditioned on X_c in more detail. We propose the following variation on the Inclusion-Exclusion Principle (where the first equality is easily proven by induction on k): $s(i, Q^i, k) = Q^{(n-i)k} C_k^{Q^i} / C_k^{Q^n}$,

$$s(i, l, k) = C_l^{Q^i} \sum_{l_1=0}^l Q^{(n-i)l_1} \frac{C_{l_1}^l C_{k-l_1}^{Q^n - l_1 Q^{n-i}}}{C_k^{Q^n}} - \sum_{x=1}^{Q^i - l} C_l^{l+x} s(i, l+x, k), \quad (2)$$

where $s(i, l, k) = P[C_i^{(c)} = Q^i - l \mid X_c = k]$. This concludes **(A1)**.

(A2) In the binary splitting algorithm this probability is found easily by noticing that each collision at level $i-1$ involves only two MSs, otherwise it cannot be solved at level i . Clearly with Q -ary splitting this is no longer the case. Nevertheless, we still have the following equality: $P[C_{i-1}^{(a)} = u_i + x \cap C_i^{(a)} = 0 \mid X_a = k] = P[C_{i-1}^{(c)} = u_i + x \cap C_i^{(c)} = 0 \mid X_a = k]$, for $x \geq 1$. At level i we can subdivide the address space in Q^i subsets of size Q^{n-i} based on the first i bits of the addresses. Each of these subsets is defined as a *virtual* slot at level i . We state that a *virtual* slot or subset at level i is collision free during a CC when there is at most one contender with an address that is part of that subset. Next, define $p(i, l_1, k)$ as the probability that at level i a specific set of l_1 *virtual* slots is collision free and that at level $i+1$ all *virtual* slots are collision free, given that we had k contenders in the CC. Notice that the number of collisions at level i might be smaller than $Q^i - l_1$, so other *virtual* slots that do not belong to the specific set of size l_1 might also be collision free. Hence,

$$p(i, l_1, k) = \frac{1}{C_k^{Q^n}} \sum_{j=0}^{l_1} Q^{(n-i)j} C_j^{l_1} C_{k-j}^{Q^{i+1} - l_1} Q^{(n-i-1)(k-j)}. \quad (3)$$

Next, define $q(i, l_1, k)$ as the probability that level i contains l_1 collisions and level $i + 1$ is collision free. Then, we have the following relationship between $p(i, l_1, k)$ and $q(i, l_1, k)$: $q(i, Q^i, k) = p(i, Q^i, k)$ and

$$q(i, l_1, k) = C_{l_1}^{Q^i} p(i, l_1, k) - \sum_{x=1}^{Q^i - l_1} C_{l_1}^{l_1+x} q(i, l_1 + x, k). \quad (4)$$

This completes **(A2)**.

(B) Let us now demonstrate how to obtain X_a . Clearly X_a is the steady-state vector of the Markovian process $(X_n^{(a)})_n$, where $X_n^{(a)}$ denotes the number of contenders during the n -th CC. Due to (A),

$$t_a(k, j) \stackrel{\text{def}}{=} P[X_{n+1}^{(c)} = j \mid X_n^{(c)} = k] = \sum_{t=1}^{n+1} \frac{(\lambda t)^j e^{-\lambda t}}{j!} P[R_a = t - 1 \mid X_a = k], \quad (5)$$

for $0 \leq j \leq Q^n - 1$. When $j = Q^n$ we assign the remaining probability mass. X_a is then found by solving the eigenvector problem. Applying the definition of the expected value gives us the mean number of participants $E[X_a]$ in a CC.

(C) Observing the system at a random arrival instance O_n , we require the probability that there are k contenders in the CC that contains O_n and that there are l contenders in the next CC. We denote $X_{cur}^{(a)}$ and $X_{next}^{(a)}$ as the number of participants in these two CCs. Some straightforward reasoning shows the following relationship between $X_{next}^{(a)}$, $X_{cur}^{(a)}$ and X_a : $P[X_{next}^{(a)} = l] = P[X_a = l]/E[X_a]$ and $P[X_{cur}^{(a)} = k] = \sum_{j=1}^{Q^n} P[X_a = k] t_a(k, j) j / E[X_a]$, where $t_a(k, j)$ was defined in (5).

(D) Define $\mathcal{F}_a(i, k)$ as the probability that a tagged request is successful at or before level i given that we had k contenders in the CC (for the ISA scheme with polling). Next, we define $v_i = 1 + N_p / Q^{n-i+1}$. Due to (1) the event $C_{i-1}^{(c)} \geq v_i$ coincides with $C_{i-1}^{(a)} \geq v_i$, when conditioned on X_c resp. X_a , which on its turn coincides with $P_a > i \cap R_a \geq i$. Therefore we have

$$\mathcal{F}_a(i, k) = P[R_a \leq i-1 \cup P_a \leq i \mid X_a = k] + \sum_{s \geq v_i} P[R_t \leq i \cap C_{i-1}^{(c)} = s \mid X_c = k], \quad (6)$$

where R_t denotes the level at which our tagged request is successful. The first probability was found in **(A1)**. For the second one, we define $t(i, s, k)$ as $P[R_t \leq i \cap C_{i-1}^{(c)} = Q^{i-1} - s \mid X_c = k]$. Then we get $t(i, Q^{i-1}, k) = Q^{(n-i+1)k} C_k^{Q^{i-1}} / C_k^{Q^n}$ and

$$t(i, s, k) = C_s^{Q^{i-1}} \sum_{l_1=0}^s Q^{(n-i+1)l_1} \frac{C_{l_1}^s C_{k-l_1}^{Q^n - s - Q^{n-i+1}}}{C_k^{Q^n}} \times \left(\frac{l_1}{k} + \left(1 - \frac{l_1}{k} \right) \frac{C_{k-l_1-1}^{Q^n - s - Q^{n-i+1} - Q^{n-i}}}{C_{k-l_1-1}^{Q^n - s - Q^{n-i+1} - 1}} \right) - \sum_{x=1}^{Q^{i-1}-s} C_s^{s+x} t(i, s+x, k). \quad (7)$$

With these values it is straightforward to find the second term of expression (6).

(A,B,C,D) Having done this we can calculate the mean delay. The delay can be split into two parts. The first D_1 is the time until the start of the next CC and the second D_2 is the number of frames needed until our tagged request is successful. Using the expressions in **(C)** and knowing that the arrivals are distributed uniformly within a CC (see section 3), the expected value for the first part $E[D_1]$ equals $\sum_{i=1}^{n+1} \sum_k P[X_{cur}^{(a)} = k] i / (1 + E[R_a | X_a = k]) P[R_a = i - 1 | X_a = k] i / 2$. By definition of the expected value the second part $E[D_2]$ equals $\sum_{i=0}^n \sum_{k \geq 1} P[X_{next}^{(a)} = k] (i + 1) (\mathcal{F}_a(i, k) - \mathcal{F}_a(i - 1, k))$, where $\mathcal{F}_a(i, k)$ was defined in **(D)**. As in [5], it is also possible to calculate the delay density function $D_a(x)$ (with x between 1 and $2 * (n + 1)$) as follows

$$D_a(x) = \sum_{s=1}^{\lfloor x \rfloor} \sum_{j=\lfloor x \rfloor - s}^{n+1} \sum_{l=1}^{Q^n} \frac{\mathcal{F}_a(s-1, l) - \mathcal{F}_a(s-2, l)}{j} \mathcal{G}_j(l) \cdot \left(\sum_{k=0}^{Q^n} \frac{j P[R_a = j - 1 | X_a = k]}{1 + E[R_a | X_a = k]} P[X_{cur}^{(a)} = k] \right). \quad (8)$$

where $\mathcal{G}_j(l)$, $1 \leq l \leq Q^n$ is a probability distribution that is equal to $\frac{(\lambda j)^{l-1}}{(l-1)!} e^{-\lambda j}$ for $l < Q^n$ (the remaining probability mass is assigned to $\mathcal{G}_j(Q^n)$). In this equation, s denotes the number of transmissions (including the successful transmission) a tagged request needs and j refers to the length (in frames) of the CC in which our tagged request is generated. Finally, $l - 1$ equals the number of other competitors in the CC apart from our tagged one.

3.2 The Throughput Analysis

First, we define a new set of random variables $S_i^{(a)}$, where $S_i^{(a)}$ is the number of slots required at level i when using the ISA scheme with polling. By definition of the throughput T_a we have that

$$T_a = \frac{E[X_a]}{\sum_{k=0}^{Q^n} P[X_a = k] * E[\sum_i S_i^{(a)} | X_a = k]} \quad (9)$$

Since we already know the probabilities $P[X_a = k]$ from the delay analysis, it is sufficient to find $E[\sum_i S_i^{(a)} | X_a = k]$. As the expected number of slots used equals the sum of the expected number of slots used at each level, we can focus on $E[S_i^{(a)} | X_a = k]$. Some preliminary calculations are presented in **(E)** (in **(E)**, N_p is equal to zero) and in **(F)** we calculate $E[S_i^{(a)} | X_a = k]$ using the results of **(E)**.

(E) 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. The definition of a *virtual* slot was presented in **(A2)**. Notice that the number of collisions at level i might be smaller than $Q^i - l_1$, thus other *virtual* slots that do not belong to the specific collection of size l_1 might also be collision free. A reasoning based on the Inclusion-Exclusion Principle

allows us to state the following:

$$p(i, l_1, l_2, k) = \frac{1}{C_k^{Q^n}} \sum_{j=0}^{l_1} Q^{(n-i)j} C_j^{l_1} C_s^{Q^{i+1}-Ql_1} \times \\ \sum_{j'=0}^s Q^{(n-i-1)j'} C_{j'}^s C_{k-j-j'}^{Q^n-l_1Q^{n-i}-sQ^{n-i-1}} - \sum_{x=1}^{Q^{i+1}-Ql_1-s} C_s^{s+x} p(i, l_1, l_2+x, k),$$

with $s = l_2 - Ql_1$ and with $p(i, l_1, l_2, k) = 0$ for $l_2 < Ql_1$. 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 $s(i, l_1, l_2, k)$: $s(i, Q^i, l_2, k) = p(i, Q^i, l_2, k)$ and

$$s(i, l_1, l_2, k) = C_{l_1}^{Q^i} p(i, l_1, l_2, k) - \sum_{x=1}^{Q^i-l_1} C_{l_1+x}^{l_1+x} s(i, l_1+x, l_2, k). \quad (10)$$

This concludes part **(E)**.

(F) Since the expected number of slots at level 0 and 1 are straightforward to obtain, we can focus on $E[S_i^{(a)} | X_a = k]$ for $i \geq 2$. We distinguish between 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, $E[S_i^{(a)}] = P(E_1^{(i)})E[S_i^{(a)} | E_1^{(i)}] + P(E_2^{(i)})E[S_i^{(a)} | E_2^{(i)}] + P(E_3^{(i)})E[S_i^{(a)} | E_3^{(i)}]$. Provided that the first event $E_1^{(i)}$ occurs, the expected number of slots $S_i^{(a)}$ at level i equals zero. As for the other two, we can rewrite the previously mentioned events as: $E_1^{(i)} = C_{i-2}^{(c)} < v_{i-1}$, $E_2^{(i)} = C_{i-2} \geq v_{i-1} \cap C_{i-1} < v_i$ and $E_3^{(i)} = C_{i-2} \geq v_{i-1} \cap C_{i-1} \geq v_i$ (v_i was defined in **(D)**). Moreover,

$$P(E_2^{(i)} | X = k)E[S_i^{(a)} | X = k \cap E_2^{(i)}] = \\ \sum_{l_1 \geq v_{i-1}} \sum_{l_2 < v_i} s(i-2, Q^{i-2} - l_1, Q^{i-1} - l_2, k) Q^{n-i+1} l_2,$$

and finally $P(E_3^{(i)} | X = k)E[S_i^{(a)} | X = k \cap E_3^{(i)}] = \sum_{l_1 \geq v_{i-1}} \sum_{l_2 \geq v_i} s(i-2, Q^{i-2} - l_1, Q^{i-1} - l_2, k) Q l_2$, where $s(i, l_1, l_2, k)$ was found in **(E)**.

4 Numerical Results

In this section, we investigate the influence of the the splitting factor Q and its interaction with the arrival rate λ , the trigger value N_p and to some extent the starting level S_l . The system parameters are set as follows. The splitting factor Q equals 2, 3 or 4, we refer to these three cases as the binary, ternary and quaternary scheme. The number of digits n depends upon the value of Q . In the binary case n equals 8, in the

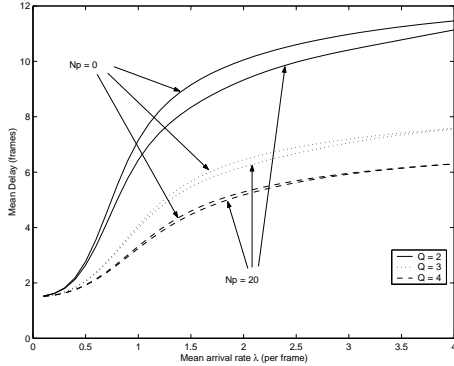


Figure 3: The impact of Q and N_p on the mean delay

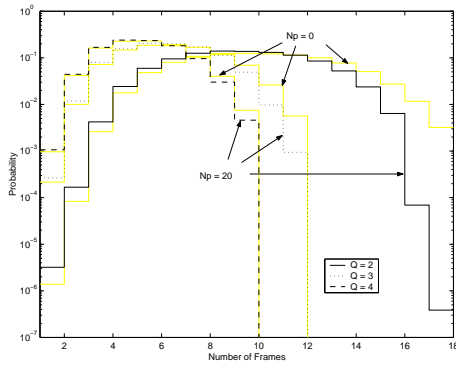


Figure 4: The impact of Q and N_p on the delay density function

ternary case n equals 5 and finally in the quaternary case n equals 4. Thus for the binary and quaternary scheme we are able to support 256 MSs, in the ternary case we can have at most 243 MSs. This small difference in the size of the address space should hardly have any effect on the results because on average the number of participating MSs in a CC is always much smaller than Q^n .

4.1 The Influence of the Splitting Factor and the Polling Threshold on the System Performance

In Figures 3 and 4, the influence of Q on the mean delay and the delay density function is shown for $N_p = 0$ and $N_p = 20$. First, a larger splitting factor Q results in a smaller delay (mean and quantiles). Also, the delay differs much more when we compare the binary and ternary scheme as opposed to the ternary and quaternary scheme. In general, a larger value for Q results in a smaller delay. Also, the delay improvement that we get from increasing Q by one decreases as Q grows. Indeed, increasing Q by one results in $1/Q$ times as many slots to resolve a collision.

Secondly, Figures 3 and 4 show that the influence of the polling threshold N_p decreases as the splitting factor Q increases (mean and quantiles), thus the polling feature is the most attractive for the binary ISA protocol. A general remark on N_p is that different values of N_p only result in a different behavior when there is at least one multiple of Q^2 in between.

Figures 5 and 6 demonstrate the influence of Q on the throughput results for $N_p = 0$ and $N_p = 20$. For $N_p = 0$, the highest throughput is obtained with the ternary scheme except for very low load conditions where the binary scheme is slightly superior. For $N_p = 20$, we also get the best results for the ternary scheme, in this case the binary scheme no longer dominates the quaternary scheme for λ around one. Taking both the delay and throughput into account, we may conclude that it is better to use a ternary scheme as opposed to a binary one. The choice between the ternary and the quaternary is a tradeoff between the delay and throughput. It has to be mentioned that there do exist some values for N_p for which the binary scheme has better throughput results

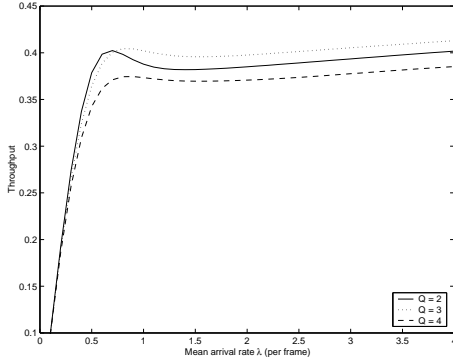


Figure 5: *The impact of Q on the throughput results for $N_p = 0$*

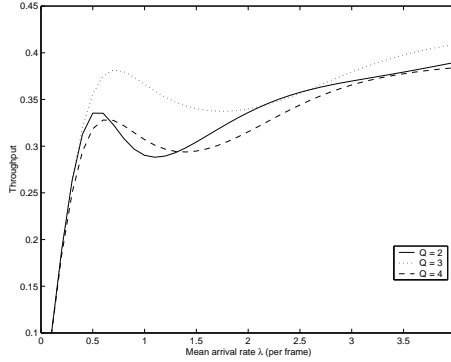


Figure 6: *The impact of Q on the throughput results for $N_p = 20$*

than the ternary, e.g., for $27 = 3^3 \leq N_p < 32 = 2^5$.

4.2 The Interaction between the Splitting Factor and the Starting Level

We only show results for $S_l = 0$ and $S_l = 1$, although the analytical model imposes no restraints on the value of the starting level S_l (expect that Q^{S_l} is bounded by L). Figures 7 and 8 show the influence of the starting level S_l and its interaction with Q for $N_p = 0$. First, the absolute delay improvement that we obtain for $S_l = 1$ is very similar in all three cases (binary, ternary and quaternary). In general, the absolute delay improvement that we obtain from a higher starting level is independent of the splitting factor Q . As for the throughput, we always get a slight improvement when we increase the starting level to one, except under low load conditions. Also, the throughput losses suffered under low load conditions become more severe as Q increases. Therefore, if we want to combine a higher starting level ($S_l \leq 1$) with a higher splitting factor Q , we suggest that it is best to make the starting

level S_l dynamic between S_{min} and S_{max} , with $S_{min} = 0$ or 1 (see [5] for the details: in [5] the procedure is given for the binary scheme but it also holds for Q -ary schemes).

5 Conclusions

In this paper we have generalized a prior performance analysis [5] of the binary Identifier Splitting Algorithm (ISA) with polling to the Q -ary case. From the numerical examples presented the following conclusions with respect to Q can be drawn.

First, better delay characteristics can be obtained (both mean and quantiles) by selecting a higher splitting factor Q . The best throughput results are obtained using the ternary scheme (with the exception of some exotic cases). Also, looking from the throughput perspective it is hard to predict whether the binary or quaternary scheme performs best. Taking both the throughput and delay into account, we may conclude that the ternary scheme outperforms the binary scheme, while the choice between the ternary and quaternary is a tradeoff between the delay and throughput characteristics.

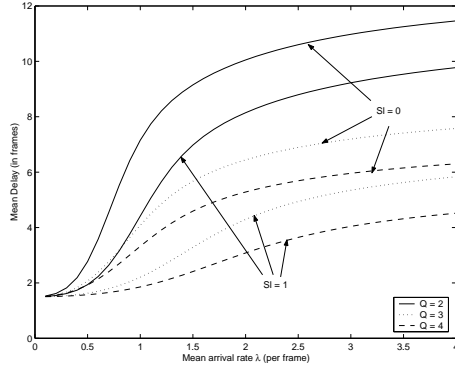


Figure 7: The impact of S_l and Q on the mean delay

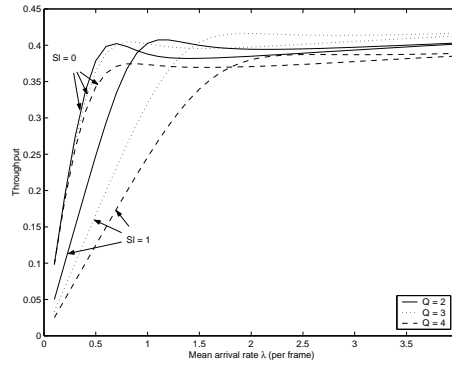


Figure 8: The impact of S_l and Q on the throughput results

Secondly, the introduction of the polling feature is the most effective in reducing the delay in the binary scheme (the higher Q the less effective).

Finally, a higher starting level results in lower delays but under low load conditions a high price is paid in terms of the throughput (the higher Q the more severe the throughput losses). As with the binary scheme, this loss of throughput under low load conditions can be combated by making the starting level dynamic (this is out of the scope of this paper).

References

- [1] J.I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Trans. Inform. Theory*, Vol 25, No 5, pp. 319 - 329, 1979.
- [2] D. Petras. Medium access control protocol for wireless, transparent ATM access in MBS. *RACE Mobile Telecommunications Summit, (Cascais, Portugal)*, 1995.
- [3] D. Petras and A. Kramling. Fast collision resolution in wireless ATM networks. *2nd MATHCOM, Vienna, Austria, Feb*, 1997.
- [4] 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.
- [5] 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 for publication*, 1999.
- [6] B. Van Houdt and C. Blondia. Performance analysis of the identifier splitting algorithm with polling in wireless ATM networks. *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 16th ITC, Edinburgh, UK*, June 1999.
- [8] B. Walke, D. Petras, and D. Plassmann. Wireless ATM: Air interface and net-

work protocols of the mobile broadband system. *IEEE Personal Communications Magazine*, Aug., 1996.