Analysis of an Identifier Splitting Algorithm Combined with Polling (ISAP) for Contention Resolution in a Wireless Access Network

Benny Van Houdt and Chris Blondia, *Member, IEEE*

***Abstract*—In this paper, a contention resolution scheme for an uplink contention channel in a wireless access network is presented. The scheme consists of a tree algorithm, namely the identifier splitting algorithm (ISA), combined with a polling scheme. Initially, ISA is used, but at a certain level of the tree, the scheme switches to polling of the stations. This scheme is further enhanced by skipping a few levels in the tree when starting the algorithm (both in a static and a dynamic way) and by allowing multiple instants simultaneously. An analytical model of the system and its variants leads to the evaluation of its performance, by means of the delay density function and the throughput characteristics. This model is used to investigate the influence of the packet arrival rate, the instant at which the ISA scheme switches to polling, the starting level of the ISA scheme, and the use of multiple instances on the mean delay, the delay quantiles, and the throughput.**

***Index Terms*—Access networks, contention resolution, ISAP.**

I. Introduction

Wireless LANs should support the various quality of service (QoS) levels defined in the fixed network (e.g., Internet DiffServ classes, service categories in ATM). This ability will heavily depend on the definition of the medium access control protocol, needed to arbitrate access to the shared radio medium.

Consider a cell in a wireless packet network (Fig. 1), consisting of a base station (BS) serving a finite set of mobile stations (MS) by means of a shared radio channel. Packets arriving at the BS are broadcasted downlink. The packets originating from an MS share the radio medium using a MAC protocol. The access technique is time division multiple access (TDMA) and frequency division duplex (FDD). Each MS in a cell receives a unique address that is used by the MAC layer in the BS.

An important class of access networks with these characteristics consists of wireless ATM LANs (e.g., MEDIAN [13], Magic WAND [6], SAMBA [8], AWACS, AMUSE). A number of medium access control (MAC) protocols have been proposed for such systems, such as MASCARA [7], PRMA [4], DSA++ [8], DQRUMA [5], and others [3], [17], [12], [14].

The MAC protocols considered in the above references have a centralized control located in the BS. The uplink channel is used by the MSs to inform the BS about their bandwidth needs (through requests) and to transmit data packets. The downlink channel is used for acknowledgments, information about the permission to use the uplink channel (permits) and data packets. Both the information on the uplink and the downlink channels is grouped into fixed length frames, leading to significant reductions in the power consumption. The requests issued by the MSs are usually piggybacked with the uplink data packets that have already been scheduled. Unfortunately, such a piggybacking scheme fails for new MSs entering the network, and MSs which become active after having remained silent for a period of time or have sudden increases in their uplink traffic. Therefore, an additional uplink contention channel is provided to allow these MSs to inform the BS about their bandwidth needs. A dynamic portion of each fixed length frame is assigned to this contention channel (Fig. 2).

We propose to use a contention resolution scheme based on a tree algorithm, the identifier splitting algorithm (ISA) defined in [9]–[11]. In order to improve its performance, the ISA is combined with a polling scheme (ISAP). The resulting scheme is further enhanced by skipping a few levels in the tree when starting the algorithm, both in a static and dynamic way and by allowing multiple instants simultaneously. The throughput and delay density functions of ISAP are analytically evaluated. The application of the analysis on numerical examples illustrates the influence of the different system parameters on the delay and throughput characteristics.

The structure of the paper is as follows. Section II presents a description of the ISAP protocol. In Section III, the analytical model to evaluate the performance of the protocol is presented, while numerical examples are given in Section IV. Conclusions are drawn in Section V.

II. The Contention Resolution Scheme

A. The Identifier Splitting Algorithm (ISA)

The identifier splitting algorithm is based on the well-known tree algorithm [2], [1], [15] and was proposed by Petras in [9]–[11]. 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. Requests that intend to use the contention resolution scheme generated by the MSs during a CC have to wait for participation till the start of the next CC.

In the first frame of a cycle, one contention slot is available. Any MS having a request ready, at the start of this CC, makes

Manuscript received May 3, 1999.

The authors are with the University of Antwerp, Department of Mathematics and Computer Science, Performance Analysis of Telecommunication Systems Research Group, Universiteitsplein, 1, B-2610 Antwerp, Belgium (e-mail: {vanhoudt; blondia}@uia.ua.ac.be).

Publisher Item Identifier S 0733-8716(00)09211-8.

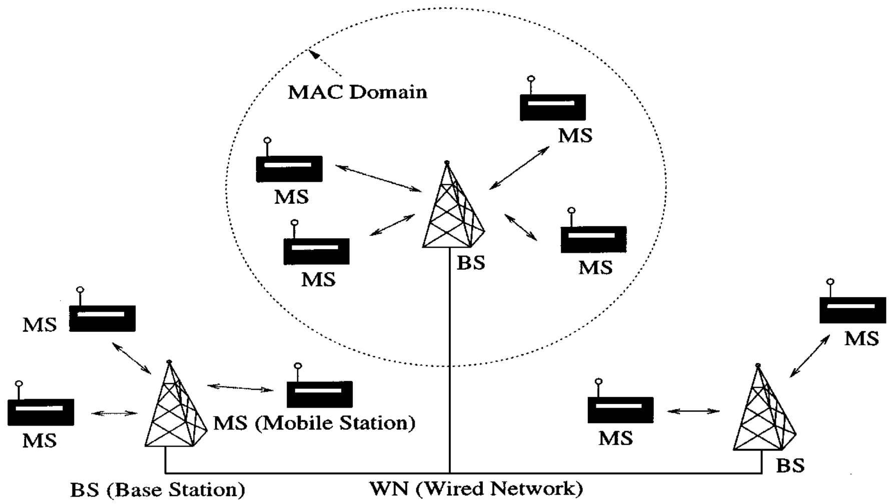


Fig. 1. Reference configuration of the system.

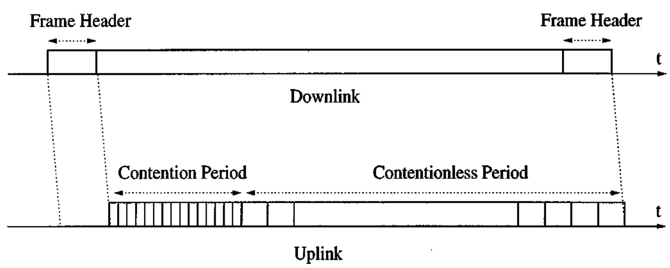


Fig. 2. Frame structure.

use of this slot. Next the BS checks whether a transmission was successful 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. 1) An MS sending in this slot was successful or the slot was empty. In this case, a new CC starts in the next frame. 2) The transmission was not successful, i.e., a collision occurred. In this case, the next (second) frame 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.

The process of generating two slots in the next frame, for each slot in which a collision occurred, is repeated frame after frame, each time using the next bit of the MAC address in case of a collision. Thus, during the ith frame of a CC, two MSs can only collide if their MAC addresses have the same $i-1$ first bits. Therefore, provided that the address that uniquely identifies an MS is n bits long, all collisions are always resolved in $n+1$ frames. Also notice that for every frame, the number of contention slots equals twice the number of collisions of the previous frame. The number of contention slots one can use in a single frame is severly bounded in a the wireless medium. In [16] we addressed this issue and have shown that this has no significant impact on the performance of ISAP for small and medium arrival rates. Finally, Fig. 3 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.

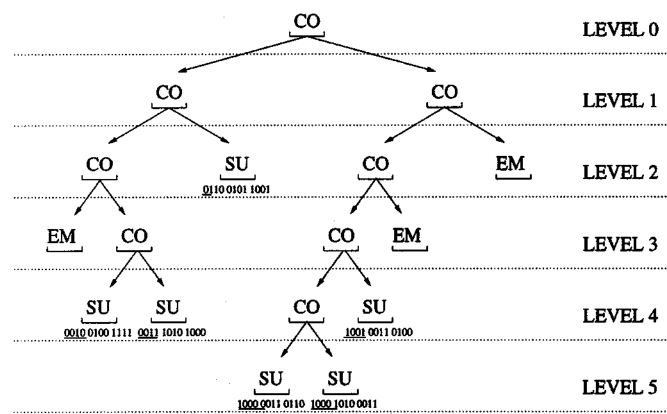


Fig. 3. Demonstrating ISA.

B. The Identifier Splitting Algorithm Combined with Polling (ISAP)

Each slot at the ith level of a CC corresponds to 2^{n-i} addresses. Therefore, provided that level i holds k collisions, the BS concludes that the remaining competing MSs can only have $k2^{n-i}$ possible addresses. This information can be used by the BS to take a decision whether to continue to use the ISA protocol or to switch to polling. The basic idea here is that when the size of the remaining MAC address space becomes smaller than some predefined value, say N_p, the protocol switches to polling. Polling, in this context, means that in the next frame, one slot is provided for each address in the remaining address space.

C. MS Behavior

The behavior of the different MSs, located within the observed cell, is described in Fig. 4 by means of a flow chart. As long as an MS is able to piggyback its requests, it remains in the *piggybacked* state. When a packet generated by an MS finds the queue with packets waiting for transmission empty, then a transition occurs to the state *blocked*. There it remains until the

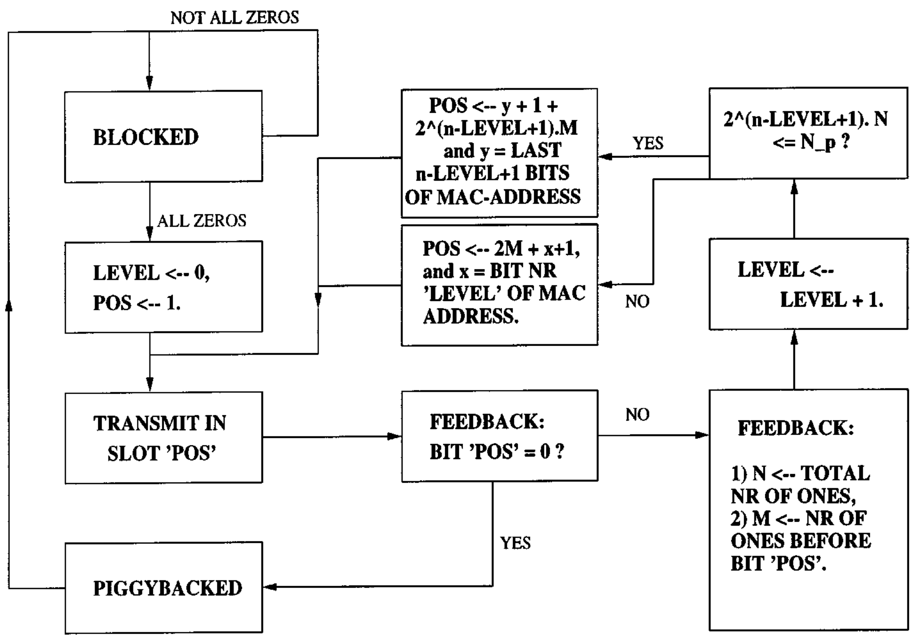


Fig. 4. The flow chart of an MS.

current CC is solved, checking the feedback field at the beginning of every frame. The feedback is given by a set of bits, one for each contention slot, where a zero indicates a success (or an empty slot) and a one a failure. Notice that one feedback bit is sufficient as we do not take capture effects into account. Once the current CC has ended, i.e., all feedback bits equal zero, two parameters LEVEL and POS are initialized. LEVEL indicates the current level of the CC, and therefore its value is incremented by one at the start of each frame during a CC. POS is a variable that holds the number of the contention slots to be used by the MS (the slots are numbered starting from one). After initialization, the *transmission* state is entered. While in this state, a transmission will take place in slot number POS and the result is found by checking the corresponding feedback bit. If successful, we return to the *piggybacked* state, otherwise the MS sets the parameters N and M, and increments LEVEL by one. Next, the MS checks to see whether a switch to the polling scheme is made, and depending on this result assigns a new value to POS. Finally the MS returns to the *transmission* state, and this routine is repeated until a successful transmission occurs.

D. Skipping the First Few Levels

In this subsection we consider CCs that provide more than one slot during their first frame. The starting level S_l is said to equal i if 2^i slots are provided in the first frame of a CC. At first, the starting level of each CC is fixed at a predefined value S_l. It is expected that this has a positive impact on the delay, and for low load situations a negative influence on the throughput. In order to minimize these throughput losses, we propose a scheme that changes the starting level dynamically, between level $S_{\min}$ and $S_{\max}$, depending on the length of the previous CC. To make this decision, the system load ρ is not taken into account, as this value is hard to measure or predict in real systems.

The starting levels are defined using the two threshold values B_l and B_m. Suppose that at some point in time the starting level equals S_l and L is the length of this CC. Then the new starting level S_l' obeys the following equation:

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

Clearly all MSs, wanting to access the contention channel, need to be aware of the current starting level. We suggest that this knowledge is broadcasted by the BS at the start of every CC. Therefore, it is not necessary for all MSs, including those that do not use the contention channel, to keep track of the lengths of the CCs.

E. Multiple Instances of ISA

It is possible to use multiple instances of the ISA protocol simultaneously. For instance, in the case of two instances: the first instance is used by all MSs with the first bit of their address equal to 0, the second instance is devoted to the other MSs, i.e., with a MAC address that starts with 1. In general, 2^m different instances of ISA can be used simultaneously. Each instance corresponds with a partition of the address space and MSs always participate in the same instance. The MS behavior is very similar as before, thus no extra complexity is added. One of the advantages of this method is that the number of contention slots is spread more uniformly over consecutive frames, as different instances are not necessarily in phase, i.e., the tops of the different trees might occur in different frames. Although it is possible to combine multiple instances and polling, this is beyond the scope of this paper.

III. Performance Analysis

A. The Analytical Model

We define n as the size of the MAC-addresses (in bits). The number of MSs located within the reach of the BS is 2^n, i.e., all MAC addresses are utilized. We assume that the aggregate traffic generated by all MSs, on the uplink contention channel, is Poisson distributed with a mean of λ requests per frame. As the number of MSs is finite and equals 2^n, the number of requests, generated during a CC, should never exceed 2^n. Therefore, we drop, at random, some of the arrivals if this value is exceeded (for $x > 2^n$ arrivals, we drop $x - 2^n$ arrivals at random). In this way, we assure 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, as $P[I_i = k] = e^{-\lambda i}\,(\lambda i)^k/k!$ for $k < 2^n$ and $P[I_i = 2^n] = e^{-\lambda i}\,\sum_{k \geq 2^n}(\lambda i)^k/k!$. Notice that we do not need to consider bursty input traffic since we are observing the access channel used by an MS 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 make the system 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 Poission distribution, where the mean depends on the length of the previous CC.

The following random variables will be used. X_c, respectively, X_a, denotes the number of contenders or participants in a CC for the ISA protocol, respectively, ISAP protocol. R_c, respectively, 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, respectively, ISAP scheme. $C_i^{(c)}$ and $C_i^{(a)}$ denote the number of collisions at level i for both protocols. These variables range from 0 to 2^i. P_a denotes the level at which we poll for the ISAP scheme. If the CC 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.

B. *The Delay Analysis*

In this subsection, we present the delay analysis of the ISAP scheme when the starting level is fixed at zero. Results for the ISA scheme are obtained by setting $N_p = 0$.

Discussion A: We start by studying the random variable R_a when conditioned on X_a. Two cases can be considered. First, the CC might be solved before level i or at level i due to polling. Second, it might be solved at level i without a switch to polling. Thus we get

$$P[R_a \leq i \,|\, X_a = k] = P[R_a < i \cup P_a \leq i \,|\, X_a = k] + P[R_a = i \cap P_a > i \,|\, X_a = k]. \quad (2)$$

The first probability is explored in Discussion A1, the second in Discussion A2.

Discussion A1: We calculate the complementary probability mass. Clearly, by definition of the polling mechanism, we have

$$P[R_a \geq i \cap P_a > i \,|\, X_a = k] = P\left[C_{i-1}^{(a)} > \left\lfloor \frac{N_p}{2^{n-i+1}} \right\rfloor \middle| X_a = k\right]. \quad (3)$$

In order to obtain the right-hand side of (3), one easily obtains the following relationship:

$$P\left[C_{i-1}^{(a)} = \left\lfloor \frac{N_p}{2^{n-i+1}} \right\rfloor + x \middle| X_a = k\right] = P\left[C_{i-1}^{(c)} = \left\lfloor \frac{N_p}{2^{n-i+1}} \right\rfloor + x \middle| X_c = k\right] \quad (4)$$

for $x \geq 1$, but not necessarily for $x \leq 0$. Thus, it is sufficient to find the probability that we have exactly l collisions at level i of a CC given k participants for the ISA scheme. In order to calculate these values in a numerically stable way, we propose the following variation on the inclusion–exclusion principle:

$$s(i, 2^i, k) = \frac{2^{(n-i)k} C_k^{2^i}}{C_k^{2^n}} \quad (5)$$

$$s(i, l, k) = C_l^{2^i} \sum_{l_1=0}^{l} 2^{(n-i)l_1} \frac{C_{l_1}^{l} C_{k-l_1}^{2^n - l2^{n-i}}}{C_k^{2^n}} - \sum_{x=1}^{2^i - l} C_l^{l+x} s(i, l+x, k) \quad (6)$$

where $0 \leq l < 2^i$ and $s(i, l, k) = P[C_i^{(c)} = 2^i - l \,|\, X_c = k]$.

Discussion A2: In this case, each collision at level $i-1$ involves only two MSs, otherwise it cannot be solved at level i. The probability that such a collision is solved, at level i, clearly equals $2^{n-i}/(2^{n-i+1} - 1)$. Thus, by means of the multivariate hypergeometric distribution, we get

$$P[R_a = i \cap P_a > i \,|\, X_a = k] = \sum_{u=u_{i-1}+1}^{\lfloor k/2 \rfloor} 2^{(n-i+1)(k-2u)} \frac{\left(C_2^{2^{n-i+1}}\right)^u C_u^{2^{i-1}} C_{k-2u}^{2^{i-1}-u}}{C_k^{2^n}} \cdot \left(\frac{2^{n-i}}{2^{n-i+1} - 1}\right)^u \quad (7)$$

where u_i denotes $\lfloor N_p / 2^{n-i} \rfloor$.

Discussion B: We define $p_a(k, i+1) = P[R_a = i \,|\, X_a = k]$. 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 nth CC. Due to Discussion A,

$$t_a(k, j) \triangleq P\left[X_{n+1}^{(a)} = j \middle| X_n^{(a)} = k\right] = \sum_{t=1}^{n+1} \frac{(\lambda t)^j e^{-\lambda t}}{j!} p_a(k, t) \quad (8)$$

for $0 \leq j \leq 2^n - 1$. When $j = 2^n$, we assign the remaining probability mass. X_a is then found by solving the eigenvector problem. We denote $E[X_a]$ as the mean number of participants in a CC.

Before we can calculate the delay, we still need to make the following observation. Assume a random arrival in a CC, then we need to know the probability that there are k contenders in this CC and that there will be l 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 that the following relationships between $X_{next}^{(a)}$, $X_{cur}^{(a)}$ and X_a hold: $P[X_{next}^{(a)} = l] = P[X_a = l]l/E[X_a]$ and $P[X_{cur}^{(a)} = k] = \sum_{j=1}^{2^n} P[X_a = k]t_a(k,j)j/E[X_a]$ where $t_a(k,j)$ was defined in (8).

Combining Discussions A and B: Having done this, we can calculate the mean delay. Clearly the delay consists of two parts D_1 and D_2. 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. Knowing that the arrivals are distributed uniformly within a CC, the expected value for the first part D_1 equals

$$E[D_1] = \sum_{i=1}^{n+1} \sum_k P\left[X_{cur}^{(a)} = k\right] \frac{i\, p_a(k,i)}{1 + E[R_a \,|\, X_a = k]}\, i/2. \tag{9}$$

By definition of the expected value, the second part is found as

$$E[D_2] = \sum_{i=0}^{n} \sum_{k \geq 1} P\left[X_{next}^{(a)} = k\right] \times (i+1)(\mathcal{F}_a(i,k) - \mathcal{F}_a(i-1,k)) \tag{10}$$

where $\mathcal{F}_a(i,k)$ denotes the probability that a tagged request is successful at or before level i given that there were $k-1$ other contenders ($\mathcal{F}_a(-1,k)$ is zero in the expression above). $\mathcal{F}_a(i,k)$ is found as follows. Denote $u_{i-1}+1$ as v_i, thus $v_i = 1 + \lfloor N_p/2^{n-i+1} \rfloor$. In Discussion A1, it was argued that the event $C_{i-1}^{(c)} \geq v_i$ is the same as $C_{i-1}^{(a)} \geq v_i$, when conditioned on X_c, respectively, X_a, which in its turn coincides with $P_a > i \cap R_a \geq i$. Therefore, we have

$$\mathcal{F}_a(i,k) = P[R_a \leq i-1 \cup P_a \leq i \,|\, X_a = k] + \sum_{s \geq v_i} P\left[R_t \leq i \cap C_{i-1}^{(c)} = s \,\middle|\, X_c = k\right] \tag{11}$$

where R_t denotes the level at which our tagged request is successful. This first probability was found in Discussion A1. The second one is calculated using (5) and (6) as follows. We define $t(i,s,k)$ as $P[R_t \leq i \cap C_{i-1}^{(c)} = 2^{i-1} - s \,|\, X_c = k]$. Then we get

$$t(i, 2^{i-1}, k) = \frac{2^{(n-i+1)k} C_k^{2^{i-1}}}{C_k^{2^n}} \tag{12}$$

$$t(i,s,k) = C_s^{2^{i-1}} \sum_{l_1=0}^{s} 2^{(n-i+1)l_1} \frac{C_{l_1}^s C_{k-l_1}^{2^n - s2^{n-i+1}}}{C_k^{2^n}} \cdot \left(\frac{l_1}{k} + \left(1 - \frac{l_1}{k}\right) \frac{C_{k-l_1-1}^{2^n - s2^{n-i+1} - 2^{n-i}}}{C_{k-l_1-1}^{2^n - s2^{n-i+1} - 1}}\right) - \sum_{x=1}^{2^{i-1}-s} C_s^{s+x} t(i, s+x, k). \tag{13}$$

With these values, it is straightforward to find the second term of (11). Adding $E[D_1]$ and $E[D_2]$ results in the mean delay.

Using Discussions A and B, it is easy to find the delay density function $D_a(x)$ [with x between 1 and $2(n+1)$]. This function is the following step function:

$$D_a(x) = \sum_{s=1}^{\lfloor x \rfloor} \sum_{j=\lceil x \rceil - s}^{n+1} \sum_{l=1}^{2^n} \frac{\mathcal{F}_a(s-1,l) - \mathcal{F}_a(s-2,l)}{j} \mathcal{G}_j(l) \cdot \left(\sum_{k=0}^{2^n} \frac{jP[R_a = j-1 \,|\, X_a = k]}{1 + E[R_a \,|\, X_a = k]} P\left[X_{cur}^{(a)} = k\right]\right) \tag{14}$$

where $\mathcal{G}_j(l) = ((\lambda j)^{l-1}/(l-1)!)e^{-\lambda j}$ for $l < 2^n - 1$ and $\mathcal{G}_j(2^n) = \sum_{j \geq 2^n - 1} ((\lambda j)^{l-1}/(l-1)!)e^{-\lambda j}$. In (14), s denotes the number of transmissions (including the successful transmission) a tagged request needs, 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 apart from our tagged one.

C. *The Throughput Analysis*

1) The Identifier Splitting Algorithm: In this subsection we determine the throughput for the ISA scheme. First, we define two more sets of random variables $S_i^{(c)}$, respectively, $S_i^{(a)}$, being the number of slots used at level i by ISA, respectively, ISAP. From the foregoing, we already obtained $P[X_c = k]$ ($P[X_a = k]$ for $N_p = 0$) and thus the throughput T_c is found as

$$T_c = \frac{E[X_c]}{\sum_{k=0}^{2^n} P[X_c = k] E\left[\sum_i S_i^{(c)} \,\middle|\, X_c = k\right]}. \tag{15}$$

Clearly,

$$E\left[\sum_i S_i^{(c)} \,\middle|\, X_c = k\right] = 1 + \sum_{i=1}^{n} E\left[S_i^{(c)} \,\middle|\, X_c = k\right]. \tag{16}$$

Moreover, the expected number of slots at level i equals twice the expected number of collisions at level $i-1$; while the expected number of collisions at level i matches

$$E\left[C_i^{(c)} \,\middle|\, X_c = k\right] = 2^i \left(1 - \frac{C_k^{2^n - 2^{n-i}}}{C_k^{2^n}} - 2^{n-i} \frac{C_{k-1}^{2^n - 2^{n-i}}}{C_k^{2^n}}\right). \tag{17}$$

2) The Identifier Splitting Algorithm Combined with Polling: We already know the probabilities $P[X_a = k]$ from the delay analysis. Thus, it is sufficient to find $E[\sum_i S_i^{(a)} \,|\, X_a = k]$. As before, the expected number of slots used equals the sum of the expected number of slots used at each level. By definition of the ISAP scheme, we have

$$E\left[S_i^{(a)} \,\middle|\, X_a = k\right] = P[P_a = i \,|\, X_a = k] E\left[S_i^{(a)} \,\middle|\, X_a = k \cap P_a = i\right] + P[P_a > i \cap R_a \geq i \,|\, X_a = k] \cdot E\left[S_i^{(a)} \,\middle|\, X_a = k \cap P_a > i \cap R_a \geq i\right]. \tag{18}$$

The second probability was obtained in Discussion A1, the first one is found as $P[R_a \leq i-1 \cup P_a \leq i \,|\, X_a = k]$ minus $P[R_a \leq$

$i-1 \mid X_a = k]$, two results that were also obtained in Discussion A. Both the expected values (for $i \geq 2$ since $S_0^{(a)}$ and $S_1^{(a)}$ are trivial to obtain) are presented in Discussions C and D.

Discussion C: First, consider $E[S_i^{(a)} \mid X_a = k \cap P_a > i \cap R_a \geq i]$. In this case, the number of slots used at level i is twice the number of collisions of the previous level. Also, in Discussion A1, it was shown that the event $P_a > i \cap R_a \geq i$ is the same as $C_{i-1}^{(c)} \geq v_i$, when conditioned on X_a, respectively, X_c. Hence, it is sufficient to find $E[C_{i-1}^{(c)} \mid X_c = k \cap C_{i-1}^{(c)} \geq v_i]$. Using (4), one finds

$$E\left[C_{i-1}^{(c)} \,\middle|\, X_c = k \cap C_{i-1}^{(c)} \geq v_i\right] = v_i + \sum_{s=0}^{2^{i-1}-v_i} s \frac{P\left[C_{i-1}^{(c)} = s + v_i \,\middle|\, X_c = k\right]}{P\left[C_{i-1}^{(c)} \geq v_i \,\middle|\, X_c = k\right]} \quad (19)$$

where we applied the following proposition. If an event $A \subseteq C$, then $P[A \mid B \cap C]$ equals $P[A \mid B]/P[C \mid B]$. In this case, A is equal to $C_{i-1}^{(c)} = s + v_i$ and C is chosen as $C_{i-1}^{(c)} \geq v_i$.

Discussion D: The expected number of slots at level i equals 2^{n-i+1} times the expected number of collisions at level $i-1$. Because the event $P_a = i$ is the same as $R_a \geq i \cap C_{i-1}^{(a)} < v_i$, we are actually looking for $E[C_{i-1}^{(a)} \mid X_a = k \cap R_a \geq i \cap C_{i-1}^{(a)} < v_i]$. We start with the following observation:

$$\begin{aligned} E\left[C_{i-1}^{(a)} \,\middle|\, X_a = k \cap R_a \geq i\right] &= P\left[C_{i-1}^{(a)} \geq v_i \,\middle|\, X_a = k \cap R_a \geq i\right] \\ &\quad \cdot E\left[C_{i-1}^{(a)} \,\middle|\, X_a = k \cap R_a \geq i \cap C_{i-1}^{(a)} \geq v_i\right] \\ &\quad + P\left[C_{i-1}^{(a)} < v_i \,\middle|\, X_a = k \cap R_a \geq i\right] \\ &\quad \cdot E\left[C_{i-1}^{(a)} \,\middle|\, X_a = k \cap R_a \geq i \cap C_{i-1}^{(a)} < v_i\right] \end{aligned} \quad (20)$$

where the expression of interest is part of the right-hand side. Both probabilities are clearly each others' complement. Because the event $C_{i-1}^{(a)} \geq v_i$ is a part of the event $R_a \geq i$ $(v_i > 0)$, the first probability equals $P[C_{i-1}^{(a)} \geq v_i \mid X_a = k]/P[R_a \geq i \mid X_a = k]$. Both these values were obtained in Discussion A.

The first expected value in the right-hand side of (20) can be reduced to (19) by noticing that the event $C_{i-1}^{(a)} \geq v_i$ is a part of the event $R_a \geq i$ and $C_{i-1}^{(a)} \geq v_i$ cioncides with $C_{i-1}^{(c)} \geq v_i$ when conditioned on $X_a = k$, respectively, $X_c = k$. In Discussion E, we calculate the left-hand side of (20), which concludes the throughput analysis.

Discussion E: Remark that the event $R_a \geq i$ coincides with $C_{i-1}^{(a)} > 0$. As the event $C_{i-2}^{(a)} \geq v_{i-1}$ contains this last event, we can also write it as $C_{i-2}^{(a)} \geq v_{i-1} \cap C_{i-1}^{(a)} > 0$. So, we want to find $E[C_{i-1}^{(a)} \mid X_a = k \cap C_{i-2}^{(a)} \geq v_{i-1} \cap C_{i-1}^{(a)} > 0]$. Some straightforward reasoning based on the definition of the expected value yields

$$E\left[C_{i-1}^{(a)} \,\middle|\, X_a = k \cap C_{i-2}^{(a)} \geq v_{i-1} \cap C_{i-1}^{(a)} > 0\right] = \frac{E\left[C_{i-1}^{(a)} \,\middle|\, X_a = k \cap C_{i-2}^{(a)} \geq v_{i-1}\right]}{1 - P\left[C_{i-1}^{(a)} = 0 \,\middle|\, X_a = k \cap C_{i-2}^{(a)} \geq v_{i-1}\right]}. \quad (21)$$

Applying $P[A \mid B \cap C] = P[A \cap C \mid B]/P[C \mid B]$, we can rewrite the probability in the denominator as

$$P\left[C_{i-1}^{(a)} = 0 \,\middle|\, X_a = k \cap C_{i-2}^{(a)} \geq v_{i-1}\right] = \frac{P\left[C_{i-1}^{(a)} = 0 \cap C_{i-2}^{(a)} \geq v_{i-1} \,\middle|\, X_a = k\right]}{P\left[C_{i-2}^{(a)} \geq v_{i-1} \,\middle|\, X_a = k\right]}. \quad (22)$$

Due to Discussion A1, we can substitute the super- and subscripts a for c in both probabilities without altering their values. Having done this, the numerator, respectively, denominator, can be calculated using similar argument as in Discussion A2, respectively, A1. We end with the determination of the expected value in the right-hand side of (21). Again, we can substitute the sub- and superscripts a for c. Then, using the definition of the expected value, we get

$$\begin{aligned} E\left[C_{i-1}^{(c)} \,\middle|\, X_c = k \cap C_{i-2}^{(c)} \geq v_{i-1}\right] &= \sum_{l \geq v_{i-1}} E\left[C_{i-1}^{(c)} \,\middle|\, X_c = k \cap C_{i-2}^{(c)} = l\right] \\ &\quad \cdot \frac{P\left[C_{i-2}^{(c)} = l \,\middle|\, X_c = k\right]}{P\left[C_{i-2}^{(c)} \geq v_{i-1} \,\middle|\, X_c = k\right]}. \end{aligned} \quad (23)$$

We calculate the numerator of this sum using the same methodology as in (5) and (6), where we define $e(i-1, s, k)$ as $E[2^{i-1} - C_{i-1}^{(c)} \mid X_c = k \cap C_{i-2}^{(c)} = 2^{i-2} - s]P[C_{i-2}^{(c)} = 2^{i-2} - s \mid X_c = k]$. This results in the following equations:

$$e(i-1, 2^{i-2}, k) = 2^{i-1} \frac{2^{(n-i+2)k} C_k^{2^{i-2}}}{C_k^{2^n}} \quad (24)$$

$$\begin{aligned} e(i-1, s, k) &= C_s^{2^{i-2}} \sum_{l_1=0}^{s} 2^{(n-i+2)l_1} \\ &\quad \cdot \left(2s + (2^{i-1} - 2s) \frac{C_{k-l_1}^{m_i} + 2^{n-i+1} C_{k-l_1-1}^{m_i}}{C_{k-l_1}^{2^n - s2^{n-i+2}}}\right) \\ &\quad \cdot \frac{C_{l_1}^s C_{k-l_1}^{2^n - s2^{n-i+2}}}{C_k^{2^n}} - \sum_{x=1}^{2^{i-2}-s} C_s^{s+x} e(i-1, s+x, k) \end{aligned} \quad (25)$$

with m_i equal to $2^n - s2^{n-i+2} - 2^{n-i+1}$.

D. Delay and Throughput when the First Levels are Skipped (STATIC)

In this subsection we describe the necessary adaptations to the evaluation methods above such that it allows us to evaluate ISAP when some of the first levels of the tree are skipped. The starting level is denoted by S_l. The following random variables are defined: X_a^+ is the number of participants in the CC, this variable ranges from 0 to 2^n. R_a^+ is the level at which the CC is resolved, this variable ranges from S_l to n. $S_i^{(a+)}$ equals the number of slots used at level i. P_a^+ denotes the level at which we poll, if the CC is solved without polling, P_a^+ obtains the value $n+1$.

1) The Delay when the First Levels are Skipped (STATIC): In this subsection we will address the most significant differences with the previous evaluation (of ISAP). The two major differences regarding the behavior of the protocol are that a CC can no longer be solved before level S_l as these levels no longer exist, and polling at level S_l is no longer possible as level $S_l - 1$ is skipped.

Discussion F: Let us start with R_a^+. Notice that if the CC was resolved at the first level S_l, then it is also solved at or before level S_l with the ISA scheme and vice versa. Second, the events $R_a^+ \leq x$ and $R_a \leq x$ coincide if $x > S_l$. Thus, we have

$$P[R_a^+ = S_l \,|\, X_a^+ = k] = \frac{2^{(n-S_l)k} C_k^{2^{S_l}}}{C_k^{2^n}}$$
$$P[R_a^+ \leq S_l + x \,|\, X_a^+ = k] = P[R_a \leq S_l + x \,|\, X_a = k]$$

for every value $x > 0$. This means that the probability of resolving a CC before or at level S_l might decrease a bit, compared to the ISAP scheme that starts at level zero. If so, the probability that it is solved at level $S_l + 1$ increases together with the probability of polling at this level. Remark that the probability of solving the scheme at level $S_l + 1$ without polling remains the same. For the other levels, everything remains exactly the same as before. Next, define $p_a^+(k,i)$ as $P[R_a^+ = S_l + i - 1 \,|\, X_a^+ = k]$.

$\mathcal{F}_a^+(i,k)$ denotes the probability that a tagged request is successful at or before level i. With similar arguments as used for R_a^+, we get $\mathcal{F}_a^+(S_l, k) = C_{k-1}^{2^n - 2^{n-S_l}} / C_{k-1}^{2^n - 1}$ and $\mathcal{F}_a^+(S_l + x, k) = \mathcal{F}_a(S_l + x, k)$ for every positive value of x. The remainder of the analysis is analogous to the one with starting level zero.

2) The Throughput when the First Levels are Skipped (STATIC): The ISAP scheme will never poll at level S_l (or before, since these levels do not exist); thus, the probability of polling at level $S_l + 1$ increases. This causes the expected number of slots at level S_l and $S_l + 1$ to be different from the ones we had before (Subsection C-2). The other expected values remain the same. The expected number of slots at level S_l matches 2^{S_l} because we start at this level. For level $S_l + 1$, we start by looking at (18) (where we add a $+$ to all random variables and set i equal to $S_l + 1$). In view of Discussion F, adding a $+$ only changes the first two values (of the right-hand side) in this expression. Based on the fact that the events at level S_l are similar to those of the ISA scheme, and with the starting level at zero, the product of these two values is given by

$$\begin{aligned} &P[P_a^+ = S_l + 1 \,|\, X_a^+ = k] \\ &\quad \cdot E\left[S_{S_l+1}^{(a+)} \,\middle|\, X_a^+ = k \cap P_a^+ = S_l + 1\right] \\ &= 2^{n-S_l} \sum_{i=1}^{u_{S_l}} i P\left[C_{S_l}^{(c)} = i \,\middle|\, X_c = k\right]. \end{aligned} \tag{26}$$

E. Delay and Throughput Analysis when the First Levels are Skipped (DYNAMIC)

In this subsection, the analysis for the static starting level is extended to the proposed dynamic model. We use the same random variables as above, but substitute the $+$ sign for a $*$ to indicate the dynamic nature of the scheme. We also introduce a new random variable B^* as the starting level.

1) The Delay when the First Levels are Skipped (DYNAMIC): We start by studying R_a^* when conditioned on X_a^* and B^*. Assuming that the starting level equals S_l, we have the following (due to the STATIC part):

$$\begin{aligned} &P[R_a^* = S_l \,|\, X_a^* = k \cap B^* = S_l] \\ &\quad = \frac{2^{(n-S_l)k} C_k^{2^{S_l}}}{C_k^{2^n}} \end{aligned} \tag{27}$$

$$\begin{aligned} &P[R_a^* \leq S_l + x \,|\, X_a^* = k \cap B^* = S_l] \\ &\quad = P[R_a \leq S_l + x \,|\, X_a = k] \end{aligned} \tag{28}$$

with x a positive number. We denote $p_a^*(k, S_l, x+1)$ as $P[R_a^* = S_l + x \,|\, X_a^* = k \cap B^* = S_l]$. In order to find the joint distribution of (X_a^*, B^*), it is sufficient to construct the following transition matrix and do the matrix inversion as shown in the equation at the bottom of the page. When we observe the system at an arrival instant, then the probability that there are k contenders in the current CC and that this CC started at level S_l is needed. We also need this probability for the next CC (after the one containing the random arrival). These values are a natural extensions of Discussion B

$$\begin{aligned} &P\left[X_{next}^{(a*)} = k \cap B_{next}^* = S_l\right] \\ &\quad = \frac{P[X_a^* = k \cap B^* = S_l] k}{E[X_a^*]} \end{aligned} \tag{29}$$

and

$$\begin{aligned} &c_a^*(k, S_l) \\ &\quad \triangleq P\left[X_{cur}^{(a*)} = k \cap B_{cur}^* = S_l\right] \\ &\quad = \sum_{j=1}^{2^n} \sum_{S_n = S_{\min}}^{S_{\max}} \frac{P[X_a^* = k \cap B^* = S_l] t_a^*(k; S_l, j; S_n) j}{E[X_a^*]}. \end{aligned} \tag{30}$$

Finally, we need to find $\mathcal{F}_a^*(i, k, S_l)$, being the probability that a tagged arrival is successful at or before level i, knowing that there were $k - 1$ other participants, and the CC started at level S_l. Using the results of the previous section (see STATIC), we find that $\mathcal{F}_a^*(S_l, k, S_l) = C_{k-1}^{2^n - 2^{n-S_l}} / C_{k-1}^{2^n - 1}$ and $\mathcal{F}_a^*(S_l + x, k, S_l) = \mathcal{F}_a(S_l + x, k)$ for $x > 0$. These results can be combined to obtain the average delay of the system. Let us now focus on the delay density function. As in (14), s is the number

---

$$\begin{aligned} t_a^*(k; S_b, j; S_a) &= P\left[X_{n+1}^{(a*)} = j \cap B_{n+1}^* = S_a \,\middle|\, X_n^{(a*)} = k \cap B_n^* = S_b\right] \\ &= \sum_{t=1}^{n+1-S_b} \frac{(\lambda t)^j e^{-\lambda t}}{j!} p_a^*(k, S_b, t) 1_{\{(t \leq B_l \wedge S_a = S_b - 1) \vee (B_l < t < B_m \wedge S_a = S_b) \vee (t \geq B_m \wedge S_a = S_b + 1)\}} \end{aligned}$$

of frames that the tagged element competes, j is the length of the CC (in frames) in which the tagged request was generated, and S_b the level at which this CC started. While $l-1$ is the number of other competitors next to the tagged element,

$$D_c(x) = \sum_{s=1}^{\lfloor x \rfloor} \sum_{S_b=S_{\min}}^{S_{\max}} \sum_{j=\lceil x \rceil - s}^{n+1} \sum_{l=1}^{2^n} \cdot \frac{\mathbf{\Delta}_1 \mathcal{F}_a^*(f(S_b,j)+s-1, l, f(S_b,j))}{j} \mathcal{G}_j(l) \cdot \left(\sum_{k=0}^{2^n} \frac{jP[R_a^* = S_b + j - 1 | X_a^* = k \cap B^* = S_b]}{1 + E[R_a^* - S_b | X_a^* = k \cap B^* = S_b]} c_a^*(k, S_b) \right) \tag{31}$$

where the function $f(S_b, j)$ is given by

$$f(S_b, j) = \begin{cases} \max(S_{\min}, S_b - 1) & j \le B_l \\ S_b & B_l < j < B_m \\ \min(S_{\max}, S_b + 1) & j \ge B_m \end{cases} \tag{32}$$

and $\mathbf{\Delta}_1 \mathcal{F}_a^*(x,y,z)$ equals $\mathcal{F}_a^*(x,y,z) - \mathcal{F}_a^*(x-1,y,z)$.

2) The Throughput when the First Levels are Skipped (DYNAMIC): In the previous section we obtained the joint distribution of (X_a^*, B^*). This is used to derive an expression for the throughput T_a^*, as shown in (33) at the bottom of the page, where the expected values were obtained in the evaluation of the static model.

IV. Numerical Results

In this section, we use the analytical model to investigate the impact of the arrival rate λ, the trigger value N_p, and the starting level S_l on the mean delay, the delay density function, and the throughput. The system parameters are set as follows. The number of mobiles is 128, hence $n = 7$. The arrival rate λ (requests per frame) varies between 0.05 and 3.5. The three values studied for the polling threshold N_p are 0, 20, and 40, where the first case corresponds with the ISA scheme. The starting level S_l will vary from level 0 to 2. When studying a system with a dynamic starting level, B_l and B_m are set to 1 and 4, respectively. Therefore, the level is decreased by one if the CC is solved in one frame; and is increased by one if the CC consists of four or more frames. The boundary values are set as follows: $S_{\min} = 0$ and $S_{\max} = 2$. The number of instances varies between 1 and 4.

We study four different scenarios. First, we investigate the impact of the polling threshold N_p. In that case, the starting level is fixed at 0. Next the influence of the starting level S_l is discussed. Then the impact of the use of a dynamic scheme for the starting level is considered. Finally, we look at the effect of using multiple instances of ISA. Additional numerical results can be found in [17].

A. The Influence of the Polling Threshold on the System Performance

Figs. 5 and 6 show the influence switching to polling has on the mean delay and the throughput. As expected, we get a tradeoff between the delay and throughput characteristics: the sooner the ISAP protocol switches to polling, the shorter the mean delay, but the lower the throughput.

From the figures, we observe that the protocol behaves very similar for different N_p values when the arrival rate λ is small (below 0.25). A similar result is obtained for large values of λ (beyond 5). Both these results are intuitively clear. Polling is not an issue in these cases: for λ very small, collisions rarely occur, and hence ISAP solves the collisions; if λ is very large, the remaining size of the address space is too large to switch to polling.

Let us now consider moderate values for λ. Recall that for the polling threshold $N_p = 40$, respectively, $N_p = 20$, the protocol will never start polling until level 3, respectively, level 4. Thus, the impact of N_p on the performance measures is low for small values of λ. If the arrival rate increases (look at the range 0.5 till 1), the probability that collisions at level 2, respectively, 3 are introduced increases. In most cases, these collisions contain very few participants, and hence occasionally 32, respectively, 16 polling slots are provided at level 3, respectively, 4 to poll very few competitors. Therefore, the throughput decreases with increasing value of N_p. If λ is increased even more, beyond one, polling is postponed in most cases to a later level (as the expected number of collisions at level 2, respectively, 3 becomes larger than one) and will contain more participants. This results in higher throughput values for a fixed value of N_p.

Fig. 7 shows the impact of polling on the delay distribution function (for $\lambda = 1$). It illustrates that the main improvement of the delay is located within the tail of the distribution.

B. The Influence of Skipping Levels (STATIC) on the System Performance

Figs. 9 and 10 illustrate the impact of S_l on the average delay and the throughput. In these figures we have three different types of curves—full, dotted, and dashed—corresponding to $S_l = 0$, 1, and 2, respectively. Moreover, for each value of S_l, the results for $N_p = 0$, 20, and 40 are depicted. For a fixed value of S_l, the upper curve corresponds to $N_p = 0$, the middle curve to $N_p = 20$, and the lowest to $N_p = 40$.

$$T_a^* = \frac{E[X_a^*]}{\sum_{k=0}^{2^n} \sum_{S_l=S_{\min}}^{S_{\max}} P[X_a^* = k \cap B^* = S_l] E\left[\sum_i S_i^{(a*)} \middle| X_a^* = k \cap B^* = S_l \right]} \tag{33}$$

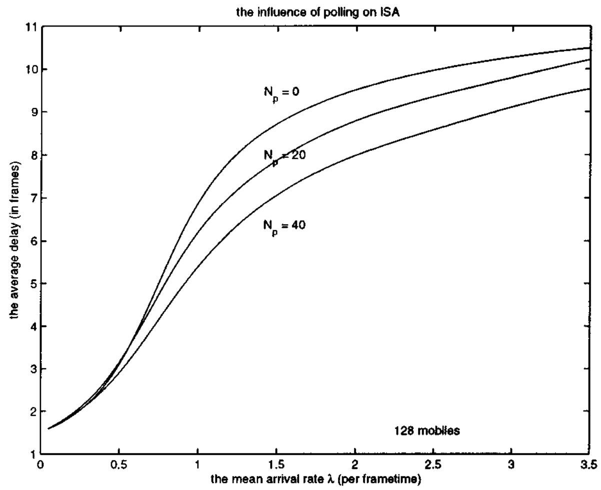


Fig. 5. The impact of polling on the mean delay.

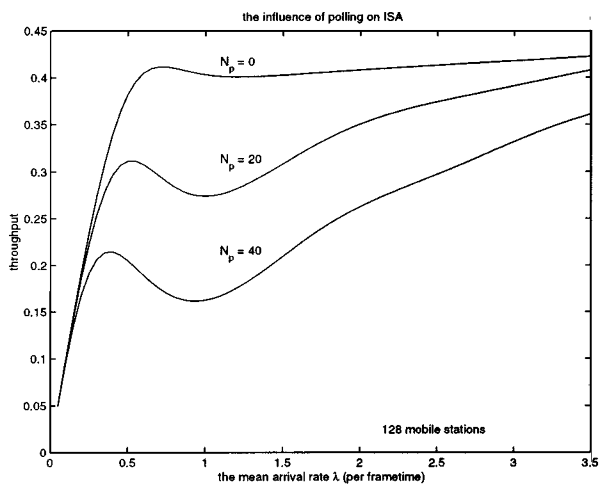


Fig. 6. The impact of polling on the throughput.

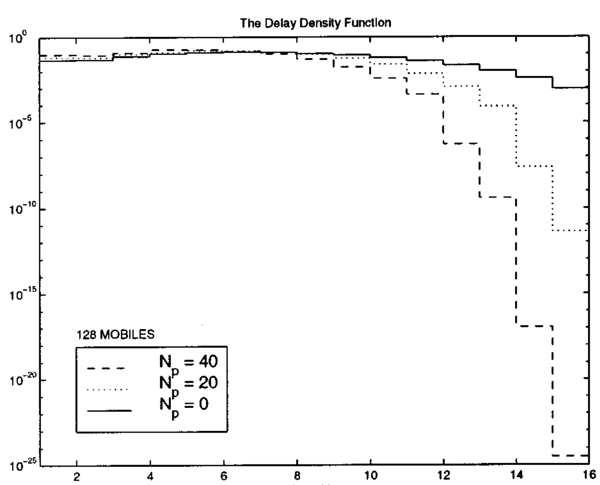


Fig. 7. The impact of polling on the delay density function with $\lambda = 1$.

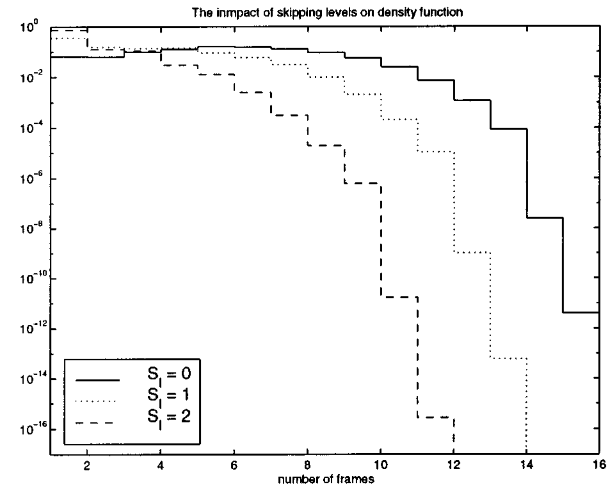


Fig. 8. The impact of skipping with $\lambda = 1$ and $N_p = 20$.

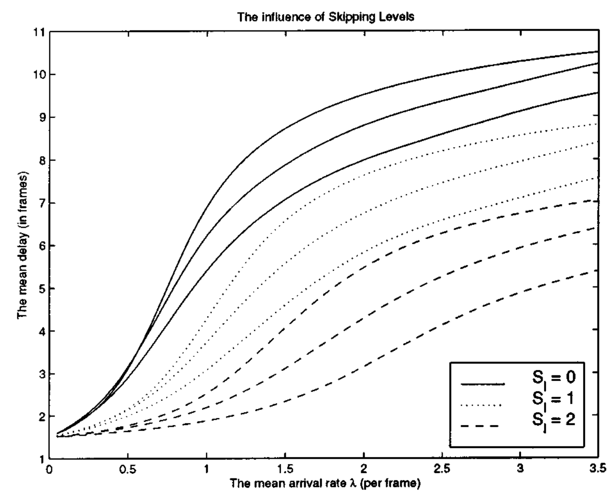


Fig. 9. The influence of skipping on the mean delay.

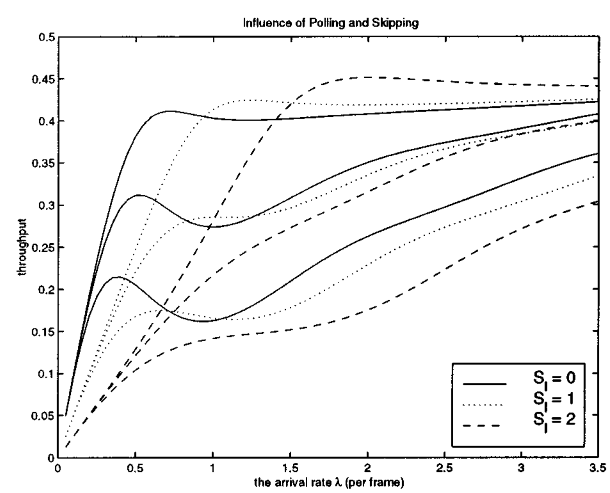


Fig. 10. The influence on the throughout for $N_p = 0$, 20, and 40.

Skipping the first levels leads to a decrease of the mean waiting time. Let us now focus on the impact of polling for variable values of S_l. First, Fig. 9 shows a faster decrease of the delay due to polling, when the starting level is larger. This can be seen by observing the area between the curves for $N_p = 0$, 20 and $N_p = 40$. Second, notice that the curves converge slower for increasing value of S_l (observe the differences for $\lambda = 3.5$ in Fig. 9). Fig. 10 represents the throughput results for $N_p = 0$, 20, and 40. We see that for low values of λ, skipping levels results in a lower throughput (as most of the 2^{S_l} slots are

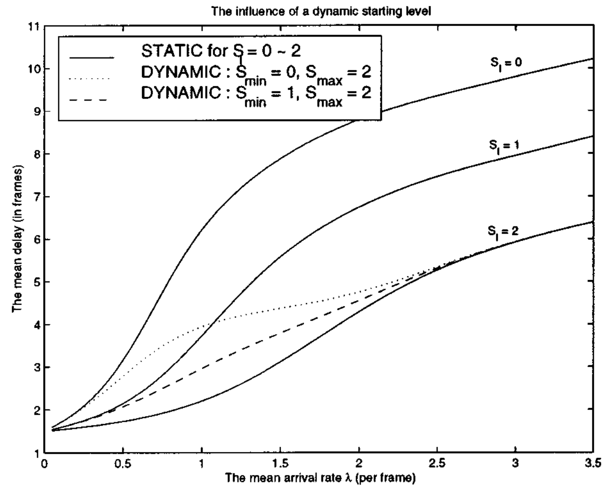


Fig. 11. The influence of dynamic skipping on the delay $(N_p = 20)$.

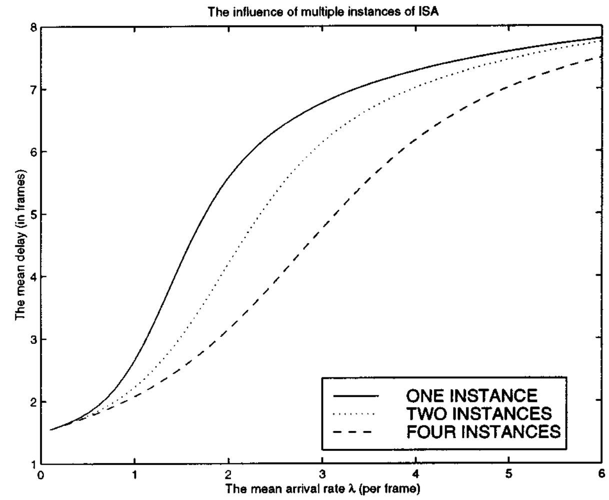


Fig. 13. The influence of multiple instances on the delay.

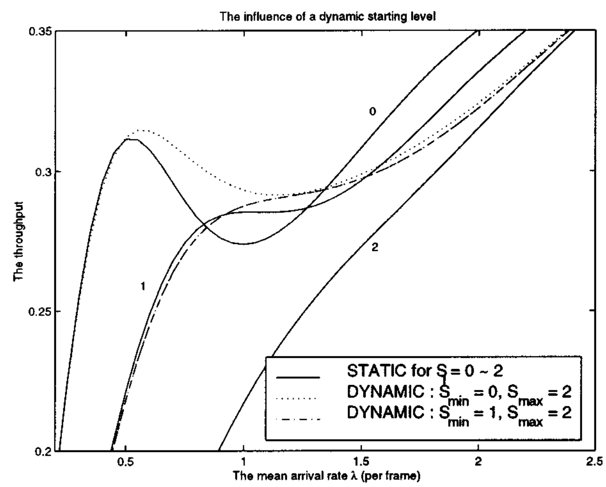


Fig. 12. The influence of dynamic skipping on the throughput for $N_p = 20$.

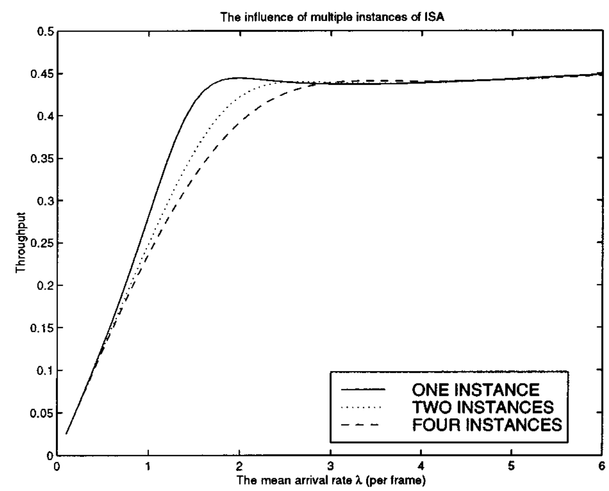


Fig. 14. The influence of multiple instances on the throughput.

wasted). If λ becomes larger, this loss is converted in a small gain, due to the fact that the majority of the slots before level S_l contains collisions. The influence of skipping levels on the delay density function is shown in Fig. 9.

C. The Influence of Skipping Levels (DYNAMIC) on the System Performance

From the previous sections we may conclude that a higher starting level has a positive impact on the delay and even on the throughput, especially for larger values of λ. Unfortunately, a high price is paid for this in terms of throughput if λ is small. The aim of this section is to show that the dynamic scheme as proposed in Section II-D solves this problem. That is, if λ is small, the result should tend to the results for $S_l = S_{\min}$; while for λ large, the behavior should be similar to the one corresponding to $S_l = S_{\max}$.

Figs. 11 and 12 show that this is the case (for $N_p = 20$), meaning that a system where the levels are skipped dynamically is able to limit the maximum delay while keeping the throughput high.

D. The Influence of Multiple Instances of ISA on the System Performance

Notice that we can apply the analysis presented in Section III in order to evaluate the influence of multiple instances. In this final scenario, λ varies between 0 and 6. Figs. 13 and 14 show the delay and throughput results for three configurations. In the first, we have one instance and the starting level S_l is fixed at 2. In the second, we have two instances, with $S_l = 1$. Finally, we have four instances, with $S_l = 0$.

Clearly, the more instances we use, the better the average delay; except for very small and very large values of λ, where all scenarios perform alike. For more moderate values of λ, there exists a tradeoff between the delay and throughput; thus, the more instances we use, the smaller the delay and the lower the throughput is. Still, this time, the decrease in throughput is considerably smaller, thereby making use of multiple instances attractive.

V. Conclusions

In this paper, the performance analysis of the identifier splitting algorithm combined with polling (ISAP), when employed on an uplink contention channel used in a wireless access network, was presented. Both the delay density function and the throughput characteristics were obtained analytically. Multiple numerical results have shown that the identifier splitting algorithm combined with polling (ISAP), enhanced by a mechanism of skipping in a dynamic way the first levels, leads to a good tradeoff between low delay and high throughput results. Finally, the use of multiple instances was shown to have a good impact on the delay, with little throughput losses.

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[2] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. 25, pp. 319–329, 1979.

[3] C. S. Chang, K. C. Cheng, M. Y. You, and J. F. Chang, "Guaranteed quality-of-service wireless access control for wireless ATM environments," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 106–118, Jan. 1997.

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

[5] M. J. Karol, Z. Liu, and K. Y. Eng, "Distributed-queueing request update multiple access (DQRUMA) for wireless packet (ATM) networks," *Wireless Networks*, vol. 1, no. III, 1995.

[6] J. Mikkonen, J. Aldis, G. Awater, A. Lunn, and D. Hutchison, "The MAGIC WAND-functional overview," *IEEE J. Select. Areas Commun.*, vol. 16, Aug. 1998.

[7] N. Passas, S. Paskalis, D. Vali, and L. Merakos, "Quality-of-service-oriented medium access control for wireless ATM networks," *IEEE Commun. Mag.*, Nov. 1997.

[8] D. Petras, "Medium access control protocol for wireless, transparent ATM access in MBS," presented at the RACE Mobile Telecommun. Summit, Cascais, Portugal, 1995.

[9] D. Petras and A. Kramling, "MAC protocol with fast collision resolution for an ATM air interface," presented at the IEEE ATM Workshop, San Francisco, Aug. 1996.

[10] D. Petras, A. Kramling, and A. Hettich, "MAC protocol for wireless ATM: Contention free versus contention based transmission of reservation requests," presented at the 7th IEEE PIMRC, 1996.

[11] D. Petras and A. Kramling, "Fast collision resolution in wireless ATM networks," presented at the 2nd MATHCOM, Vienna, Austria, Feb. 1997.

[12] J. Porter and A. Hopper, "An ATM based protocol for wireless LANs," ORL Tech. Rep., 1994.

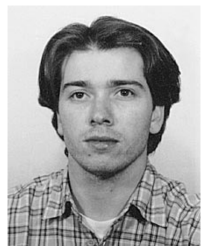
[13] F. D. Priscolli, "Medium access control protocol for the median systems," presented at the ACTS Mobile Summit, Granada, Nov. 1996.

[14] P. F. M. Smulders and C. Blondia, "Application of the asynchronous transfer mode in indoor radio networks," in *Proc. Joint PIMRC WCN Conf. 94*, The Hague, The Netherlands, 1994, p. 839.

[15] B. Tsybakov and N. Vvedenskaya, "Random multiple access stack algorithms," *Prob. Inform. Trans.*, vol. 16, pp. 230–243, July–Sept. 1980.

[16] B. Van Houdt and C. Blondia, "Performance evaluation of the identifier splitting algorithm with polling in wireless ATM networks," *Int. J. Wireless Inform. Networks*, vol. 7, no. 2, pp. 91–103, 2000.

[17] B. Van Houdt, C. Blondia, O. Casals, J. Garcia, and D. Vazquez-Cortizo, "A MAC protocol for wireless ATM systems: Supporting the ATM service categories," in *Proc. 16th Int. Teletraffic Conf. (ITC)*, 1999.



Benny Van Houdt graduated from the University of Antwerp in 1997 in mathematics and computer science. Currently he holds an Assistant position at the University of Antwerp (UIA) and is an active member of the Performance Analysis of Telecommunication Systems (PATS) research group. His main research interest goes to the performance evaluation and stochastic modeling of medium access control protocols for wired and wireless communication networks.

Chris Blondia (M'98) obtained the Master degree in science and the Ph.D. degree in mathematics, both from the University of Ghent (Belgium) in 1977 and 1982, respectively. In 1983 he joined Philips Belgium, where he was a researcher between 1986 and 1991 in the Philips Research Laboratory Belgium in the group Computer and Communication Systems. Between August 1991 and end 1994 he was an Associate Professor at the Computer Science Department of the University of Nijmegen (The Netherlands). In 1995 he joined the Department of Mathematics and Computer Science of the University of Antwerp, where is currently a Professor and head of the research group "Performance Analysis of Telecommunication Systems." He is lecturing Telecommunication and Performance Evaluation courses. His main research interests are related to mathematical models (stochastic modeling, queueing theory) for performance evaluation of communication systems and the impact of the performance on the architecture of these systems. In this context the areas of research are performance issues related to traffic modeling, traffic control functions, access control protocols. He has published a substantial number of papers in international journals and conferences on these research areas. He has been member of many program committees of international conferences and is editor of the Journal of Network and Computer Applications. He is member of IFIP W.G. 6.3 on "Performance of Computer Networks."