ARTICLE IN PRESS
Computer Networks xxx (2013) xxx–xxx
Contents lists available at ScienceDirect
Computer Networks
journal homepage: www.elsevier.com/locate/comnet

A branching process approach to compute the delay and energy efficiency of tree algorithms with free access

R. Block [*,1], G.T. Peeters, B. Van Houdt [*]

University of Antwerp – iMinds, Dept. Mathematics and Computer Science, Middelheimlaan 1, B-2020 Antwerp, Belgium

ARTICLE INFO

Article history:
Received 8 February 2013
Received in revised form 1 July 2013
Accepted 14 August 2013
Available online xxxx

Keywords:
Conflict resolution
Tree algorithms
Delay
Energy usage
Branching process

ABSTRACT

This paper presents a branching process approach to determine the main performance measures of a variety of conflict resolution algorithms known as tree algorithms with free access. In particular we present an efficient approach to calculate the mean delay, number of transmission attempts, collision resolution interval length and energy usage with arbitrary precision.

1. Introduction

Tree algorithms (TAs) form a well studied class of conflict resolution algorithms [1–7]. Depending on the manner in which new users join the channel, they are termed *free* or *blocked* access TAs. Under free access users transmit new packets at the start of the next time slot, meaning no channel sensing is required. Under blocked access some rules are in place that indicate when a new user is allowed to transmit a packet for the first time, which requires either limited or full sensing of the channel. It is fair to state that free access algorithms are typically harder to analyze than their blocked access counterparts, which is one of the reasons why fewer results on free access algorithms have appeared in the literature.

In this paper we extend the branching process approach of [8], which was used to study the maximum stable throughput (MST) of the same class of tree algorithms as considered in this paper, to analyze the mean delay, mean number of transmission attempts, mean length of the conflict resolution interval (CRI) and mean energy usage. As such the current paper heavily relies on the technique developed in [8] and the derivation of the mean length of the CRI is a rather trivial extension. However, extending the approach in [8] to determine the mean delay, mean number of transmission attempts and mean energy usage is far less obvious as one needs to determine, amongst others, the distribution of the number of packets that are transmitted during a packet's first transmission attempt (called the top-of-stack observed states in [9]). Once this distribution is obtained, it is not hard to adapt the branching process of [8] to determine the mean number of transmission attempts, but adapting it to determine the mean delay (and energy usage) from this distribution is still less obvious.

Existing results for the mean delay or energy characteristics [9–11] are typically expressed through some operator $S(f(\cdot),z)$ (see Section 4.6 for details), the numerical evaluation of which is very time and memory consuming and can often only be used to compute the first 4 or 5 digits accurately. Our branching approach on the other hand requires hardly any time (a fraction of a second) or memory and is able to produce results with very high accuracy (15 digits or more).

* Corresponding authors. Tel.: +32 2653882 (R. Block).
E-mail addresses: robbe.block@ua.ac.be (R. Block), benny.vanhoudt@ua.ac.be (B. Van Houdt).

[1] R. Block is a PhD-fellow of the Research Foundation – Flanders (FWO).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.08.022

One of the key characteristics of the branching process approach in [8] is the use of the truncation parameter d. Numerical experiments showed that the MST could be determined with a precision of as many as 15 digits even for moderate values of d, e.g., $d \leqslant 20$. The branching process approach introduced in this paper also relies on the truncation parameter d and we show that the approach can be used to calculate the performance measures of interest up to arbitrary precision by comparing them to the results presented in [9–12]. The method of [13,14] cannot be used to study the impact of d as it relies on a similar truncation. Apart from reproducing the existing results of [9,11,12], we also obtain many new results for other TAs with free access. All of these novel results were confirmed by means of discrete event simulations.

We consider the following standard channel model and user behavior (see [3–5] for a detailed discussion):

1. The channel is divided into fixed length time slots and all packets have the same length as a slot. Users are only allowed to start transmitting at the beginning of a time slot.
2. There is an infinite set of users whose aggregate packet generation process is a Poisson process with rate $\lambda > 0$.
3. Whether a transmission is successful only depends on the number of packets sent during the particular slot. Packet reception fails whenever two or more packets are transmitted simultaneously.
4. At the end of each time slot, the receiver sends either binary (collision/no collision) or ternary (idle/success/collision) feedback to the users.

While analyzing the TAs considered in this paper we often relax some of the four assumptions above, for instance we will consider channels with errors, multiple reception capabilities and probabilistic capture. Although all the presented numerical results assume Poisson arrivals, the branching process approach presented in this paper can be readily extended to any arrival process in which the number of arrivals in consecutive time slots are independent and identically distributed.

The paper is structured as follows. We start by discussing some related work in Section 2. Next, in Section 3 we briefly revisit the operation of the basic q-ary TA, the definition of a multi-type branching process and the manner in which these processes were used in [8] to determine the MST. Our branching process approach to compute the mean delay, mean number of transmission attempts, mean CRI duration and mean energy usage of the basic q-ary TA is discussed in detail in Section 4, while Section 5 indicates that the same general approach can be used to analyze a large variety of free access TAs without much additional effort. All the results in Sections 4 and 5 have either been validated by existing results or simulation. Concluding remarks are given in Section 6.

2. Related work

The maximum stable throughput (MST) under free access was obtained for the basic and modified q-ary TA in [15,2], for a channel with errors in [16,17], for variable length packets in [10], while [12] analyzed the impact of having some control sub-channels with separate feedback and [18] considered a system in which an interference cancellation mechanism is deployed. For other performance measures such as the delay or energy usage even fewer analytical results are available.

The delay of the basic binary TA algorithm with free access is analyzed in [9] using functional equations. The solution of these equations is expressed through some operator $S(f(\cdot),z)$ which is defined as a sum over a semigroup H. Numerically evaluating this operator (for $p \neq 1/2$) is computationally heavy and considerable care is needed even when computing the first 5 digits only. A similar approach as in [9] was used in [10] to obtain the delay characteristics of a TA with variable length packets. The mean energy usage of the basic binary TA with free access was determined in [11] and was defined based on the mean delay $E[D]$ and mean number of transmission attempts $E[T]$, where $E[T]$ was expressed using the same operator $S(f(\cdot),z)$. A computational method based on matrix analytic methods to calculate the mean delay of the basic q-ary TA was also presented in [13,14]. Finally, [12] provided a closed form expression for the mean delay in case of an infinite number of control sub-channels, which corresponds to the coordinated splitting algorithm in [8].

More recently, a novel technique to determine the MST of TAs with free access was presented in [8]. Given the arrival rate λ and a particular TA with free access, the technique existed in defining a branching process such that the process is sub-critical if and only if the TA is stable under Poisson arrivals with rate λ. The MST of various TAs with free access could therefore be determined in an efficient manner by means of a simple bisection algorithm.

3. Preliminaries

Before introducing our branching process approach in Section 4, we briefly discuss the operation of the basic q-ary algorithm, the definition of a multi-type branching process and the branching process used in [8] to determine the MST of the basic q-ary TA.

The basic q-ary TA with free access, where $q \geqslant 2$ is an integer, operates in the following manner. Whenever a user has a packet ready for transmission, he becomes active. All active users maintain a single variable called the *counter*, the value of which is updated at the end of each time slot. A user is allowed to transmit in the next slot whenever his counter equals zero. Users that become active initialize their counter to zero and are therefore allowed to transmit in the next time slot. Binary feedback is provided at the end of each time slot and the counter of a user is updated as follows:

1. If the slot holds a collision, all users with a counter larger than zero increase their counter by $q-1$. Each user involved in the collision sets his counter equal to i, with probability p_{i+1}, for $i = 0, \ldots, q-1$. The probabilities $p_1, \ldots, p_q$ are protocol parameters that sum to one.

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

2. If the slot does not hold a collision, all users with a counter larger than zero decrease their counter by 1. If the slot held a successful transmission, the successful users becomes inactive.

Hence, the basic q-ary TA is a collision resolution algorithm in which each user involved in a collision joins group G_i ($i \in \{1,q\}$) with probability p_i. Users belonging to group G_1 retransmits in the next slot, the group G_i users refrain from retransmitting their packets until group G_1 to G_{i-1} are resolved, for $i = 2, \ldots, q$. Whenever a collision occurs during one of the retransmissions, the same procedure is applied recursively. Note, any new users that join while group G_i is being resolved also need to transmit their packet successfully before group G_{i+1} can retransmit. The counter that is maintained by each user basically reflects the number of groups that still need to be resolved before the user can retransmit, because a collision creates $q-1$ additional groups and an idle/successful slot indicates that one (possibly empty) group got resolved.

A multi-type (Galton–Watson) branching process (BP) [19] describes the evolution of a population of nodes of $d+1$ types, labeled type 0 to d, for some integer d. Each node is of some type $i \in \{0,\ldots,d\}$, belongs to some generation x and is a child of a generation $x-1$ node, unless $x = 0$. The process is completely characterized by a set of probabilities $\{p_{j,k_0,k_1,\ldots,k_d} | j = 0,\ldots,d; k_i \geqslant 0; i = 0,\ldots,d\}$, where $p_{j,k_0,k_1,\ldots,k_d}$ is the probability that a type j node has k_i children of type i. This probability is independent of the generation to which a node belongs and the number of children that any two nodes have is independent of each other. Many important properties of the evolution of the BP can be expressed solely based on the mean number of type j children that a type i node has, for $i, j = 0, \ldots, d$, which we denote as $m_{i,j}$. Therefore, there is no need to specify the probabilities $p_{j,k_0,k_1,\ldots,k_d}$.

In [8] it was shown that for any arrival rate λ, we can construct a multi-type BP such that the process is sub-critical if and only if the basic q-ary TA algorithm is stable under Poisson arrivals with rate λ. A multi-type BP is sub-critical if and only if the spectral radius $sp(M)$ of the expectation matrix M is less than one (provided that M is irreducible), with

$$M = \begin{bmatrix} m_{0,0} & m_{0,1} & \ldots & m_{0,d} \\ m_{1,0} & m_{1,1} & \ldots & m_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{d,0} & m_{d,1} & \ldots & m_{d,d} \end{bmatrix}.$$

The branching process in [8] was constructed as follows. A node in a multi-type BP was associated with each slot. The type of the node associated with a slot in which n packets are transmitted is n. Type 0 and 1 nodes have no children, while a type $n \geqslant 2$ node has exactly q children. These q children correspond to the q slots in which group G_1 to G_q are allowed to retransmit for the first time. Let g_i be the number of users that selected group G_i and ℓ_i the number of new arrivals that transmit in the slot that allows group G_i to retransmit. This results in a slot in which $n_i = g_i + \ell_i$ users will transmit and thus the ith child is of type n_i. Whenever n_i exceeded the truncation parameter d (e.g., $d = 20$), $n_i - d$ new arrivals were dropped and the node is defined as a type d node.

The size $d+1$ expectation matrix $M^{(q)}$ of this BP was derived as follows. We first define $B^{(q)}$ as the expectation matrix given that we ignore any new arrivals. Therefore entry (i,j) of $B^{(q)}$ holds the expected number of groups with j packets after a size i collision is split:

$$B^{(q)}_{i,j} = \begin{cases} \binom{i}{j} \sum_{k=1}^{q} p_k^j (1-p_k)^{i-j} & i \geqslant 2,\ i \geqslant j, \\ 0 & \text{otherwise}. \end{cases} \quad (1)$$

Indeed, the expected number of groups with j users that chose the kth group to retransmit equals $\binom{i}{j} p_k^j (1-p_k)^{i-j}$. Next, we take the new arrivals into account. Assume a slot of type j consists of $k \geqslant 0$ packets that require retransmission, then the number of new arrivals in that slot must equal $j-k \geqslant 0$. If we denote $b_k = \lambda^k e^{-\lambda}/k!$, entry (i,j) of $M^{(q)}$ equals $\sum_{k=0}^{j} B^{(q)}_{i,k} b_{j-k}$, for $j < d$. For $j = d$ we must have $d-k$ *or more* new arrivals. Defining E as

$$E = \begin{bmatrix} b_0 & b_1 & \cdots & \sum_{k=d}^{\infty} b_k \\ & b_0 & \cdots & \sum_{k=d-1}^{\infty} b_k \\ & & \ddots & \vdots \\ & & & \sum_{k=0}^{\infty} b_k = 1 \end{bmatrix},$$

$M^{(q)}$ can be written as $B^{(q)}E$, where E depends on the arrival rate λ.

4. Basic *q*-ary tree algorithm

In this section we indicate how the branching process approach of [8] can be extended such that we can also compute other performance measures such as the mean CRI duration, the mean delay and number of transmission attempts of the basic q-ary TA. While doing so, we will also make use of the truncation parameter d and will show that even moderate values of d, e.g., $d = 20$, suffice to get highly accurate results. In the next section we will apply the same approach to a large variety of other TAs with free access, for which there are currently hardly any results available in the literature.

4.1. Mean CRI duration

We start with the computation of the mean CRI duration, which is almost straightforward to obtain from the BP with expectation matrix $M^{(q)}$ described in Section 3. Any slot in which no (possibly empty) group G_i is allowed to retransmit is defined as the first slot of a CRI. A CRI ends when the next CRI starts. Hence, CRIs that do not start with a collision end after one slot, while otherwise a CRI lasts at least $q+1$ slots. The length of a CRI is defined as the number of slots part of the CRI, unless otherwise stated.

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

As every slot corresponds to a node in the BP and vice versa, the mean CRI duration of a CRI that starts with a collision of i users is equal to the expected total number of offspring generated by the BP with a single type i node in generation 0. The expected number of type j nodes in generation n in a multi-type BP that is initiated by a type i node, is found as entry (i,j) of the nth power of its expectation matrix [19]. Summing over all generations implies that entry (i,j) of

$$C^{(q)} = \sum_{n=0}^{\infty}(M^{(q)})^n = \left(I - M^{(q)}\right)^{-1} = \left(I - B^{(q)}E\right)^{-1},$$

holds the expected total number of type j offspring of a type i node. Note that $C^{(q)}$ is only properly defined if the population goes extinct with probability 1 (otherwise the sum does not converge), that is, if the algorithm is stable under rate λ.

To obtain the CRI length conditioned on the size of its initial slot, we can simply compute the row sums of $C^{(q)}$ for $0 \leqslant i \leqslant d$. The unconditional mean CRI duration is found by remarking that the first slot of a CRI contains i packets with probability b_i, due to the Poisson arrivals. Hence, the mean CRI duration $E[C]$ equals $E[C] = bC^{(q)}e$, where e is a column vector with all its entries equal to 1 and $b = (b_0, b_1, \ldots, b_{d-1}, \sum_{k=d}^{\infty} b_k)$.

4.2. Packet distribution

To determine the mean number of transmission attempts and the mean delay of an arbitrary packet, we need to determine the distribution of the number of other packets that are transmitted during a packet's first transmission attempt. This number is called the *top-of-stack observed states* in [9].

First we define e_1^T as a size $d + 1$ row vector the first entry of which equals 1 and the others equal 0 and I_{d+1} as the size $d + 1$ identity matrix. We also define E_x as

$$E_x = I_{d+1} \otimes b = \begin{bmatrix} b_0 & \cdots & \sum_{k=d}^{\infty} b_k & & & \\ & & & \cdots & & \\ & & & b_0 & \cdots & \sum_{k=d}^{\infty} b_k \end{bmatrix},$$

in which $\otimes$ denotes the Kronecker product of two matrices as defined in [20]. Using these matrices, and defining $M_x = B^{(q)}E_x$ we let

$$H^{(q)} = (e_1^T \otimes I) + \sum_{n=1}^{\infty}(B^{(q)}E)^{n-1}M_x = (e_1^T \otimes I) + C^{(q)}M_x. \quad (2)$$

When expanding M_x, we see that entry $(i,j + (d + 1)k)$ contains the expected number of type $j + k$ children of a type i node, where among the $j + k$ packets, j are new arrivals (first time transmissions) and k are retransmissions. Furthermore, it is not hard to see that $(B^{(q)}E)^{n-1}M_x$ contains the same information for generation n and thus $\sum_{n=1}^{\infty}(B^{(q)}E)^{n-1}M_x$ for all generations starting from generation 1. As every CRI starts with new transmissions only, $e_1^T \otimes I$ contains the same information for generation 0. Consequently, $(H^{(q)})_{i,j+(d+1)k}$ equals the expected total number of slots in a CRI in which j new packets and k retransmission occur, given that the CRI started with a slot containing i packets.

The number of users in the first slot of a CRI is determined by b. Therefore, if we reshape the size $(d + 1)^2$ vector $bH^{(q)}$ to the square $d + 1$ matrix $S^{(q)}$ (column by column), the latter contains the expected total number of slots with $i - j$ new packets and j retransmissions in a CRI in entry $(i - j,j)$.

A tagged packet will arrive together with $k - 1$ other new packets with probability kb_k/λ, where $\lambda = \sum_{k\geqslant 1} kb_k$, as it is k times more likely to be in a group of size k as opposed to being in a group of size 1. Similarly,

$$f_i^{(q)} = \frac{\sum_{j=0}^{i-1}(i-j)S_{i-j,j}^{(q)}}{\sum_{k=1}^{d}\sum_{j=0}^{k-1}(k-j)S_{k-j,j}^{(q)}},$$

for $i = 1, \ldots, d$, holds the probability that $i - 1$ other users also transmit when an arbitrary user makes his first transmission attempt, where some of the other $i - 1$ users may be retransmitting. For further use, denote $f_0^{(q)} = 0$.

4.3. Mean number of transmission attempts

In this section we will determine the mean number of transmission attempts needed to successfully transmit an arbitrary packet. This performance measure is of interest when defining the energy consumption in Section 4.5. We will set up a BP in which each node corresponds to a slot in which a tagged user transmits. To construct the expectation matrix $M_{tag}^{(q)}$ of this BP, we first introduce the matrix $B_{tag}^{(q)}$ which ignores new arrivals similar to $B^{(q)}$ defined in (1).

A tagged user that transmits together with $i - 1 \geqslant 1$ other users selects group k with probability p_k, for $k = 1, \ldots, d$. The probability that $j - 1$ out of the $i - 1$ other users select the same group, which results in a slot in which j users retransmit, clearly equals $\binom{i-1}{j-1}p_k^{j-1}(1-p_k)^{i-j}$. Combining these two observations implies that group G_k contains the tagged user together with $j - 1$ other users with probability $\binom{i-1}{j-1}p_k^{j}(1-p_k)^{i-j}$. As the tagged user may retransmit in any of the q groups, the average number of type j children of a type i node equals

$$B_{tag,i,j}^{(q)} = \begin{cases} \binom{i-1}{j-1}\sum_{k=1}^{q}p_k^j(1-p_k)^{i-j} & i \geqslant 2, i \geqslant j > 0, \\ 0 & \text{otherwise}. \end{cases} \quad (3)$$

As in Section 3, post-multiplying the matrix $B_{tag}^{(q)}$ with E adds the new arrivals. Hence, the expectation matrix $M_{tag}^{(q)} = B_{tag}^{(q)}E$. Using the same line of reasoning as in Section 4.1, we now define $T^{(q)}$ as

$$T^{(q)} = \left(I - B_{tag}^{(q)}E\right)^{-1},$$

where entry (i,j) of $T^{(q)}$ holds the expected total number of slots in which a tagged user transmits together with $j - 1$ other users, given that the initial transmission took place together with $i - 1$ other users. Using the probabilities

$f_i^{(q)}$ obtained in Section 4.2, the expected number of transmission attempts of a tagged used is given by $E[T] = f^{(q)}T^{(q)}e$, where $f^{(q)} = \left(f_0^{(q)}, f_1^{(q)}, \ldots, f_d^{(q)}\right)$.

4.4. Mean delay

When counting the number of slots required by an arbitrary user to transmit a packet successfully, we define two classes of slots: class (I) consists of the slots in which the user tries to transmit his packet and class (II) consists of the slots that lie in between these transmission attempts. Note, class (II) slots are those slots that, given that the tagged user chooses group $s > 1$, are used for resolving groups 1 to $s - 1$. Both classes have $d + 1$ types, where the type of a slot corresponds to the number of packets transmitted in the slot, leading to a total of $2(d + 1)$ types. Although class (I) slots holding zero packets cannot exist, we use them in our description as it simplifies the notation.

When ignoring the new arrivals, class (I) slots generate new class (I) slots according to the matrix $B_{tag}^{(q)}$ defined in (3). Due to the definition of class (II) slots, class (I) slots generate new class (II) nodes if the tagged user chooses some group $s > 1$. Given that the tagged user collided with $i - 1$ other users and selected the sth group, the expected number of class (II) slots of type j equals

$$\varphi_{i,j,s} = \binom{i-1}{j} \sum_{k=1}^{s-1} p_k^j (1 - p_k)^{i-1-j}. \tag{4}$$

In order to generate class (II) slots, the tagged user can choose any group s ranging from 2 to q. Combining this with (4) yields

$$B_{notag,i,j}^{(q)} = \begin{cases} \sum_{s=2}^{q} p_s \varphi_{i,j,s} & i \geqslant 2,\ i > j, \\ 0 & \text{otherwise}. \end{cases} \tag{5}$$

The creation of offspring of a class (II) slot is clearly determined by the matrix $B^{(q)}$ defined in (1), as the q groups formed by a collision in a class (II) slot are also class (II) slots. This also means that class (II) nodes have no class (I) children. Hence, if we order the $2(d + 1)$ types of this branching process such that the class (I) nodes occur first, we have

$$B_{del}^{(q)} = \begin{bmatrix} B_{tag}^{(q)} & B_{notag}^{(q)} \\ 0 & B^{(q)} \end{bmatrix}.$$

Using $E_2 = I_2 \otimes E$, we now add the new arrivals by post-multiplying $B_{del}^{(q)}$ with E_2 such that $M_{del}^{(q)} = B_{del}^{(q)} E_2$ is the expectation matrix. If we define

$$D^{(q)} = \left(I - M_{del}^{(q)}\right)^{-1},$$

we can calculate the mean delay $E[D]$ of an arbitrary packet as $E[D] = g^{(q)}D^{(q)}e + 0.5$, where $g^{(q)} = (1,0) \otimes f^{(q)}$, as the initial collision is a class (I) slot and on average *0.5* slots pass between the arrival time of a packet and the start of the next time slot.

4.5. Energy usage

The TAs considered in this paper are free access algorithms, therefore a user only needs to monitor the channel when he has a packet ready for transmission. Let $E[T]$ represent the mean number of transmission attempts of a tagged packet and $E[D]$ its mean delay, including the mean residual slot length of the slot in which the packet arrived. Normalizing the energy unit to the amount of energy needed for a single transmission attempt and using ξ to denote the amount of energy used for monitoring the channel during one time slot, [11] defines

$$E[W] = E[T] + \xi E[D].$$

as the mean energy usage $E[W]$. As we have Poisson arrivals, the mean residual slot length is half the mean length $E[S]$ of a slot in which a tagged arrival occurs. For most of the TAs considered in this paper all the slots have length 1, meaning $E[S] = 1$. However, for some TAs considered further on, the slot length distribution may depend on the number of packets that are transmitted in it. In such case we will determine $E[S]$ as follows. Denote q_i as the probability that i users transmit in an arbitrary slot. q_i can be calculated as

$$q_i = (bC^{(q)})_i / (bC^{(q)}e).$$

Next, define $l_{i,j}$ as the probability that a type i slot has length j. Due to the Poisson arrivals, the probability that a tagged arrival occurs in a slot of length j can be expressed as

$$\frac{j\left(\sum_{i=0}^{d} q_i l_{i,j}\right)}{\sum_{j \geqslant 1} j\left(\sum_{i=0}^{d} q_i l_{i,j}\right)},$$

and therefore the expected length of a slot in which a tagged arrival occurs is

$$E[S] = \frac{\sum_{j \geqslant 1} j^2\left(\sum_{i=0}^{d} q_i l_{i,j}\right)}{\sum_{j \geqslant 1} j\left(\sum_{i=0}^{d} q_i l_{i,j}\right)}.$$

4.6. Validation

In this section we investigate the impact of the truncation parameter d on the accuracy of the results. For the basic binary TA an expression for $E[C]$, $E[D]$ and the probabilities $f_i^{(2)}$ can be found in [9], while $E[T]$ was determined in [11]. All of these expressions make use of the operator $S(f(\cdot),z)$, which is defined as a sum over some semi-group H. Evaluating the operator is therefore computationally heavy and even obtaining the first few digits requires care. However, in case $p_1 = p_2 = 1/2$, the sum over H can be simplified significantly leading to

$$S(f(\cdot),z) = f(z) - f(0) - zf'(0) + \sum_{n \geqslant 1} [2^n f(\lambda_n + z/2^n) - 2^n f(\lambda_n) - zf'(\lambda_n)], \tag{6}$$

where $\lambda_n = \lambda(2 - 1/2^{n-1})$. Although some care is still needed to avoid bit cancellation between $2^n f(\lambda_n + z/2^n)$ and $2^n f(\lambda_n)$, we can use this formula to compute the perfor-

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

mance measures of the basic binary TA with arbitrary precision to determine the impact of d on the accuracy.

Fig. 1 depicts the relative error of our BP approach as a function of the parameter d for the basic binary TA with $p_1 = 1/2$ and $\lambda = 0.35$. It indicates that highly accurate results can be obtained with moderate d values, i.e., $d \leqslant 20$. Decreasing λ tends to further reduce the relative error (note, the maximum stable throughput is approximately *0.3602*), while the absolute error of the probabilities $f_i^{(2)}$, for $i > 1$, tends to decrease with increasing i.

A similar comparison for $p_1 \neq 1/2$ cannot be made as evaluating $S(f(\cdot),z)$ to an accuracy of 20 digits is computationally too demanding. Small d values however still sufficed to reproduce the results in Tables I and II of [21] and Tables 3 and 4 of [11].

4.7. Numerical Examples

In this section we discuss some numerical results that can be readily obtained using our BP approach. Fig. 2 depicts the optimal splitting factor q that minimizes the mean energy usage $E[W]$ as a function of λ and ξ (when $p_i = 1/q$ for all i), where $E[W] = E[T] + \xi E[D]$. It shows that for low to moderate arrival rates λ, decreasing ξ tends to increase the optimal q value. This can be understood by noting that the energy usage is composed of two parts: $E[T]$ the energy needed for all the transmission attempts and $\xi E[D]$ the energy needed to monitor the channel. Increasing q should reduce $E[T]$ as the chance of another collision is reduced, but increases $E[D]$. As decreasing ξ reduces the cost of the increased delay, larger q values benefit more.

Fig. 3 shows the values of p_1 to p_q that minimize the mean delay as a function of λ for $q = 2$ and 3. As expected, the optimal p_1 is larger than $1/q$ for small arrival rates λ and tends to decrease as λ increases. For $q = 2$ there is however also a range of λ values for which the optimal p_1 is slightly *less* than 1/2. This is unexpected as setting $p_1 = 1/2 - \epsilon/2$ instead of $p_1 = 1/2 + \epsilon/2$, with $\epsilon > 0$, will on average result in ϵ more class (II) slots, which may seem to suggest that the delay should increase. However, the mean number of users transmitting in these class (II) slots is smaller, meaning fewer slots are required to resolve one class (II) slot. As λ increases the latter advantage becomes more important and at some point cancels the effect of having more class (II) slots. As the MST is optimal for $p_1 = 1/2$, the optimal p_1 converges back to 1/2 as λ approaches the optimal MST.

The lowest optimal p_1 value for $q = 2$ corresponds to $\lambda = 0.335$ and equals *0.4989*, which is still close to 0.5. However, if we consider the basic q-ary TA on a channel with multiple reception capabilities, where up to $k > 1$ packets can be received successfully in a single slot, we see more a substantial drop. For instance, for $k = 3$ the optimal p_1 is slightly below 0.49 when $\lambda = 0.952$, while for $k = 10$ and $\lambda = 3.631$ the optimal p_1 is close to 0.48. To analyze the performance of the basic q-ary TA on a channel with multiple reception capabilities it suffices to adapt the BP such that type 2, ..., k nodes do not generate any children either. In other words it suffices to replace the $i \geqslant 2$ condition appearing in (1), (3) and (5) by $i \geqslant k + 1$.

For $q = 2$ the results in Fig. 3 also confirm that the optimal p_1 converges to $2 - \sqrt{2}$ as λ goes to zero [9]. We also note that the optimal p_2 for $q = 3$ is not exactly equal to 1/3, neither does $p_1 - p_2$ match $p_2 - p_3$.

5. Other tree algorithms with free access

In this section we use the same BP approach to determine the mean CRI duration, number of transmission attempts, delay and energy usage for a variety of TAs with free access. We start with the somewhat artificial coordinated splitting TA and the variable packet length TA as these are the only other TAs for which we can validate our approach using existing results in the literature.

5.1. Coordinated splitting tree algorithm

Consider a channel with multiple reception capabilities, where the receiver can successfully receive up to k successful transmissions in a single slot. Further, if $i > k$ users are

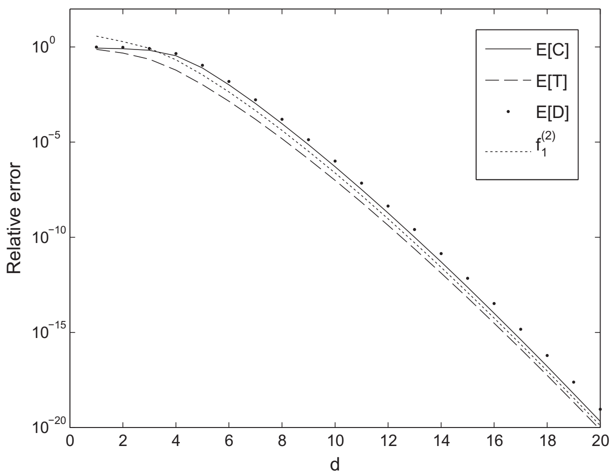

Fig. 1. Influence of the truncation parameter d on the mean CRI duration $E[C]$, the probability $f_1^{(2)}$, the mean number of transmissions $E[T]$ and the mean delay $E[D]$ for the basic binary TA with $p_1 = 1/2$ and $\lambda = 0.35$.

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

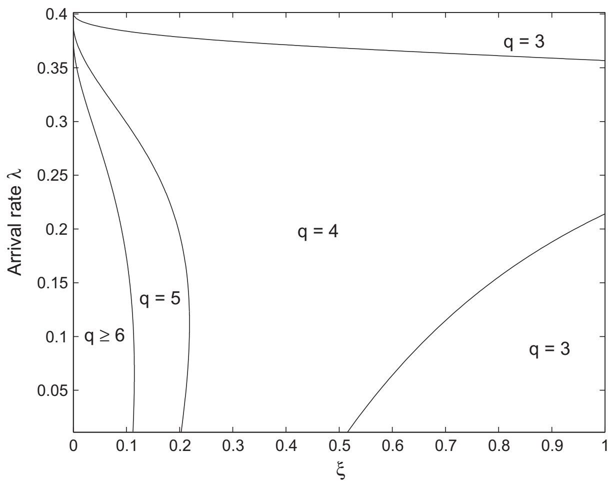

Fig. 2. Optimal splitting factor q to minimize the mean energy usage $E[W]$ using fair coins (i.e., $p_i = 1/q$ for all i).

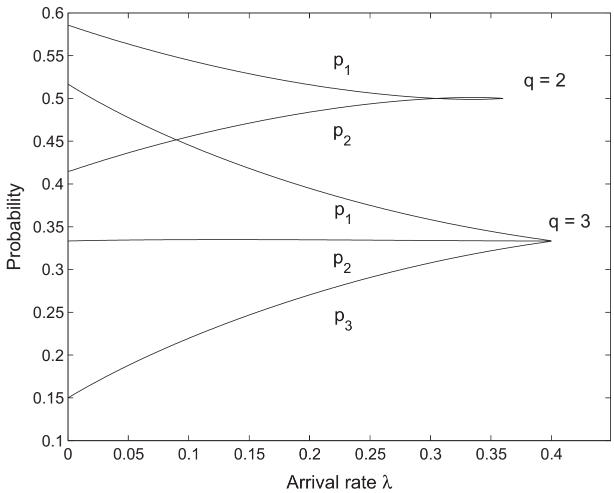

Fig. 3. Optimal probabilities p_1 to p_q to minimize the mean delay $E[D]$ for the binary ($q = 2$) and ternary ($q = 3$) basic TA as a function of the arrival rate λ.

involved in a collision, they are allowed to communicate with each other such that they split in exactly i groups containing one user. New users however still transmit in the next slot and users are unaware of the arrival times of the new packets.

The MST of the coordinated splitting TA with multiple reception capabilities was determined in [8, Section IV.I] by the BP with expectation matrix $M = BE$, where entry $B_{i,j}$ is equal to i, for $i > k$ and $j = 1$, and 0 otherwise. Therefore, the mean CRI duration $E[C] = b(I - M)^{-1}e$ and the probability f_i that a new packet is transmitted together with $i - 1$ other packets can be computed as in Section 4.2 if we replace $H^{(q)}$ by

$$\left(e_1^T \otimes I\right) + (I - M)^{-1} BE_x.$$

To determine the mean number of transmission attempts $E[T]$ and the mean delay $E[D]$, we can use the approach of Sections 4.3 and 4.4, respectively, we only need to redefine B_{tag} and B_{notag}.

B_{tag} counts the slots in which the tagged user transmits, hence $B_{tag,i,j}$ equals 1, for $i > k$ and $j = 1$, and 0 otherwise. For $B_{notag,i,j}$ we need to determine the mean number of class (II) slots of type j (with the new arrivals ignored). As a collision of i users is split into i groups holding a single packet, we have on average $(i - 1)/2$ groups ahead of the tagged user. We may therefore conclude that $B_{notag,i,j}$ equals $(i - 1)/2$, for $i > k$ and $j = 1$, and 0 otherwise.

The behavior of the coordinated splitting TA with $k = 1$ coincides with the TA considered in [12] when the number of sub-control channels equals infinity. The approach taken in [12] to determine the mean delay for this particular case, which contains a number of typos, can be generalized to any $k \geqslant 1$ and results in the following expression for the mean delay $E[D]$

$$E[D] = \frac{3}{2} + \frac{\lambda + \lambda^2 - p_0 \sum_{i=1}^{k} i^2 b_i + (1 - p_0) \sum_{i=0}^{k-1} (1 - i^2) b_i}{2\lambda \left(\sum_{i=1}^{k} i b_{i-1} - \lambda \right)},$$

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

with $p_0 = \left(\lambda - \sum_{i=1}^{k} i b_{i-1}\right) / \left(k b_k - \sum_{i=1}^{k} b_{i-1}\right)$. This expression simplifies to

$$E[D] = \frac{2(1-\lambda+\lambda^2) + e^{-\lambda}(1-3\lambda)}{2(1-\lambda)(e^{-\lambda}-\lambda)}, \tag{7}$$

for $k = 1$. Numerical experiments not depicted here confirmed that the relative errors of our BP approach are below 10^{-20} using moderate d values, e.g., $d = 20$ for $k = 1$.

As the mean delay tends to infinity as λ approaches the MST, this expression indicates that the MST is the smallest non-negative solution to the equation $\sum_{i=1}^{k} i b_{i-1} = \lambda$ (as the denominator of p_0 is negative). This allows us to confirm the results in the first column of Table V in [8].

5.2. Variable packet length tree algorithm

In this section we consider variable length packets, where a packet requires k slots with probability l_k. We assume that once the first slot is transmit successfully the channel is reserved for the remainder of the packet. New arrivals that occur during a successful transmission are resolved separately and their CRI starts in the first slot after the successful transmission, while possible retransmissions wait. This setting was originally introduced and analyzed in [22,10], allowing us to validate the results for the mean CRI duration $E[C]$ and mean delay $E[D]$. As in [10] we use a slightly different definition for the CRI duration, that is, if a CRI ends in a success according to the definition in Section 4.1, the next CRI is also considered part of the previous CRI.

The BP with expectation matrix $M_{var}^{(q)}$ is constructed as in [8, Section IV.H] via a minor change to the BP of Section 4. In particular, the reservation of a slot for new arrivals after every successful transmission is modeled as a child node of a type 1 slot. Therefore, $M_{var}^{(q)}$ is defined as

$$(M_{var}^{(q)})_{i,j} = \begin{cases} \sum_{k=1}^{\infty} l_k \frac{(\lambda k)^j}{j!} e^{-\lambda k} & i=1,\ j<d, \\ \sum_{k=1}^{\infty} l_k \sum_{m=d}^{\infty} \frac{(\lambda k)^m}{m!} e^{-\lambda k} & i=1,\ j=d, \\ (B^{(q)}E)_{i,j} & \textit{otherwise}. \end{cases}$$

The mean CRI duration $E[C]$ is computed as $b\left(I - M_{var}^{(q)}\right)^{-1} l_{var}$, with $l_{var} = (1, \sum_{k=1}^{\infty} k l_k, 1, \ldots, 1)^T$; where the duration of the successful transmissions is taken into account via the vector l_{var}.

As the child node of a type 1 node does not contain any retransmissions, only the first $d+1$ columns of $M_x^{(q)} = B^{(q)} E_x$ need to be modified, resulting in

$$\left(M_{var,x}^{(q)}\right)_{i,j} = \begin{cases} \sum_{k=1}^{\infty} l_k \frac{(\lambda k)^j}{j!} e^{-\lambda k} & i=1,\ j<d, \\ \sum_{k=1}^{\infty} l_k \sum_{m=d}^{\infty} \frac{(\lambda k)^m}{m!} e^{-\lambda k} & i=1,\ j=d, \\ \left(B^{(q)}E_x\right)_{i,j} & \textit{otherwise}, \end{cases}$$

which leads to $H_{var}^{(q)} = (e_1^T \otimes I) + \left(I - M_{var}^{(q)}\right)^{-1} M_{var,x}^{(q)}$.

The child of a type 1 node in which the tagged user transmitted is neither a class (I) or class (II) slot, such that $M_{var,tag}^{(q)} = B_{tag}^{(q)} E$ and

$$M_{var,del}^{(q)} = \begin{bmatrix} M_{tag}^{(q)} & M_{notag}^{(q)} \\ 0 & M_{var}^{(q)} \end{bmatrix}. \tag{8}$$

When all the packets have length 1, $E[T]$ reflects both the mean number of transmission attempts and the mean amount of energy spent by these transmission attempts. For variable length packets we define $E[T]$ as the energy usage of the transmission attempts (such that the definition of $E[W] = E[T] + \xi E[D]$ remains meaningful), meaning $E[T] = f_{var}^{(q)}\left(I - M_{var,tag}^{(q)}\right)^{-1} l_{var}$, where $f_{var}^{(q)}$ is computed as in Section 4.2 using $H_{var}^{(q)}$. Finally, $E[D] = \left((1,0) \otimes f_{var}^{(q)}\right) \left(I - M_{var,del}^{(q)}\right)^{-1} ((1,1)^T \otimes l_{var}) + E[S]/2$, where $E[S]$ was defined in Section 4.5.

Fig. 4 depicts the relative error of the BP approach as a function of d for $p_1 = 1/2$ when compared to the functional equation approach in [22,10], where we made use of (6) to accurately compute $E[D]$. Three different distributions for the packet length with mean 10 are considered: (i) deterministic lengths, (ii) uniform length between 2 and 18 and (iii) packets of length 2 and 18 only (with equal probability). The results show that small d values suffice to get highly accurate results. The error tends to increase as the packet length distribution becomes more variable, which is not unexpected as increased variability in the packet lengths also decreases the MST (from 0.876 to 0.867 and 0.852 for the three distributions considered in Fig. 4).

5.3. Tree algorithms with probabilistic capture

In this section we consider probabilistic capture. Capture is defined as the event in which $i > 1$ users transmit simultaneously and the receiver is able to decode x of the i transmitted signals, meaning $i - x$ users involved in the collision need to retransmit. In the channel model considered here, we assume the receiver only recognizes the decoded packets, i.e., he is not able to detect whether more than x packets were transmitted. Consequently, the receiver sends feedback to acknowledge the successful reception of the x decoded packets. Whenever $i > x \geqslant 1$, we state that the success slot holds a hidden collision. Clearly, only the users that were unsuccessful during a hidden collision are aware of the fact that a hidden collision took place. As these users do not know how many users were involved in the hidden collision, they retransmit in the next slot (in the hope that they were the only hidden user). In order to maximize the probability of success during these retransmissions, the other users should refrain from retransmitting their packet until the hidden collision has been resolved. However, as they cannot distinguish successful slots from hidden collisions, they should refrain from retransmitting after each success. This is accomplished by modifying the basic TA such that the counter is only decreased after idle slots. In the absence of capture, this algorithm coincides to the variable packet length

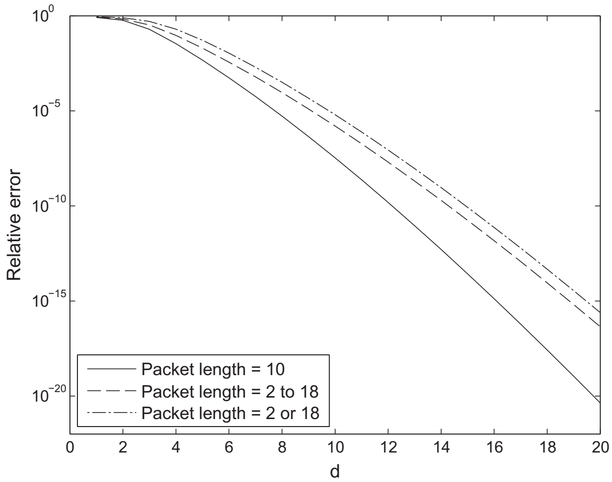

Fig. 4. Influence of the truncation parameter d on the mean delay $E[D]$ for the variable length TA with $q = 2$, $p_1 = 1/2$ and $\lambda = 0.08$ for three packet length distributions (with mean 10).

algorithm of Section 5.2 if we set the packet length equal to 1 slot.

Given that the capture probabilities are fixed and can be determined in advance, e.g., using the channel model with Rician Fading as described in [23], we will show how to adapt the branching process technique to determine the main performance measures of this tree algorithm with probabilistic capture. We define γ_i^x as the probability that x packets are decoded out of i simultaneously transmitted packets. For the sake of simplicity, we assume idle slots are always correctly perceived as idle and that whenever a slot contains a single packet, that packet is always successfully decoded, i.e. $\gamma_0^0 = \gamma_1^1 = 1$. We treat this type of errors in Section 5.6, albeit without considering probabilistic capture. However, it is also possible to combine both models.

The first step is to group the capture probabilities in the diagonal matrices $P^{(x)}$ of which the ith entry on the main diagonal corresponds to γ_i^x. Consequently, $\sum_{x \geqslant 0} P^{(x)} = I$. Then, we define $B^{(q,x)}$, entry (i,j) of which holds the expected number of type j slots after a type i collision of which x packets were decoded (without taking the new arrivals into account),

$$B_{i,j}^{(q,x)} = \begin{cases} B_{i,j}^{(q)} & x = 0, \\ 1 & x > 0,\ j = i - x \geqslant 0, \\ 0 & \text{otherwise}. \end{cases}$$

Indeed, if no packets were decoded, the entries of this matrix correspond to the ones of the basic TA, while if $x > 0$ packets were decoded out of the $i \geqslant x$ transmitted packets, the next slot is reserved for the retransmission of the $x - i$ remaining packets, leading to a single $i - x$ type branch.

Similarly, $B_{tag,i,j}^{(q,x)}$ contains the expected number of type j slots in which the tagged user transmits after he was involved in a collision together with $i - 1$ other users of which x packets were successfully decoded. If x out of the i packets were successfully decoded, the tagged is unsuccessful with probability $\frac{i-x}{i}$. Hence,

$$B_{tag,i,j}^{(q,x)} = \begin{cases} B_{tag,i,j}^{(q)} & x = 0, \\ \frac{j}{i} & x > 0,\ j = i - x \geqslant 0, \\ 0 & \text{otherwise}. \end{cases}$$

Finally, we note that class (II) slots can only be generated after a collision is split. Hence, as hidden collisions are never split, no class (II) slots are generated by a hidden collision. Consequently, $B_{notag}^{(q,0)} = B_{notag}^{(q)}$ and $B_{notag,i,j}^{(q,x)} = 0$ for $x > 0$. Using these matrices, the matrices $\overline{B}^{(q)}$, $\overline{B}_{tag}^{(q)}$ and $\overline{B}_{notag}^{(q)}$ can be determined using $\overline{B}^{(q)} = \sum_{x=0}^{d} P^{(x)} B^{(q,x)}$, $\overline{B}_{tag}^{(q)} = \sum_{x=0}^{d} P^{(x)} B_{tag}^{(q,x)}$ and $\overline{B}_{notag}^{(q)} = \sum_{x=0}^{d} P^{(x)} B_{notag}^{(q,x)}$, which in turn can be used as described in Section 4 to calculate $E[C]$, $E[T]$ and $E[D]$.

As an example, we used the setting as described in [24, Section IV]. In this setting, it is assumed that all the users are approximately at the same distance from the receiver, while they are capable of transmitting their packets using one of k different power levels: $\rho_1 < \rho_2 < \cdots < \rho_k$. Each time a user transmits a packet, he randomly selects one of the k allowed power levels, where level ρ_s is selected with probability σ_s. The packet of user r is decoded if his transmission level L_r is $C > 1$ times larger than the sum of the transmission levels of the other users combined (where C is a system parameter). In other words, during a collision of $i > 1$ packets, at most one packet is successfully received and this occurs whenever for some user r, $L_r \geqslant C\sum_{s=1,s \neq r}^{i} L_s$. In our example, we chose $C = 5$, the levels $\rho_1 = 1$, $\rho_2 = 10$ and $\rho_3 = 100$. Each level is chosen with equal probability, that is, $\sigma_1 = \sigma_2 = \sigma_3 = 1/3$. Table 1 lists both the results of the Monte Carlo simulations and the analytical results for different arrival rates λ. The MST of this setting was found to be 0.576576683. Each simulation was run 20 times, each run simulates 10^8 slots and has a warm up period of 20%. The results of each individual run were used to calculate the 95% confidence interval, depicted in the 3th column. It is clear that in each case, the results of our model are in perfect agreement with the simulation results.

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

Table 1
Comparison of simulation results and analytical results of the basic TA with $C = 5$ and random transmission levels uniformly selected out of $\{1, 10, 100\}$.

λ		Simulation results	Analytical results
0.11	$E[C]$	[1.130653, 1.130694]	1.130681
	$E[T]$	[1.104316, 1.104490]	1.104398
	$E[D]$	[1.653950, 1.654242]	1.654120
0.22	$E[C]$	[1.326807, 1.326919]	1.326849
	$E[T]$	[1.254336, 1.254519]	1.254428
	$E[D]$	[1.924258, 1.924634]	1.924414
0.33	$E[C]$	[1.672977, 1.673271]	1.673114
	$E[T]$	[1.481299, 1.481651]	1.481496
	$E[D]$	[2.501719, 2.503186]	2.502552
0.44	$E[C]$	[2.527108, 2.528043]	2.527509
	$E[T]$	[1.848859, 1.849294]	1.849062
	$E[D]$	[4.272574, 4.276838]	4.275304
0.55	$E[C]$	[10.186511, 10.216813]	10.186753
	$E[T]$	[2.504260, 2.505049]	2.504333
	$E[D]$	[23.830013, 23.927465]	23.836252

5.4. Modified tree algorithm

If the channel provides ternary (idle/success/collision) feedback, it is possible to improve the performance of the basic q-ary TA by skipping *doomed* slots that are guaranteed to hold a collision [6]. A slot is doomed whenever a collision is followed by $q-1$ empty slots, in which case all users involved in the collision must have selected the last group. The modified algorithm skips these doomed slots.

In [8, Section IV.B] the MST was obtained through the introduction of *virtual slots*. A virtual slot of type j is a skipped slot in which j packets are involved. Defining $P^{(q)}$ to be the matrix holding the expected number of virtual children, we have

$$P^{(q)}_{i,j} = \begin{cases} p_q^i b_0^{q-1} & i = j \geqslant 2, \\ 0 & \text{otherwise.} \end{cases}$$

Next, the matrix $\overline{M}^{(q)}$ was constructed using

$$\overline{M}^{(q)} = (B^{(q)} - P^{(q)})E + P^{(q)}.$$

In this matrix, we see that the virtual slots have corresponding nodes in the BP, but no new arrivals are allowed in them. It was noted that although the close connection between slots and nodes in the BP is lost, this does not influence the MST.

When calculating the mean CRI duration and delay on the other hand, it is slightly easier to restore the one-to-one correspondence between real slots and nodes in the BP. For this purpose, we create a slightly different BP, in which nodes corresponding to virtual slots are recursively replaced by their real offspring. The expectation matrix of this BP is denoted by $\widetilde{M}^{(q)}$, containing on position (i,j) the average number of type j slots created out of a type i slot. Using this matrix, we can express the average number of type j slots created out of a virtual type i slot through $(P^{(q)}\widetilde{M}^{(q)})_{i,j}$. Consequently, $\widetilde{M}^{(q)}$ can be represented recursively as

$$\widetilde{M}^{(q)} = \left(B^{(q)} - P^{(q)}\right)E + P^{(q)}\widetilde{M}^{(q)},$$

leading to

$$\widetilde{M}^{(q)} = \left(I - P^{(q)}\right)^{-1}\left(B^{(q)} - P^{(q)}\right)E,$$

and thus

$$\widetilde{B}^{(q)} = \left(I - P^{(q)}\right)^{-1}\left(B^{(q)} - P^{(q)}\right).$$

As explained in Section 4.1, the mean CRI duration can be computed via $E[C] = b(I - \widetilde{M}^{(q)})^{-1}e$.

In order to obtain $E[T]$ and $E[D]$, we first calculate the probabilities $\tilde{f}^{(q)}$ using $\widetilde{H}^{(q)} = (e_1^T \otimes I) + \left(I - \widetilde{M}^{(q)}\right)^{-1}\widetilde{B}^{(q)}E_x$ similar to $H^{(q)}$ in Section 4.2. As any virtual child of a tagged slot is tagged, $\widetilde{M}^{(q)}_{tag}$ is derived analoge to $\widetilde{M}^{(q)}$, yielding

$$\widetilde{B}^{(q)}_{tag} = \left(I - P^{(q)}\right)^{-1}\left(B^{(q)}_{tag} - P^{(q)}\right).$$

For $\widetilde{B}^{(q)}_{notag}$, note that real class (I) slots do not generate virtual class (II) slots, while virtual class (I) slots can generate class (II) slots. Therefore we have

$$\widetilde{M}^{(q)}_{notag} = B^{(q)}_{notag}E + P^{(q)}\widetilde{M}^{(q)}_{notag},$$

resulting in

$$\widetilde{B}^{(q)}_{notag} = \left(I - P^{(q)}\right)^{-1}B^{(q)}_{notag}.$$

$E[T]$ and $E[D]$ can be calculated using $\widetilde{B}^{(q)}_{tag}$ and $\widetilde{B}^{(q)}_{notag}$ using the technique explained in Sections 4.3 and 4.4.

Fig. 5 depicts the optimal splitting factor q that minimizes the mean energy usage $E[W]$ as a function of λ and ξ for the modified TA. Compared to Fig. 2, the size of region where the optimal q is larger than or equal to n (with $n = 4$, 5, 6) has decreased. This can be understood by noting that the modified TA will reduce both $E[T]$ and $E[D]$, but the reduction is more significant for smaller q values as on average more slots are skipped for smaller q values.

5.5. Tree algorithms with collision detection

In this section we consider a channel with collision detection capabilities. Let $l_c < 1$ be the fraction of a slot needed to detect a collision and assume that if a collision is detected, feedback is provided immediately. This results in shorter collision slots, hence on average fewer new arrivals will transmit in the next slot. Due to the Poisson arrival process, the probability of having i arrivals in a collision slot is $b_i^{(cd)} = exp(-\lambda l_c)(\lambda l_c)^i/i!$. As the first group retransmits in the next slot, the number of new arrivals in this group will be distributed according to $b_i^{(cd)}$. For the remaining $q-1$ other groups, b_i is still used as any CRI ends with a success or idle slot. In this section, we consider both the basic and the modified TA.

5.5.1. Basic tree algorithm

To construct the matrix $M^{(q)}_{cd}$, [8, Section IV.D] first defined E_{cd} as E, but with b_i replaced by $b_i^{(cd)}$. Then an expression was given for $B_k^{(q)}$, element (i,j) of which contains the expected number of type j nodes created by a type i node when considering the kth group only, i.e.,

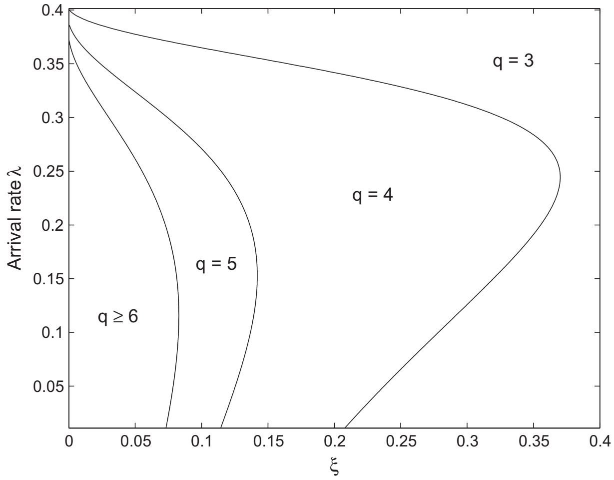

Fig. 5. Optimal splitting factor q to minimize the mean energy usage $E[W]$ using fair coins (i.e., $p_i = 1/q$ for all i).

$$\left(B_k^{(q)}\right)_{i,j} = \begin{cases} \binom{i}{j} p_k^j (1-p_k)^{i-j} & i \geqslant 2,\ i \geqslant j > 0, \\ 0 & \text{otherwise}. \end{cases}$$

Finally, $M_{cd}^{(q)}$ was calculated such that the new arrivals of the first child make use of E_{cd} rather than E,

$$M_{cd}^{(q)} = \left(B^{(q)} - B_1^{(q)}\right)E + B_1^{(q)}E_{cd}.$$

Apart from determining the MST, $M_{cd}^{(q)}$ can be used directly to express the mean number of slots per CRI via $b(I - M_{cd}^{(q)})^{-1}e$. However, to express the CRI duration $E[C]$ in time units, we need to take the slot lengths into account. Letting l_{cd} be the $d+1$ vector $l_{cd} = (1, 1, l_c, \ldots, l_c)$, $E[C]$ can be written as $b(I - M_{cd}^{(q)})^{-1}l_{cd}$. The probabilities f_i are calculated using the same procedure as explained in Section 4.2, replacing $H^{(q)}$ by

$$(e_1^T \otimes I) + \left(I - M_{cd}^{(q)}\right)^{-1}\left(\left(B^{(q)} - B_1^{(q)}\right)E_x + B_1^{(q)}E_{cd,x}\right),$$

where $E_{cd,x} = I_{d+1} \otimes b_{cd}$.

For $E[T]$ and $E[D]$, we introduce $B_{k,tag}^{(q)}$ and $B_{k,notag}^{(q)}$ which relate to $B_{tag}^{(q)}$ and $B_{notag}^{(q)}$, respectively, as $B_k^{(q)}$ relates to $B^{(q)}$:

$$\left(B_{k,tag}^{(q)}\right)_{i,j} = \begin{cases} \binom{i-1}{j-1} p_k^j (1-p_k)^{i-j} & i \geqslant 2,\ i \geqslant j > 0, \\ 0 & \text{otherwise}, \end{cases}$$

and

$$\left(B_{k,notag}^{(q)}\right)_{i,j} = \begin{cases} \binom{i-1}{j} p_k^j (1-p_k)^{i-1-j} \sum_{s=k+1}^{q} p_s & i \geqslant 2,\ i > j, \\ 0 & \text{otherwise}. \end{cases}$$

Define $B_{k,del}^{(q)}$ as $B^{(q)}$ by replacing $B_{tag}^{(q)}$ and $B_{notag}^{(q)}$ by $B_{k,tag}^{(q)}$ and $B_{k,notag}^{(q)}$, respectively. Taking the different slot lengths into consideration, let

$$T_{cd}^{(q)} = \left(I - \left(\left(B_{tag}^{(q)} - B_{1,tag}^{(q)}\right)E + B_{1,tag}^{(q)}E_{cd}\right)\right)^{-1} diag(l_{cd}),$$

and

$$D_{cd}^{(q)} = \left(I - \left(\left(B_{del}^{(q)} - B_{1,del}^{(q)}\right)E_2 + B_{1,del}^{(q)}E_{cd,2}\right)\right)^{-1} diag(l_{cd,2}),$$

where $E_{cd,2} = I_2 \otimes E_{cd}$, $l_{cd,2} = (1,1) \otimes l_{cd}$ and $diag(x)$ is a diagonal matrix with x on its main diagonal. These matrices can be used to determine $E[T]$ and $E[D]$ as in Sections 4.3 and 4.4.

5.5.2. Modified tree algorithm

Combining the arguments used in Sections 5.4 and 5.5.1, the BP corresponding to the modified TA with collision detection is acquired. As the vector b_{cd} determines the number of new arrivals in a slot following a collision, the matrix $P^{(q)}$ needs to be adapted to

$$\left(P_{cd}^{(q)}\right)_{i,j} = \begin{cases} p_q^i b_0^{(cd)} b_0^{q-2} & i = j \geqslant 2, \\ 0 & \text{otherwise}. \end{cases}$$

A virtual slot always follows an idle slot, hence the collision detection mechanism does not affect the splitting of a virtual slot. Therefore, we get

$$\widetilde{M}_{cd}^{(q)} = \left(B^{(q)} - B_1^{(q)} - P_{cd}^{(q)}\right)E + B_1^{(q)}E_{cd} + P_{cd}^{(q)}\widetilde{B}^{(q)}E,$$

$$\widetilde{M}_{cd,tag}^{(q)} = \left(B_{tag}^{(q)} - B_{1,tag}^{(q)} - P_{cd}^{(q)}\right)E + B_{1,tag}^{(q)}E_{cd} + P_{cd}^{(q)}\widetilde{B}_{tag}^{(q)}E,$$

and

$$\widetilde{M}_{cd,del}^{(q)} = \left(B_{del}^{(q)} - B_{1,del}^{(q)} - P_{cd,2}^{(q)}\right)E_2 + B_{1,del}^{(q)}E_{cd,2} + P_{cd,2}^{(q)}\widetilde{B}_{del}^{(q)}E_2,$$

where $P_{cd,2}^{(q)} = I_2 \otimes P_{cd}^{(q)}$, which can be used to obtain $E[C]$, $E[T]$ and $E[D]$.

5.6. Tree algorithms on a noisy channel

To study a more realistic setting, noise has been incorporated into the channel model [16,17]. On such a channel it is possible to perceive an empty slot, or a slot in which a single user transmits, as a collision. As in [16], we assume memoryless channel errors and define δ and ϵ respectively as the probability that an empty slot and a single transmission attempt are perceived as a collision.

5.6.1. Basic tree algorithm

Due to the errors on the channel, type 0 and type 1 nodes may have offspring. More specifically, when ignoring the new arrivals, a type 0 node has q type 0 children with probability δ and a type 1 node has $q-1$ type 0 and 1 type 1 child with probability ϵ. Define

$$(B_e^{(q)})_{i,j} = \begin{cases} \binom{i}{j}\sum_{k=1}^{q} p_k^j (1-p_k)^{i-j} & i \geqslant 0,\ i \geqslant j, \\ 0 & \text{otherwise.} \end{cases}$$

and let $D = diag(\delta, \epsilon, 1, \ldots, 1)$. The expectation matrix $M_e^{(q)}$ of the new BP is defined as $M_e^{(q)} = DB_e^{(q)}E$, from which $E[C]$ and $f_e^{(q)}$ can be computed as in Section 4. Note this BP is not exactly the same as in [8, Section IV.C], but results in the same MST.

To determine the expected number of slots in which a tagged user transmits, we note that a tagged user is never part of the child of an empty slot. Therefore,

$$\left(B_{e,tag}^{(q)}\right)_{i,j} = \begin{cases} \binom{i-1}{j-1}\sum_{k=1}^{q} p_k^j (1-p_k)^{i-j} & i \geqslant 1,\ i \geqslant j > 0, \\ 0 & \text{otherwise,} \end{cases}$$

leading to $M_{e,tag} = DB_{e,tag}E$. Similarly,

$$\left(B_{e,notag}^{(q)}\right)_{i,j} = \begin{cases} \sum_{s=2}^{q} p_s \varphi_{i,j,s} & i \geqslant 1,\ i > j, \\ 0 & \text{otherwise,} \end{cases}$$

and $M_{e,notag} = DB_{e,notag}E$, from which the mean number of transmission attempts $E[T]$, mean delay $E[D]$ and energy usage $E[W]$ can be obtained as before.

5.6.2. Modified tree algorithm

A slot can only be skipped if the first $q-1$ slots are detected as empty, therefore

$$\left(P_e^{(q)}\right)_{i,j} = \begin{cases} p_q^i ((1-\delta)b_0)^{q-1} & i = j \geqslant 0, \\ 0 & \text{otherwise.} \end{cases}$$

As children of skipped slots may also be skipped, we find

$$\widetilde{M}_e^{(q)} = D\left(I - P_e^{(q)}\right)^{-1}\left(B_e^{(q)} - P_e^{(q)}\right)E.$$

Note the inverse $\left(I - P_e^{(q)}\right)^{-1}$ is used as a skipped child of a type 0 or 1 node generates offspring with probability 1 (instead of with probability δ and ϵ, respectively).

Similarly, we find

$$\widetilde{M}_{e,tag}^{(q)} = D\left(I - P_e^{(q)}\right)^{-1}\left(B_{e,tag}^{(q)} - P_e^{(q)}\right)E,$$

and

$$\widetilde{M}_{e,notag}^{(q)} = D\left(I - P_e^{(q)}\right)^{-1} B_{e,notag}^{(q)} E.$$

The relative increase in the mean energy usage of the basic and modified TA due to the presence of errors on the channel is shown in Fig. 6 for $\lambda = 0.3$. We consider three cases: channel errors occur with probability e on the empty slots ($\delta = e, \epsilon = 0$), on successful transmissions ($\delta = 0, \epsilon = e$), or on both ($\delta = \epsilon = e$). The results indicate that the relative performance of the modified algorithm is more sensitive to the channel errors. This is especially true for the setups with $\delta = e$ as the modified algorithm gets trapped in a loop until a new arrival occurs each time an empty slot is perceived as a collision. In three of the six cases the algorithm has an MST below 0.3 for e sufficiently large (and below 0.5).

5.7. Tree algorithms with control subchannels

In this section every slot is divided into $g \geqslant 2$ mini-slots, called control mini-slots (CMSs), and a data slot. The length of the data slot corresponds to the length of a single packet, the length of the CMSs is kept as short as possible, but long enough for users to announce their presence. At the end of each slot separate feedback is provided for each of the CMSs and the data slot. The feedback given for the data slot can be either binary or ternary, although we restrict ourselves to binary feedback in this section. For each CMS, the feedback can be binary or ternary as well, but the binary feedback differs from the known feedback as it distinguishes between something/nothing as opposed to collision/no collision. The BF/BF and TF/BF algorithms considered in this section are identical to the g-ary FA TA/M-BF (1) and FA TA/M-TF (2) algorithm in [12], where the MST was determined analytically and the mean delay by simulation. The channel conditions are also identical to the BF/BF/DF and TF/BF/DF setup in [25], which focused on a number of block access algorithms.

5.7.1. BF/BF

The BF/BF algorithm is very similar to the basic g-ary TA, but whenever a user transmits in the data slot it also randomly picks a CMS and announces its presence in that slot. The selected CMS determines the group the user will select in case of a collision (via a one-to-one mapping) and this information is then used to skip groups corresponding to empty CMSs.

As in [8], it suffices to define $B_{BF/BF}^{(g)}$ identical to $B^{(g)}$, except that its first column is equal to zero (as the empty groups are skipped). Further, $B_{BF/BF,tag}^{(g)} = B_{tag}^{(g)}$ as the first column of $B_{tag}^{(g)}$ is already zero, while $B_{BF/BF,notag}^{(g)}$ is equal to $B_{notag}^{(g)}$, except for the first column, which also equals zero. Given these changes the approach of Section 4 applies to determine $E[C]$, $E[T]$, $E[D]$ and $E[W]$.

5.7.2. TF/BF

If ternary feedback is provided for each CMS, not only empty groups can be skipped, but groups corresponding to mini-slots holding a collision can be split immediately into 2 groups. As indicated in [12] immediately splitting these guaranteed collisions into $m > 2$ groups reduces the MST. To construct $M_{TF/BF}^{(g)}$, [8] introduced $B_{TF/BF1}^{(g)}$ and $B_{TF/BF2}^{(g)}$, the former containing the slots corresponding to groups with one user and the latter containing the nodes

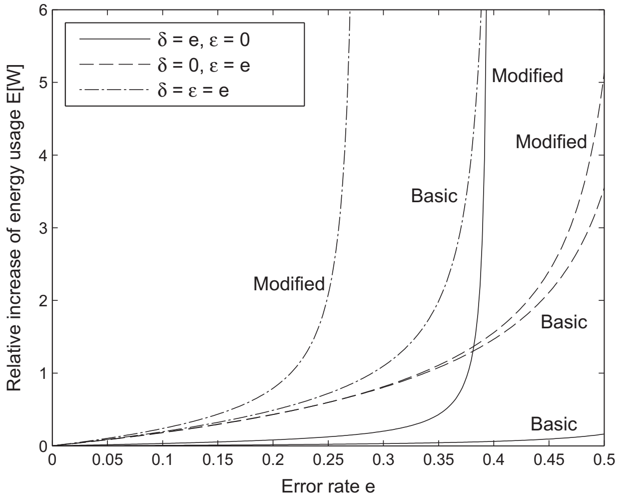

Fig. 6. Relative increase of the mean energy usage as a function of the error rate e for the basic and modified binary TA with $\lambda = 0.3$.

corresponding to groups containing more than one user. It should be clear that $(B^{(g)}_{TF/BF1})_{i,j} = B^{(g)}_{i,j}$ for $j = 1$ and zero elsewhere, while $(B^{(g)}_{TF/BF2})_{i,j} = B^{(g)}_{i,j}$ for $j > 1$ and zero elsewhere. $M^{(g)}_{TF/BF}$ was acquired next via

$$M^{(g)}_{TF/BF} = B^{(g)}_{TF/BF1}E + B^{(g)}_{TF/BF2}B^{(2)}E.$$

Analogous, $B^{(g)}_{TF/BF1,tag}$ and $B^{(g)}_{TF/BF1,tag}$ can be constructed through extracting the correct columns out of $B^{(g)}_{tag}$ and consequently we have

$$M^{(g)}_{TF/BF,tag} = B^{(g)}_{TF/BF1,tag}E + B^{(g)}_{TF/BF2,tag}B^{(2)}_{tag}E.$$

To construct $M^{(g)}_{TF/BF,notag}$, assume the tagged user chose CMS s. There are three possibilities in which a class (I) slot can generate a class (II) slot:

1. A user can be the single user that selected CMS $i < s$.
2. A user can select CMS $i < s$ together with at least one other user.
3. A user can collide with the tagged user in CMS s, but selects the first group after splitting the corresponding group, while the tagged user selects the second group.

After constructing $B^{(g)}_{TF/BF1,notag}$ and $B^{(g)}_{TF/BF2,notag}$ based on $B^{(g)}_{notag}$ (similar to the manner in which $B^{(g)}_{TF/BF1}$ and $B^{(g)}_{TF/BF2}$ were based on $B^{(g)}$), we can write $M^{(g)}_{TF/BF,notag}$ as

$$M^{(g)}_{TF/BF,notag} = B^{(g)}_{TF/BF1,notag}E + B^{(g)}_{TF/BF2,notag}B^{(2)}E + B^{(g)}_{TF/BF2,tag}B^{(2)}_{notag}E.$$

Figs. 7 and 8 depict the mean delay as a function of λ for the BF/BF and TF/BF algorithm respectively for $g = 2, 4, 8, 16$ and ∞. As g increases the mean delay decreases to the mean delay of the coordinated splitting TA given by (7). These figures are in agreement with the simulation results in Figs. 2 and 3 in [12].

If the channel supports TF/TF feedback, the modified binary TA can be used to split groups corresponding to CMSs holding collisions. To apply the technique to this TA merely requires replacing $B^{(2)}$, $B^{(2)}_{tag}$ and $B^{(2)}_{notag}$ by $\widetilde{B}^{(2)}$, $\widetilde{B}^{(2)}_{tag}$ and $\widetilde{B}^{(2)}_{notag}$.

5.7.3. Modified BF/BF

The performance of the BF/BF TA can be improved by remarking that if all the users selected the same CMS during a collision, we have a doomed slot that can be skipped. Using the same argument as in Section 5.4, we construct $P^{(g)}_{BF/BF}$:

$$(P^{(q)}_{BF/BF})_{i,j} = \begin{cases} \sum_{k=1}^{q} p^i_k & i = j \geqslant 2, \\ 0 & \text{otherwise.} \end{cases}$$

Note that as the doomed slots are detected via the CMSs, we do not need to include the factor b_0^{q-1}. The doomed slots are immediately split into two groups using the basic TA, leading to

$$\widetilde{B}^{(g)}_{BF/BF} = B^{(g)}_{BF/BF} - P^{(g)}_{BF/BF} + P^{(g)}_{BF/BF}B^{(2)}.$$

Similarly, we have

$$\widetilde{B}^{(g)}_{BF/BF,tag} = B^{(g)}_{BF/BF,tag} - P^{(g)}_{BF/BF} + P^{(g)}_{BF/BF}B^{(2)}_{tag},$$

and, as it is impossible for a class (I) slot to create a class (II) doomed slot,

$$\widetilde{B}^{(g)}_{BF/BF,notag} = B^{(g)}_{BF/BF,notag} + P^{(g)}_{BF/BF}B^{(2)}_{notag}.$$

Table 2 compares the MST of the modified BF/BF algorithm (which was not considered in [8]) with the BF/BF and TF/BF algorithm. Perhaps somewhat unexpected the MST of the modified BF/BF algorithm is closer to the MST of the TF/BF algorithm for moderate g values.

If the data channel allows ternary feedback (i.e., BF/TF feedback), we can use the modified TA to split detected doomed slots. $\widetilde{B}^{(g)}_{BF/TF}$, $\widetilde{B}^{(g)}_{BF/TF,tag}$ and $\widetilde{B}^{(g)}_{BF/TF,notag}$ are constructed analoge to the previous paragraph, replacing $B^{(2)}, B^{(2)}_{tag}$ and $B^{(2)}_{notag}$ by $\widetilde{B}^{(2)}$, $\widetilde{B}^{(2)}_{tag}$ and $\widetilde{B}^{(2)}_{notag}$.

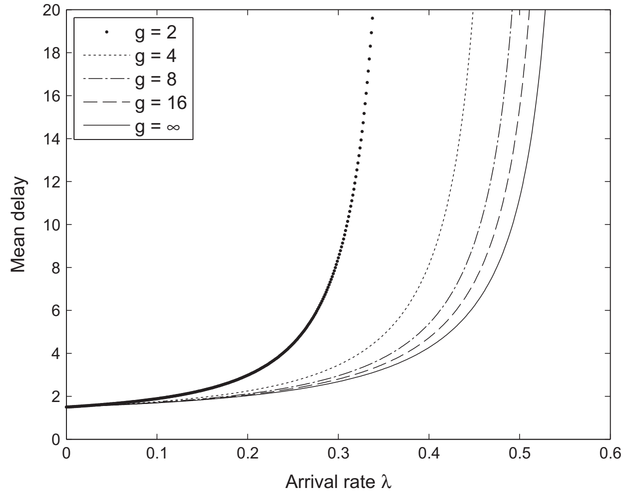

Fig. 7. Mean delay $E[D]$ as a function of λ for the BF/BF algorithm with $g = 2, 4, 8, 16$ and ∞.

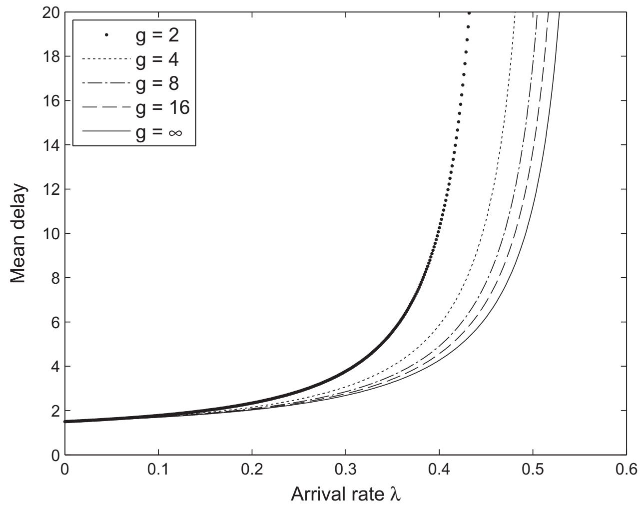

Fig. 8. Mean delay $E[D]$ as a function of λ for the TF/BF algorithm with $g = 2, 4, 8, 16$ and ∞.

Table 2
Maximum stable throughput of the BF/BF, modified BF/BF and TF/BF TAs with g control channels (not counting overhead).

g	BF/BF	modified BF/BF	TF/BF
2	0.376815	0.440312	0.470771
3	0.455546	0.483622	0.503665
4	0.488476	0.505441	0.519728
5	0.506464	0.518334	0.529281
10	0.538895	0.543377	0.548256
100	0.564490	0.564831	0.565255
∞	0.567143	0.567143	0.567143

5.8. Interference cancellation tree algorithm

Another improvement of the basic binary TA can be accomplished through the use of an *Interference Cancellation* (IC) mechanism [26,27], which allows the receiver to cancel out signals from each other. More specifically, assuming that the receiver at one point received the signal $v = w_A + w_B$ and at another point the signal w_A, he can construct the signal w_B using $w_B = v - w_A$. Further, we suppose this operation can be applied flawlessly. A free access TA using IC was introduced in [18] and its MST was also analyzed in [8, Section IV.F]. The core concept of the algorithm is to store collision signals such that when the first group retransmits, IC can be used to retrieve extra information, possibly decoding a packet or predicting the signal of the second group, allowing us to skip the corresponding slot in the latter case. For example, suppose i users collide and all but one user select group 1 while there are no new arrivals during the collision. In this case, the packet of the user that selected the second group is recovered via the cancellation operation. Further, the TA in [18] also makes use of a control bit that is set equal to 1 whenever a packet is transmitted for the first time.

Table 3
The cases in which a slot can be skipped after a collision of i packets.

Case	Type of skipped slot	Mean no. of occurrences
$0/i$	i	$(1-p)^i b_0$
$1/i-1$	$i-1$	$ip(1-p)^{i-1}b_0$
$1(new)/i$	i	$(1-p)^i b_1$
$i/0$	0	$p^i b_0$
$i-1/1$	1	$ip^{i-1}(1-p)b_0$
$i+1/0$	0	$p^i b_1$

Assume a collision of i packets occurs and use n_x/n_y to denote that n_x and n_y users chose the first and second group respectively, then Table 3 lists all cases in which the slot of the second group can be skipped. A detailed explanation can be found in [8]. For our purpose, it suffices to know the type of the skipped slot, the probability that it occurs (which is also listed in Table 3) and that a packet is recovered in the last two cases (note case 2 with $i = 2$ coincides with case 5). Moreover, in the last case, a new arrival is recovered and the algorithm continues with i users rather than $i + 1$ users.

To model the aforementioned TA, we first introduce $P^{(ic)}$, containing the expected number of skipped slots. Grouping the entries of Table 3 per skipped slot type leads to

$$P^{(ic)}_{i,j} = \begin{cases} p^i(b_0+b_1) & i \geqslant 2,\ j=0, \\ ip^{i-1}(1-p)b_0 & i \geqslant 2,\ j=1, \\ ip(1-p)^{i-1}b_0 & i \geqslant 2,\ j=i-1, \\ (1-p)^i(b_0+b_1) & i=j \geqslant 2, \\ 0 & \text{otherwise}. \end{cases} \quad (9)$$

This matrix is equal to the one introduced in [8] [Eq. (24)], we merely changed the notation. Next, noting that in cases 4, 5 and 6 (and 2 if $i = 2$) the skipped slot does not induce any new slots, we define $Q^{(ic)}$ to be

$$Q^{(ic)}_{i,j} = \begin{cases} p^i(b_0+b_1) & i \geqslant 2,\ j=0, \\ ip^{i-1}(1-p)b_0 & i \geqslant 2,\ j=1, \\ 0 & \text{otherwise}, \end{cases} \quad (10)$$

such that $P^{(ic)} - Q^{(ic)}$ only contains those skipped slots that are split immediately. To account for case 6, in which only i users of a type $i + 1$ slot are divided into groups, the matrix R is introduced as

$$R_{i,j} = \begin{cases} p^i b_1 & i=j \geqslant 2, \\ -p^i b_1 & i=j-1 \geqslant 2, \\ 0 & \text{otherwise}. \end{cases} \quad (11)$$

Finally we can combine previous matrices to obtain the expectation matrix $M^{(ic)}$ of the BP via

$$M^{(ic)} = \left(B^{(2)} - P^{(ic)}\right)E + R + \left(P^{(ic)} - Q^{(ic)}\right)M^{(ic)},$$

yielding

$$M^{(ic)} = \left(I - P^{(ic)} + Q^{(ic)}\right)^{-1}\left(\left(B^{(2)} - P^{(ic)}\right)E + R\right), \quad (12)$$

which can be used to compute $E[C]$ as before. Note in [8] we did not need the matrix $Q^{(ic)}$ as we were merely interested in the MST.

To calculate the probabilities $f_i^{(ic)}$, case 6 requires extra care. As a new arrival is successfully transmitted in case 6, the i retransmissions did not occur in this slot from the new user's perspective. Letting

$$(R_x)_{i,j} = \begin{cases} p^i b_1 & j=1, i \geqslant 2, \\ -p^i b_1 & j = 1+(d+1)i, i \geqslant 2, \\ 0 & \text{otherwise}, \end{cases}$$

we can therefore define $H^{(ic)} = (e_1^T \otimes I) + \left(I - M^{(ic)}\right)^{-1} \left(I - P^{(ic)} + Q^{(ic)}\right)^{-1}\left(\left(B^{(2)} - P^{(ic)}\right)E_x + R_x\right)$ and the technique introduced in Section 4.2 applies to compute the probabilities $f_i^{(ic)}$.

Only counting the skipped slots in which the tagged user transmits leads to

$$(P^{(ic)}_{tag})_{i,j} = \begin{cases} p^{i-1}(1-p)b_0 & i \geqslant 2,\ j=1, \\ (i-1)p(1-p)^{i-1}b_0 & i \geqslant 2,\ j=i-1, \\ (1-p)^i(b_0+b_1) & i=j \geqslant 2, \\ 0 & \text{otherwise}. \end{cases}$$

Similarly,

$$(Q^{(ic)}_{tag})_{i,j} = \begin{cases} p^{i-1}(1-p)b_0 & i \geqslant 2,\ j=1, \\ 0 & \text{otherwise}, \end{cases}$$

and $M^{(ic)}_{tag} = \left(I - P^{(ic)}_{tag} + Q^{(ic)}_{tag}\right)^{-1}\left(\left(B^{(2)}_{tag} - P^{(ic)}_{tag}\right)E + R\right)$, which can be used to calculate $E[T]$ as in Section 4.3. The mean delay $E[D]$ can be obtained by defining

$$P^{(ic)}_{del} = \begin{bmatrix} P^{(ic)}_{tag} & 0 \\ 0 & P^{(ic)} \end{bmatrix},$$

$$Q^{(ic)}_{del} = \begin{bmatrix} Q^{(ic)}_{tag} & 0 \\ 0 & Q^{(ic)} \end{bmatrix},$$

and $R_2 = I_2 \otimes R$.

6. Conclusion

We presented a branching process approach to determine the main performance measures, such as the mean delay and energy usage, of a broad class of tree algorithms with free access. The approach is closely related to the branching process approach introduced in [8] to determine the maximum stable throughput. As in [8], a key feature of the approach exists in the use of the truncation parameter d. Using the existing results [9,11,12,10,22], we demonstrated that our approach can provide highly accurate results, even for moderate values of d (e.g. $d < 20$). Moreover, most of the existing results are expressed using some operator $S(f(\cdot),z)$, the evaluation of which is computationally demanding (unless $p = 1/2$ due to the simplifications discussed in Section 4.6), whereas our approach requires hardly any computational effort. We further showed that our approach can also be used to produce many new results without much additional effort.

The same limitations as discussed in [8, Section V] also apply to the approach introduced in this paper. Further, the approach is limited to the determination of the mean

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022

performance measures and it is unclear whether it can be extended to obtain higher moments. The variance of the mean delay and CRI duration of the basic binary TA have been expressed in [9] via some operator $T(f(\cdot),z)$, the evaluation of which requires even heavier computations than $S(f(\cdot),z)$.

References

[1] J. Capetanakis, Tree algorithms for packet broadcast channels, IEEE Trans. Inform. Theory 25 (5) (1979) 505–515.
[2] B.S. Tsybakov, V. Mikhailov, Free synchronous packet access in a broadcast channel with feedback, Problemy Peredachi Informatsii 14 (4) (1978) 32–95.
[3] D.P. Bertsekas, R.G. Gallager, Data Networks, Prentice Hall, 1992.
[4] A. Ephremides, B. Hajek, Information theory and communication networks: an unconsummated union, IEEE Trans. Inform. Theory 44 (6) (1998) 2416–2434.
[5] M.L. Molle, G.C. Polyzos, Conflict Resolution Algorithms and their Performance Analysis, Tech. Rep. July 1993.
[6] J. Massey, Collision resolution algorithms and random-access communication, in: G. Longo (Ed.), CISM Courses and Lectures, vol. 256, Springer Verlag, Wien-New York, 1981, pp. 73–137.
[7] R. Rom, M. Sidi, Multiple Access Protocols: Performance and Analysis, Telecommunication Networks and Computer Systems, Springer-Verlag, 1990.
[8] G.T. Peeters, B. Van Houdt, On the maximum stable throughput of tree algorithms with free access, IEEE Trans. Inform. Theory 55 (11) (2009) 5087–5099.
[9] G. Fayolle, P. Flajolet, M. Hofri, P. Jacquet, Analysis of a stack algorithm for random multiple-access communication, IEEE Trans. Inform. Theory 31 (2) (1985) 244–254.
[10] P. Jacquet, E. Merle, Analysis of a stack algorithm for CSMA-CD random length packet communication, IEEE Trans. Inform. Theory 36 (2) (1990) 420–426.
[11] A. Bergman, M. Sidi, Energy efficiency of collision resolution protocols, Computer Communications 29 (17) (2006) 3397–3415.
[12] Y. Oie, T. Suda, H. Miyahara, T. Hasegawa, Throughput and delay analysis of free access tree algorithm with minislots, IEEE Transactions on Communications 38 (2) (1990) 137–141.
[13] B. Van Houdt, C. Blondia, Stability and performance of stack algorithms for random access communication modeled as a tree structured QBD markov chain, Stochastic Models 17 (3) (2001) 247–270.
[14] B. Van Houdt, Performance and Analysis of Contention Resolution Algorithms in Random Access Systems, Ph.D. Thesis, University of Antwerp, 2001.
[15] P. Mathys, P. Flajolet, *q*-Ary collision resolution algorithms in random-access systems with free or blocked channel access, IEEE Trans. Inform. Theory 31 (2) (1985) 217–243.
[16] N.D. Vvedenskaya, B.S. Tsybakov, Random multiple access of packets to a channel with errors, Problemy Peredachi Informatsii 19 (2) (1983) 52–68.
[17] R. Liang, H. Tan, On the error analysis of single-channel free-access collision resolution algorithms, in: Aerospace Conference Proceedings, 2000 IEEE, vol. 1, 2000, pp. 129–140.
[18] G.T. Peeters, B.V. Houdt, C. Blondia, A multiaccess tree algorithm with free access, interference cancellation and single signal memory requirements, Perform. Eval. 64 (9–12) (2007) 1041–1052.
[19] C. Mode, Multitype Branching Processes: Theory and Applications, Modern Analytic and Computational Methods in Science and Mathematics, American Elsevier Publishing Company, 1971.
[20] F.R. Gantmacher, Matrix Theory, American Mathematical Society 2 (2000).
[21] G. Fayolle, P. Flajolet, M. Hofri, On a functional equation arising in the analysis of a protocol for a multiaccess broadcast channel, Advances in Applied Probability 18 (1986) 441–472.
[22] P. Jacquet, E. Merle, Analysis of a Stack Algorithm for Random Length Packet Communication, Tech. Rep. RR-0831, INRIA, April 1988.
[23] M. Zorzi, Capture probabilities in random-access mobile communications in the presence of Rician fading, IEEE Transactions on Vehicular Technology 46 (1) (1997) 96–101.
[24] I. Cidon, H. Kodesh, M. Sidi, Erasure, capture, and random power level selection in multiple-access systems, IEEE Transactions on Communications 36 (3) (1988) 263–271.
[25] D. Towsley, P. Vales, Announced arrival random access protocols, IEEE Transactions on Communications 35 (5) (1987) 513–521.
[26] Y. Yu, G. Giannakis, SICTA: a 0.693 contention tree algorithm using successive interference cancellation, in: INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami (USA), 2005, pp. 1908–1916.
[27] Y. Yu, G. Giannakis, High-throughput random access using successive interference cancellation in a tree algorithm, IEEE Trans. Inform. Theory 53 (12) (2007) 4628–4639.

Robbe Block (robbe.block@ua.ac.be) received his M.Sc. degree in Computer Science from the University of Antwerp (Belgium) in July 2011. In October 2011, he became a PhD-fellow of the Research Foundation – Flanders (FWO) and joined the "Performance Analysis of Telecommunication Systems" research group, at the Mathematics and Computer Science Department of the University of Antwerp. His main research interests include the performance analysis of medium access control algorithms and cognitive radio networks.

Gino T. Peeters (gino.peeters@ua.ac.be) received his M.Sc. degree in Computer Science and a PhD in Science from the University of Antwerp (Belgium) in July 2005, and September 2009, respectively. His main research interests include the performance analysis of telecommunication systems, more specifically medium access control problems and scheduling issues in satellite communication networks. From July 2010 onwards he is working at Technicolor, Belgium.

Benny Van Houdt (benny.vanhoudt@ua.ac.be) received his M.Sc. degree in Mathematics and Computer Science, and a PhD in Science from the University of Antwerp (Belgium) in July 1997, and May 2001, respectively. From August 1997 until September 2001 he held an Assistant position at the University of Antwerp. Starting from October 2001 onwards he has been a postdoctoral fellow of the FWO-Flanders. In 2007, he became a professor at the Mathematics and Computer Science Department of the University of Antwerp, where he is a member of the PATS research group. His main research interest goes to the performance evaluation and stochastic modelling of wired and wireless communication networks. He has published a several papers, containing both theoretical and practical contributions, in a variety of international journals (e.g., IEEE JSAC, IEEE Trans. on Inf. Theory, IEEE Trans. on Comm., Performance Evaluation, Journal of Applied Probability, Stochastic Models, Queueing Systems, etc.) and in conference proceedings (e.g., ACM Sigmetrics, Networking, Globecom, Opticomm, ITC, etc.).

Please cite this article in press as: R. Block et al., A branching process approach to compute the delay and energy efficiency of tree algorithms with free access, Comput. Netw. (2013), http://dx.doi.org/10.1016/j.comnet.2013.08.022