A Novel Mechanism for Contention Resolution in HFC Networks

M.X. van den Broek[†,**], I.B.J.F. Adan[†], Sai Shankar N.[*], Sem Borst[†,‡,††]

[†]Department of Mathematics & Computer Science, Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[*]Philips Research Laboratories, 345 Scarborough Road, Briarcliff Manor, New York, USA

[‡]CWI – Center for Mathematics & Computer Science, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract— **The Medium Access Control (MAC) scheme proposed by DAVIC/DVB, IEEE 802.14 and DOCSIS for the upstream channel of Hybrid Fiber Coaxial (HFC) access networks is based on a mixable contention-based/contention-less time slot assignment. Contention-less slots are assigned by the head end to end stations according to a reservation scheme. Contention-based slots are randomly accessed by active terminals without any preliminary allocation, so that collisions may occur. To resolve contention, the contention tree algorithm has been widely accepted by the DVB/DAVIC, IEEE 802.14 and DOCSIS standards for MAC because of higher throughput and lower access delay. In this paper we propose a novel contention resolution mechanism and compare its performance with that of existing procedures. The proposed procedure is termed as static arrival slot mechanism. In this mechanism, one slot in each frame is exclusively reserved for new arrivals that wish to access the channel using contention resolution, and at least one slot is reserved for resolving their contention if there was one in the arrival slot. The performance of the proposed mechanism is evaluated through analysis and simulation. The results show that the proposed mechanism outperforms existing contention resolution procedures under heavy traffic.**

Index Terms— **Contention resolution, contention trees, HFC networks, reservation mechanisms, sojourn time, waiting time.**

I. Introduction

Cable TV networks, nowadays also known as Hybrid Fiber Coaxial (HFC) networks, were originally designed for broadcasting analogue TV signals downstream from a central Head End (HE) to the homes. Currently, they are being upgraded to enable the provision of bi-directional, digital communication services between the home and the rest of the world. Besides the installation of the return amplifiers in the upstream path, from the homes to the HE, and active network terminations or network terminals (NT), which separate this public network from the in-house private network, are required to provide the necessary functionality for the support of these services. In the meantime, several standards are becoming available to ensure the interoperability between a HE and a, possibly multi-vendor, set of NTs. The three major ones are DAVIC/DVB [5], IEEE 802.14 [8] and DOCSIS 1.1 [11].

[**]Currently Director of MathMetrics

[††]Also with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974, USA

The allocation of a single broadcast communication channel among a large number of independent users requires more advanced Medium Access Control (MAC) protocols than Time-Division Multiple Access (TDMA). The reason is that TDMA provides low performance with respect to channel utilization, unless all the users are transmitting continuously. It provides low access delay if and only if there are few users accessing the channel. In 1970 the ALOHA protocol [1] was introduced, which provided random access to the channel. The concept of random access implies that two or more users may want to use the channel at the same time, prohibiting error-free reception. If a collision occurs, the users may try again later, each one after a randomly chosen time. However, the performance of the ALOHA scheme becomes very poor when the channel occupancy increases beyond a certain level. Basically, there are two approaches to improve the performance of MAC protocols: carrier sense multiple access [9], and collision resolution algorithms [4], [14].

A given access network contains a finite number of users or customers, where this number typically ranges from ten to a few thousand. All users share the same up- and downstream channel. From the users' point of view, the HE can be seen as their common connection with the outer world. Furthermore, the HE controls the traffic within the access network. The structure of an access network is schematically depicted in Figure 1. An individual user or customer is also indicated as NT.

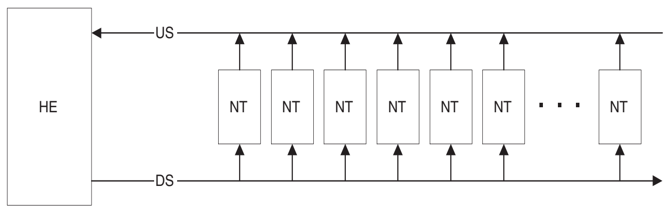

Fig. 1. Schematic overview of an HFC access network.

When a user wants to transmit data over the upstream channel, in general the following sequential *three-step* procedure is followed:

1) At a certain moment the user transmits a *request* to the HE for using a particular amount of upstream channel

time.
2) The user waits for feedback from the HE. The HE will eventually indicate when the user's data is allowed to be transmitted over the upstream channel.
3) The user transmits the data at the time indicated by the HE. In the meantime, it 'virtually' waits in the so-called data queue.

In most access networks, the requests mentioned in the first step can only be transmitted at specific times. The HE indicates to a group of users when a request can be transmitted. When more than one user sends a request at the same time, collisions occur. To resolve these collisions, a so-called collision resolution procedure is used.

To describe how these procedures work, it is necessary to discuss the structure of the upstream channel first. The upstream communication channel is divided into time slots of fixed length. The length of a time slot corresponds to the time required to send a 64-byte data packet. There are typically two types of slots: data slots and contention slots.

- A contention slot consists of three mini-slots. Such a slot is used by users to transmit requests only. So when the user wants to transmit data, it first transmits requests that are smaller than the actual data. This is the reason mini-slots were devised. When a collision occurs, this collision will be resolved by retransmitting the requests using a tree resolution algorithm that will be explained below. The retransmission of the requests will again take place during contention slots. There are typically two types of contention slots: ALOHA slots and tree slots, corresponding to two different contention resolution procedures. Only the tree slots will be discussed further now. For more information on the ALOHA protocol, we refer to [1], the first publication on this subject.
- A data slot, also known as a *reservation slot*, is 64-byte in length. Such a slot is exclusively used for transmitting data of a particular user.

The slots are organized in frames, which typically comprise 18 slots. The HE determines at the beginning of each frame the type of each slot in the upcoming frame. In Figure 2, this frame structure is illustrated.

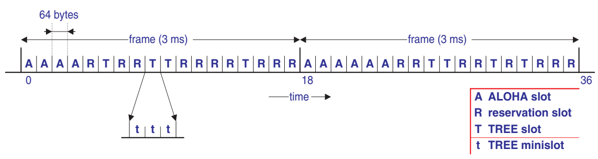

Fig. 2. Schematic representation of the frame structure.

In summary, the transmission of data by an individual user consists of two stages. First a contention stage takes place in which the individual user has to compete with other stations for exclusive data slots. Once a request has been successfully received by the HE, a customer enters the second stage. In this stage, a customer joins the data queue and finally occupies a certain number of data slots.

As Figure 2 shows, these two stages are run in parallel from an overall viewpoint (and not from the viewpoint of an individual customer). A certain number of slots are used for contention resolution, while the other slots are reserved for the individual transmission of data.

In the present paper we describe and evaluate a novel mechanism for contention resolution, which was first proposed and analyzed in [3], [13]. This algorithm reserves in each frame exactly one slot for new arrivals that wish to access the channel using contention resolution, and at least one slot for resolving their contention if there was one in the arrival slot. It is shown through analysis and simulation that the novel algorithm outperforms existing procedures.

The remainder of the paper is organized as follows. In Section 2 we describe the basics of some commonly used contention resolution procedures. We discuss some relevant performance measures in Section 3. In Section 4 we establish a relationship between the static arrival slot mechanism and a periodic $Geo/G/1$ queue. In Section 5, we derive a functional equation for the generating function of the service time distribution of a super customer, which we use in Sections 6 and 7 to determine the first two moments of the service time. We provide a capacity analysis in Section 8. In Section 9, we analyze the waiting-time distribution of a super customer, and give an expression for the mean waiting time in terms of the first two moments of the service time. We determine the mean service time of individual customers in Section 10. In Section 11 we present some numerical results comparing the static arrival slot mechanism with two other contention resolution procedures. Section 12 concludes the paper.

II. Contention resolution procedures

To describe the basics behind the most frequently used contention resolution procedures, it is useful to ignore all data slots and think of all slots as being contention slots. The basic principle of contention resolution is called *ternary tree* and will be explained now in more detail. Assume that the HE allows a certain group of customers to send a request for data slots during a given contention slot. Each customer individually flips a three-sided coin with equal probabilities for each side in order to determine in which of the three mini-slots it will send its request (if it wants to send a request). After this coin-flipping, every customer sends its request accordingly. There are three possible outcomes:

1) A mini-slot is empty.
2) A mini-slot contains exactly one customer.
3) A mini-slot contains more than one customer.

For each of the mini-slots, the HE can determine after the current slot time, which of these three possible outcomes occurred. In case the mini-slot contains exactly one customer, the corresponding request can be processed and the customer will receive a message from the HE, which prescribes when to use the upstream channel for the transmission of data. In case the mini-slot contains more than one customer, the HE can only conclude that there was a collision. The communication between the HE and NT usually involves a round-trip propagation delay which is not explicitly considered in this paper. Collisions in different mini-slots are treated separately. For all customers involved in a certain collision,

0-7803-7753-2/03/$17.00 (C) 2003 IEEE

a new contention slot will be generated in the near future. This means that each of these customers has to retransmit its request in that certain (mini-)slot in the future, again using the coin-flipping procedure. The HE indicates which slot this will be by means of broadcasting a grant just before the generation of the contention slot. Due to the fact that there will be some processing time in between the transmission and the retransmission of requests, it is allowed for each customer to update his request for more data slots than in the previous request. This procedure of processing successful and retransmitting unsuccessful requests continues until all requests are processed. Such a tree will be called a *contention tree*. This is shown in Figure 3. For details the readers are referred to [2], [3], [4], [6], [7], [10], [12], [13], [14].

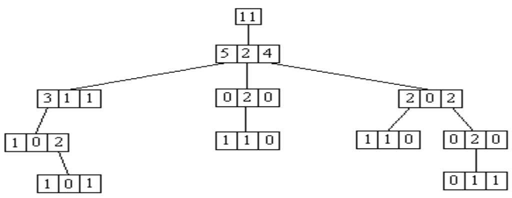

Fig. 3. Example of a ternary contention tree.

The basic idea of a ternary contention tree can be used in several ways to obtain slightly different contention resolution procedures. We now describe three different such contention resolution mechanisms. Two of the mechanisms are in fact well-known and are used in practice already. In the explanation of the contention resolution mechanisms, the transmission of individual data using data slots will be ignored. In a real access network, the majority of the slots is used for data transmission, but this is of no interest for the description of the contention resolution mechanism.

A. Gated mechanism

This contention resolution algorithm works as follows. Once a contention tree has been started, it is uninterruptedly being processed until termination. In the meantime, new customers may want to transmit a request, but there is no contention slot in which they are allowed to send their request because all the contention slots are dedicated to the tree completion. As soon as the tree terminates, *all* these waiting customers will enter in the first open slot. One can imagine this as if they did already arrive and have to wait (in front of a gate) for a new slot to enter the next tree, hence the term ‘gated’.

B. Non-blocked mechanism

This contention resolution algorithm is in some sense opposite to the gated mechanism. In the non-blocked mechanism, new individual users do not have to wait for the current tree to be terminated. In fact, they can enter the current tree directly when they arrive. This means that as soon as an individual user wants to send a request, it can send it in the next contention slot that occurs. So, in every slot that a tree is being processed, it may happen that there are new customers who also compete in that slot.

C. Static periodic arrival slot mechanism

Let s be a fixed integer, with $s \geq 1$. Suppose a tree is being processed at the moment. After every s contention slots, the current tree is interrupted and one slot is dedicated to new arrivals. Thus, requests that were generated during the previous period of s contention slots can be transmitted now for the first time. So every customer that ‘arrived’ during the previous s contention slots (and the data slots in between), now enters in this so-called *arrival slot*. After this slot, some of the new customers may be ‘lucky’. They leave the system immediately. The other customers can be considered as a tree that has been processed for just one slot. This tree (if there is one) is placed in a so-called tree queue (which is an example of a so-called *distributed queue*). In the next contention slot the interrupted tree is further being processed, exactly where it was interrupted. A tree consisting of several individual customers that is placed in the tree queue will be called a *super customer*. (Examples of super customers are $[2, 0, 4]$, $[4, 5, 3]$ and $[2, 0, 7]$; see Figure 4.) The super customers in the tree queue are served according to a First-Come First-Served discipline.

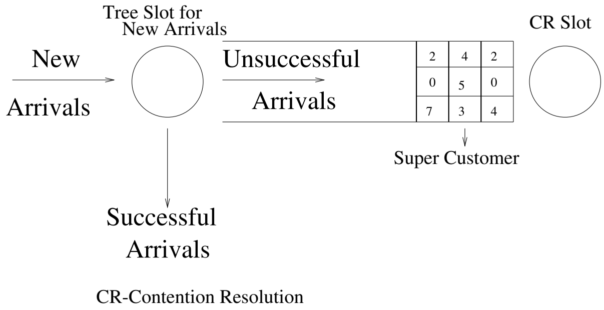

Fig. 4. Schematic representation of the queueing system.

III. IMPORTANT PERFORMANCE MEASURES

Relevant performance measures are the expectation and variance of the waiting time, the service time and the sojourn time of an individual customer and the capacity of the system. Observe that the service time depends on the actual tree mechanism: the order in which a contention tree is processed. Two commonly used methods are *breadth-first* and *depth-first*. The results we present are for depth-first only. Because we finally want to compare the various contention resolution mechanisms, we will now discuss the waiting time in more detail for all mechanisms. The waiting time consists of two parts. First a customer has to wait for an arrival slot to enter the system. This is called the ‘waiting-room time’.

- In the gated mechanism this can take a very long time.
- In the non-blocked mechanism this takes zero slots.
- In the static arrival-slot mechanism this takes at most s slots.

Second, a customer has to wait for his tree to be processed. This is called the ‘distributed-queue time’.

- In the gated mechanism this waiting time is zero.
- In the non-blocked mechanism this time is also zero.

- In the arrival-slot mechanism this waiting time can be zero if the customer is lucky, but in general this time is positive and corresponds to the waiting time in the distributed queue. It is a matter of definition whether the arrival slot in which the tree is formed is included in the waiting time or not. In the former case, the arrival slot is seen as a part of the service time and not of the waiting time.

The service time of an individual customer is defined as the total time that a customer 'stays' in the tree that it is involved in, only counting the slots that are dedicated to the tree. So interrupting arrival slots are not counted as service time. The sojourn time of a customer is defined as the sum of the waiting time and the service time.

A basic assumption in analyzing each of the mechanisms is that requests of users (customers) are generated according to a Poisson process with rate μ requests per slot. Thus, we ignore the fact that the population of users is finite, which is reasonable if their number is large.

IV. Relationship between static arrival slot mechanism and periodic $Geo/G/1$ queue

Our main goal is to analyze the waiting time and service time, together forming the sojourn time, of an individual customer in the static arrival slot mechanism. The waiting-room time of a customer is easily analyzed, since this random variable follows a uniform distribution over the s slots. A customer can be lucky and experience no distributed-queue time. The probability that a customer is lucky can easily be calculated. Analyzing the distributed-queue time for non-lucky customers is non-trivial, and requires the distribution of the service time of a super customer as will turn out. Therefore, we will start the analysis in the next section with the derivation of a functional equation that enables us to determine the generating function of the service time distribution of a super customer. This functional equation will be used in analyzing the individual service time as well.

Consider the static arrival slot mechanism and let us ignore the arrival slots. This has the consequence that arrival slots do not occupy time anymore. Lucky individual customers immediately leaving the system after their arrival slot are not seen anymore. Only super customers, that are put in the tree queue, remain. The resulting model can be seen as a special case of a periodic $Geo|G|1$-queue, where every s slots, with a constant probability, a (super) customer is placed in the queue.

Problem description

Consider a discrete-time (slotted) queue with discrete service times and a single server which serves the queue according to a First-Come First-Served discipline. Define a *frame* to be exactly s consecutive slots.

An arrival of a (super) customer can only occur just before the first slot of every frame: with (constant) probability α, there is an arrival just before the first slot of a frame, and with probability $1-\alpha$, there is no arrival. In case of an arrival, the (super) customer enters the queue and service continues normally during this first slot of every frame. The service time distribution of a super customer is allowed to be general. Both α and the service time distribution will be chosen such that the model correctly describes the arrival slot mechanism.

In the system with arrival slots, individual customers can only arrive in the first slot of every frame which has a length of $s+1$ slots. Since individual customers arrive according to a Poisson process with rate μ per slot, the number of customers contending in an arrival slot is Poisson distributed with mean $\lambda(s)$, where $\lambda(s) = (s+1)\mu$. This implies that α is given by

$$\alpha_{\lambda(s)} = 1 - e^{-\lambda(s)} \left(1 + \frac{\lambda(s)}{3}\right)^3;$$

the subscript is added to indicate that α depends on $\lambda(s)$.

Each super customer corresponds to the root of a so-called ternary contention tree. The service time of a super customer is the time needed (measured in slots) to complete this contention tree; it wil be determined in the next section.

V. Derivation of a functional equation

Let the random variable $\tilde{B}(n)$ represent the length of a tree which starts with n customers, $n \geq 2$. One can derive a recursive formula for the distribution of $\tilde{B}(n)$. The actual number of slots to complete a tree can be seen as 1 plus the number of slots to complete the three trees that have been formed in that initial slot. So, for all $n \geq 2$, the following recursion holds:

$$\begin{aligned} P(\tilde{B}(n) = k) &= \sum_{\substack{n_1,n_2,n_3 \geq 0 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3) \cdot \\ &\qquad P(\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) = k-1), \\ &\qquad k = 1,2,\ldots, \end{aligned}$$

with:

$$\begin{aligned} \tilde{B}(0) &= \tilde{B}(1) = 0, \\ P(\tilde{B}(n) = 0) &= 0, \qquad n = 2,3,\ldots, \\ \xi(n_1,n_2,n_3) &= \frac{n!}{n_1!n_2!n_3!} \left(\frac{1}{3}\right)^n. \end{aligned}$$

Let us introduce the following generating function:

$$\tilde{B}_n(z) := \sum_{k=0}^{\infty} P(\tilde{B}(n) = k) z^k, \qquad |z| \leq 1.$$

Multiplying both sides of the above equation with z^k and summing over $k = 1,2,\ldots$ yields:

$$\begin{aligned} \tilde{B}_n(z) &= z \sum_{k=1}^{\infty} \sum_{\substack{n_1,n_2,n_3 \geq 0 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} \left(\frac{1}{3}\right)^n \\ &\qquad P(\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) = k-1) z^{k-1}, \end{aligned}$$

and thus

$$3^n \frac{\tilde{B}_n(z)}{n!} = z \sum_{\substack{n_1,n_2,n_3 \geq 0 \\ n_1+n_2+n_3=n}} \frac{\tilde{B}_{n_1}(z)}{n_1!} \frac{\tilde{B}_{n_2}(z)}{n_2!} \frac{\tilde{B}_{n_3}(z)}{n_3!}.$$

Introducing $f_n(z) := \frac{\tilde{B}_n(z)}{n!}$ for $n \geq 0$ the last equation becomes:

$$3^n f_n(z) = z \sum_{\substack{n_1,n_2,n_3\geq 0\\ n_1+n_2+n_3=n}} f_{n_1}(z) f_{n_2}(z) f_{n_3}(z), \quad n \geq 2.$$

This equation can be used to obtain a closed-form expression for $f_n(z)$. To find $\mathbb{E}B(n)$ we can use the relation $\mathbb{E}B(n) = n! f_n'(1)$. A slightly more direct approach is to obtain a recursion for $\mathbb{E}B(n)$ by differentiating the above recursive equation for $\tilde{B}_n(z)$ with respect to z and substituting $z = 1$. Using $\tilde{B}_n'(1) = \mathbb{E}B(n)$, one obtains a recursion for $\mathbb{E}B(n)$. Multiplying both sides of the equation for $f_n(z)$ by x^n, we obtain:

$$(3x)^n f_n(z) = z \sum_{\substack{n_1,n_2,n_3\geq 0\\ n_1+n_2+n_3=n}} f_{n_1}(z) f_{n_2}(z) f_{n_3}(z) x^n, \ n \geq 2.$$

Summing these equations over $n \geq 2$ and introducing the following generating function:

$$F(x,z) := \sum_{n=0}^{\infty} f_n(z) x^n,$$

which is defined for all $z \leq 1$ and all x, we find:

$$F(3x,z) - 3xf_1(z) - f_0(z) = z\left(F^3(x,z) - f_0^3(z) - 3xf_0^2(z) f_1(z)\right).$$

Furthermore, we know:

$$f_0(z) = \frac{\tilde{B}_0(z)}{0!} = 1,$$
$$f_1(z) = \frac{\tilde{B}_1(z)}{1!} = 1.$$

Substituting this in the previous equation, we finally obtain the following functional equation:

$$F(3x,z) - 3x - 1 = z(F^3(x,z) - 1 - 3x),$$

or equivalently

$$F(3x,z) = zF^3(x,z) + (1+3x)(1-z).$$

Let the random variable $B_{\lambda(s)}$ denote the service time of an arbitrary super customer. Now we return to the arrival slot mechanism. Define $\pi_{\lambda(s)}(n)$ as the probability that an arriving batch of customers has size n, i.e., consists of n customers, given that a super customer is formed, $n \geq 2$. Then $B_{\lambda(s)}$ is distributed as follows:

$$B_{\lambda(s)} = \begin{cases} \tilde{B}(2) & \text{w.p. } \pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3), \\ \tilde{B}(3) & \text{w.p. } \frac{1}{7}\pi_{\lambda(s)}(3), \\ \tilde{B}(n) - 1 & \text{w.p. } \pi_{\lambda(s)}(n),\ n \geq 4. \end{cases}$$

Here $\pi_{\lambda(s)}(3)$ is the probability that an arriving batch of customers has size 3. A batch of customers of size 3 has 27 equi-probable possibilities to distribute itself over three mini-slots:

- 6 of these arrangements result in no super customer, because all 3 customers are lucky.
- 3 of these arrangements result in a super customer of size 3.
- The remaining 18 arrangements lead to a super customer of size 2.

This implies that given that a super customer is formed and the original batch has size 3, this results in a super customer of size 3 with probability $\frac{3}{27-6} = \frac{1}{7}$, and in one of size 2 with probability $\frac{6}{7}$. This explains the distribution of $B_{\lambda(s)}$ as given above. So:

$$\begin{aligned} P(B_{\lambda(s)} = 0) &= 0, \\ P(B_{\lambda(s)} = k) &= \sum_{n=4}^{\infty} \pi_{\lambda(s)}(n) P(\tilde{B}(n) = k+1) + \\ &\quad (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3)) P(\tilde{B}(2) = k) \\ &\quad + \frac{1}{7}\pi_{\lambda(s)}(3) P(\tilde{B}(3) = k), \quad k = 1, 2, \ldots \end{aligned}$$

Expressions for the $\pi_{\lambda(s)}(n)$'s are as follows:

$$\begin{aligned} \pi_{\lambda(s)}(2) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{(\lambda(s))^2}{2!} - 3\left(\frac{\lambda(s)}{3}\right)^2\right) \\ &= \frac{1}{6}\alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} (\lambda(s))^2, \\ \pi_{\lambda(s)}(3) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{\lambda(s)^3}{3!} - \left(\frac{\lambda(s)}{3}\right)^3\right) \\ &= \frac{7}{54}\alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} (\lambda(s))^3, \\ \pi_{\lambda(s)}(n) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \frac{(\lambda(s))^n}{n!}, \qquad n \geq 4. \end{aligned}$$

Now $B_{\lambda(s)}(z)$ will be expressed in terms of $F(x,z)$. Observing that $P(\tilde{B}(n) = 0) = P(\tilde{B}(n) = 1) = 0$ for $n \geq 4$, we find:

$$\begin{aligned} B_{\lambda(s)}(z) &= \sum_{k=1}^{\infty} P(B_{\lambda(s)} = k) z^k \\ &= \sum_{k=1}^{\infty}\sum_{n=4}^{\infty} \pi_{\lambda(s)}(n) P(\tilde{B}(n) = k+1) z^k \\ &\quad + \sum_{k=1}^{\infty} (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3)) P(\tilde{B}(2) = k) z^k \\ &\quad + \sum_{k=1}^{\infty} \frac{1}{7}\pi_{\lambda(s)}(3) P(\tilde{B}(3) = k) z^k \\ &= \sum_{n=4}^{\infty} \frac{1}{z}\pi_{\lambda(s)}(n) \tilde{B}_n(z) \\ &\quad + \pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3)) \tilde{B}_2(z) \\ &\quad + \frac{1}{7}\pi_{\lambda(s)}(3) \tilde{B}_3(z). \end{aligned}$$

Henceforth, we will use λ as shorthand for $\lambda(s)$. Plugging in the expressions for the $\pi_{\lambda(s)}(n)$'s, we obtain:

$$\begin{aligned} B_{\lambda}(z) &= \alpha_{\lambda}^{-1} e^{-\lambda} \Bigg(\frac{1}{z} \Big(\sum_{n=4}^{\infty} \lambda^n f_n(z) \Big) + \\ &\quad \Big(\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3 \Big) f_2(z) + \frac{1}{9}\lambda^3 f_3(z) \Bigg) \\ &= \alpha_{\lambda}^{-1} e^{-\lambda} \Big(\frac{1}{z} (F(\lambda,z) - f_0(z) - \lambda f_1(z) \end{aligned}$$

$$-\lambda^2 f_2(z) - \lambda^3 f_3(z)) +$$
$$+(\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3) f_2(z) + \frac{1}{9}\lambda^3 f_3(z)).$$

So, when it is possible to determine $F(\lambda(s), z)$, we have found a way of determining $B_{\lambda(s)}(z)$ for arbitrary z with $|z| \leq 1$.

VI. Determination of $\mathbb{E}B_{\lambda(s)}$

In the previous section we found a relation between $F(x, z)$ and $B_{\lambda(s)}(z)$. In this section $\mathbb{E}B_{\lambda(s)}$ will be expressed in terms of an infinite sum. As will turn out later, $\mathbb{E}B_{\lambda(s)}$ is needed for the evaluation of the mean waiting time of a super customer. This will be done by using the 2-dimensional functional equation, presented in Section V. After some algebraic manipulations we have:

$$\begin{aligned} \mathbb{E}B_\lambda &= B'_\lambda(1) \\ &= \alpha_\lambda^{-1} e^{-\lambda} \left(\frac{\partial}{\partial z} F(\lambda, 1) - \frac{3}{4}\lambda^2 - \frac{3}{8}\lambda^3 \right) \\ &\quad -\alpha_\lambda^{-1} e^{-\lambda} \left(e^\lambda - 1 - \lambda - \frac{1}{2}\lambda^2 - \frac{1}{6}\lambda^3 \right) \\ &\quad +\alpha_\lambda^{-1} e^{-\lambda} \left(\frac{1}{4}\lambda^2 + \frac{1}{6}\lambda^3 + \frac{1}{24}\lambda^3 \right) \\ &= \alpha_\lambda^{-1} e^{-\lambda} \left(\frac{\partial}{\partial z} F(\lambda, 1) - (e^\lambda - 1 - \lambda) \right). \end{aligned}$$

As it turns out, $\frac{\partial}{\partial z} F(\lambda, 1)$ can be represented as an infinite sum, which is easily computed numerically. This makes that $\mathbb{E}B_\lambda$ can be evaluated numerically. We now indicate how this can be done.

Define the generating function $\tilde{F}(x)$:

$$\tilde{F}(x) := \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{B}(n)}{n!} x^n, \qquad x \in \mathbb{R}.$$

This $\tilde{F}(x)$ is related to $F(x, z)$ as follows:

$$\begin{aligned} \tilde{F}(x) &= \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{B}(n)}{n!} x^n \\ &= \sum_{n=2}^{\infty} \frac{\tilde{B}'_n(1)}{n!} x^n \\ &= \frac{\partial F}{\partial z}(x, 1) - \frac{\tilde{B}'_0(1)}{0!} x^0 - \frac{\tilde{B}'_1(1)}{1!} x^1 \\ &= \frac{\partial F}{\partial z}(x, 1). \end{aligned}$$

So, if we take the (partial) derivative with respect to z on both sides of the (2-dimensional) functional equation for $F(x, z)$ and substitute $z = 1$, this gives the following 1-dimensional functional equation in $\tilde{F}(x)$:

$$\begin{aligned} \tilde{F}(3x) &= F^3(x, 1) + 1 \cdot 3F^2(x, 1)\tilde{F}(x) - 1 - 3x \\ &= e^{3x} - 3x - 1 + 3e^{2x}\tilde{F}(x) \\ &= e^{3x} - 3x - 1 + 3e^{2x} \left(e^x - x - 1 + 3e^{\frac{2}{3}x} \tilde{F}(\frac{1}{3}x) \right) \\ &= \ldots \end{aligned}$$

This functional equation can be solved by iteration. Continuing in this way yields the following:

$$\begin{aligned} \tilde{F}(3x) &= \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})} \left(e^{\frac{3x}{3^i}} - \frac{3x}{3^i} - 1 \right) + \\ &\quad \lim_{n\to\infty} \left[\tilde{F}(\frac{x}{3^n}) \prod_{i=0}^{n} 3e^{\frac{2x}{3^i}} \right]. \end{aligned}$$

We will now show that this expression converges. First, $\tilde{F}(\frac{x}{3^n})$ is investigated. Because of the fact that $\tilde{F}(0) = \tilde{F}'(0) = 0$, the following holds:

$$\tilde{F}(\frac{x}{3^n}) = \frac{x^2}{3^{2n}} \frac{\tilde{F}''(0)}{2} + \mathcal{O}\left(\frac{x^3}{3^{3n}}\right).$$

This implies that the limit, that appears in the expression for $\tilde{F}(3x)$ converges. For given x, $\left(e^{\frac{3x}{3^i}} - \frac{3x}{3^i} - 1\right)$ is of order $\mathcal{O}\left(\frac{1}{3^{2i}}\right)$, as can be verified by Taylor expansion of $e^{\frac{3x}{3^i}}$. This implies that the summand is of order $\mathcal{O}\left(\frac{1}{3^i}\right)$, which means that the sum converges geometrically fast as $i \to \infty$. The resulting formula for $\mathbb{E}B_\lambda$ becomes:

$$\mathbb{E}B_\lambda = B'_\lambda(1) = \alpha_\lambda^{-1} e^{-\lambda} \left(\tilde{F}(\lambda) - (e^\lambda - 1 - \lambda) \right).$$

VII. Determination of $\mathbb{E}B^2_{\lambda(s)}$

In the previous section we presented a method to determine $B'_{\lambda(s)}(1) = \mathbb{E}B_{\lambda(s)}$. In this section the focus is on determining an expression for $B''_{\lambda(s)}(1)$, because this quantity is needed for the evaluation of $\mathbb{E}W_{\lambda(s)}$. We start again from the 2-dimensional functional equation for $F(x, z)$ presented in Section V. After some algebraic manipulations we have:

$$\begin{aligned} B''_\lambda(1) &= \alpha_\lambda^{-1} e^{-\lambda} \Bigg(\frac{\partial^2}{\partial z^2} F(\lambda, 1) - 2\frac{\partial}{\partial z} F(\lambda, 1) + \\ &\quad 2(e^\lambda - 1 - \lambda) + \frac{1}{12}\lambda^2 \Bigg). \end{aligned}$$

As we have already seen, $\frac{\partial}{\partial z} F(\lambda, 1)$ can be represented as an infinite sum. Thus, we are now interested in finding an expression for $\frac{\partial^2}{\partial z^2} F(\lambda, 1)$ which can easily be computed. We will show how this can be done.

Define the following function $\tilde{C}(x)$:

$$\tilde{C}(x) := \frac{\partial^2}{\partial z^2} F(x, 1).$$

We start by taking the second (partial) derivative with respect to z on both sides of the (2-dimensional) functional equation and substitute $z = 1$, this gives the following 1-dimensional functional equation in $\tilde{C}(x)$:

$$\begin{aligned} \tilde{C}(3x) &= 0 + 2 \cdot 1 \cdot 3F^2(x, 1)\tilde{F}(x) + \\ &\quad 1 \cdot \left(6F(x, 1)\tilde{F}^2(x) + 3F^2(x, 1)\tilde{C}(x) \right) \\ &= 6e^{2x}\tilde{F}(x) + 6e^x\tilde{F}^2(x) + 3e^{2x}\tilde{C}(x). \end{aligned}$$

This functional equation for $\tilde{C}(3x)$ can be solved by iteration. (Because of page restrictions we omit the details.) The resulting expression is obtained as follows:

$$\begin{aligned}\tilde{C}(3x) &= 6e^{2x}\tilde{F}(x) + 6e^{x}\tilde{F}^2(x) + \\ &\quad 3e^{2x}\left(6e^{\frac{2x}{3}}\tilde{F}(\frac{x}{3}) + 6e^{\frac{x}{3}}\tilde{F}^2(\frac{x}{3}) + 3e^{\frac{2x}{3}}\tilde{C}(\frac{x}{3})\right) \\ &= \dots \\ &= \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})}\left(6e^{\frac{2x}{3^i}}\tilde{F}(\frac{x}{3^i}) + 6e^{\frac{x}{3^i}}\tilde{F}^2(\frac{x}{3^i})\right) + \\ &\quad \lim_{n\to\infty}\left[\tilde{C}(\frac{x}{3^n})\prod_{i=0}^{n} 3e^{\frac{2x}{3^i}}\right].\end{aligned}$$

One can verify, analogously as demonstrated before, that the above sum converges and that the limit vanishes. So this gives a method to evaluate $\tilde{C}(3x)$ because of the fact that $\tilde{F}(x)$ also can be evaluated accurately. The final expression for $B''_\lambda(1)$ becomes:

$$B''_\lambda(1) = \alpha_\lambda^{-1}e^{-\lambda}\left(\tilde{C}(\lambda) - 2\tilde{F}(\lambda) + 2(e^\lambda - 1 - \lambda) + \frac{1}{12}\lambda^2\right).$$

Now we present two graphs which compare the numerical results with simulation results. The first graph in Figure 5 shows $\mathbb{E}B_\lambda$ as a function of μ. The second graph in Figure 6 shows $\mathbb{E}B_\lambda^2$ as a function of μ. In both graphs, the simulation results are plotted as stars. As one can see, the stars are almost on the curve, which demonstrates the agreement between analysis and simulation.

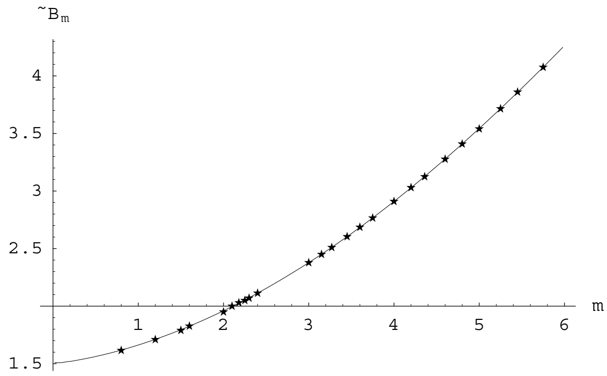

Fig. 5. Plot of $\mathbb{E}B_\lambda$ against μ.

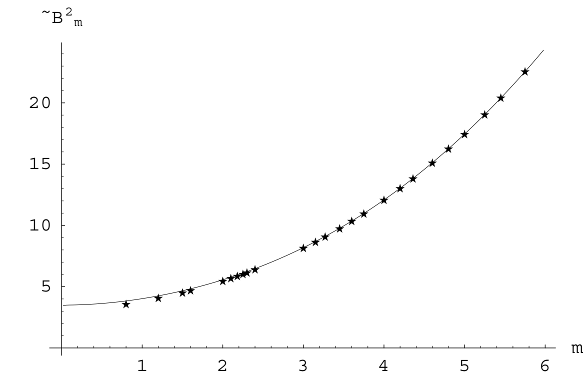

Fig. 6. Plot of $\mathbb{E}B_\lambda^2$ against μ.

VIII. Channel capacity analysis of arrival slot mechanism

In this section the capacity of the queuing system with $s+1$ slots per frame with the first being an arrival slot is determined numerically. By this we mean that the minimal rate $\mu_{\max}$ per slot of the individual arrival process is determined for which the system is not stable. This is of course equivalent with determining the rate $\lambda(s)_{\max}$ because $\lambda(s)_{\max} = (s+1)\mu_{\max}$. To determine the capacity of the system, it is relevant to see how the rate $\lambda(s)$ and the arrival probability α are related. So far, we have only looked at ternary contention trees. The relation between $\lambda(s)$ and α can be easily obtained for q-ary contention trees.

$$\begin{aligned}1-\alpha_{\lambda(s)} &= P(\text{no super customer arrives}) \\ &= P(\text{no collisions in all of the } q \text{ mini-slots }) \\ &= \sum_{k=0}^{q} P(\text{no collisions}|\#\text{ individual arrivals} = k) \\ &\quad P(\#\text{ individual arrivals} = k) \\ &= \sum_{k=0}^{q}\left(\frac{\binom{q}{k}k!}{q^k}\right)e^{-\lambda(s)}\frac{(\lambda(s))^k}{k!} \\ &= \sum_{k=0}^{q} e^{-\lambda(s)}\binom{q}{k}(\frac{\lambda(s)}{q})^k \\ &= e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q,\end{aligned}$$

so

$$\alpha_{\lambda(s)} = 1 - e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q.$$

Now that this relation is obtained, we return to the stability condition:

$$\alpha\mathbb{E}B = \alpha B'(1) < s,$$

or in this case

$$\begin{aligned}\alpha_{\lambda(s)}\mathbb{E}B_{\lambda(s)} &= \alpha_{\lambda(s)}B'_{\lambda(s)}(1) \\ &= \left(1 - e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q\right)B'_{\lambda(s)}(1) < s.\end{aligned}$$

We now present the capacity results for q-ary trees for the three mechanisms outlined. Mathematical results for the gated and the non-blocked system may be found in [3]; here we only present numerical results for these two mechanisms for comparison with the arrival slot mechanism. Because of the fact that in case of the gated mechanism, the capacity is very close to $\log q$, we have used that value in Table I.

When looking at Table I, there are a few remarkable observations. When $q = 4$, the non-blocked mechanism really outperforms the other mechanisms. When $q = 2$ or $q = 3$, the mechanism with $s = 2$ has the highest capacity of all mechanisms. Furthermore, the gated mechanism reaches a maximum capacity for $q = 3$. In fact, this maximum is over *all* values of q. It is also true that the case $q = 3$ dominates the case $q = 4$ for all studied mechanisms. But for binary trees some mechanisms have a higher capacity and some have a lower capacity than their ternary equivalent. An interesting question is whether there is a non-integer s for which the

Mechanism	$q=2$ $\lambda_{\max}$	$q=3$ $\lambda_{\max}$	$q=4$ $\lambda_{\max}$
Gated	0.3466	0.3662	0.3466
Non-blocked	0.360	0.40	0.40
$s=1$	0.420	0.4012	0.368
$s=2$	0.427	0.4132	0.378
$s=3$	0.419	0.4080	0.374
$s=4$	0.410	0.4017	0.369
$s=20$	0.363	0.3753	0.352
$s=100$	0.350	0.3680	0.348
$s=2000$	0.347	0.3662	0.347

TABLE I

CAPACITY RESULTS IN CASE OF q-ARY CONTENTION TREES

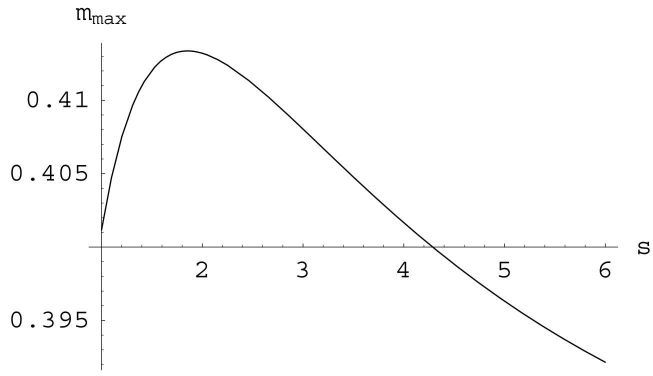

Fig. 7. Capacity as a function of s.

capacity is slightly higher than for $s = 2$. This is studied next.

Capacity in case of a non-integer s

We now proceed to discuss capacity results in case s is not restricted to a positive integer value. At first sight, a mechanism with a non-integer s seems perhaps impractical, due to the fact that it does not represent a 'real' algorithm. But it gives at least an indication of the optimality of the system with $s = 2$. There is practical relevance for mechanisms with non-integer values of s in the sense that they can indeed be interpreted as real mechanisms in the average sense. If the value of $s = 1.8$, then we can conclude that the number of contention slots in 10 frames is 18 or in 100 frames in 180. As far as the formulas are concerned, there is no reason to restrict ourselves to integer values of s. Therefore, we will present a plot, given in Figure 7, with the capacity given as a function of s. This plot is obtained as follows. In Section VIII, the stability condition for an arrival slot mechanism was presented. This condition was the following:

$$\left(1 - e^{-\lambda(s)}\left(1 + \frac{\lambda(s)}{q}\right)^q\right) B'_{\lambda(s)}(1) < s.$$

If we replace the inequality sign by an equality sign, then we obtain an equation. If we fix s to some real value and subsequently solve this equation numerically for $\lambda(s)$, or equivalently because s is fixed, for μ, then we obtain $\mu_{\max}$ for that particular value of s.

As one can see, the function reaches an optimum for $s \approx 1.8$. This is a rather positive result for the moment, because 1.8 is relatively close to 2, and the optimal capacity is close to the capacity of the system with $s = 2$. Furthermore, one can see that the function is decreasing after the optimum is reached.

IX. WAITING TIME ANALYSIS OF THE PERIODIC $Geo/G/1$ QUEUE

Let the random variable W denote the waiting time of a (super) customer. In this section, an expression for the generating function of W is derived. Next, it will be indicated how the, yet unknown, stationary probabilities in this expression can be found. An important remark is that the waiting time of an arbitrary (super) customer is the same as the total amount of work just before the first slot of a frame by virtue of the BASTA (Bernoulli Arrivals See Time Averages) property; see [15]. So we can also treat W as a workload variable, as will be done below.

Consider the discrete Markov chain defined by the total amount of work at the queue just before the first slot of every frame. This total amount of work is defined as the total time it takes to complete service of all the customers that are currently in the system. A possible new arrival is not counted yet. The stationary distribution satisfies the following balance equations:

$$\begin{aligned} w_0 &= \alpha \sum_{n=0}^{s} \sum_{k=1}^{s-n} b_k w_n + (1-\alpha) \sum_{n=0}^{s} w_n \\ &= \alpha \sum_{n=0}^{0} b_{s+n} w_{-n} + \alpha \sum_{n=1}^{s-1} b_{s-n} w_n + (1-\alpha) w_s + \\ &\quad \alpha \sum_{n=0}^{s-1} \sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha) \sum_{n=0}^{s-1} w_n, \\ w_k &= \alpha \sum_{n=0}^{k} b_{s+n} w_{k-n} + \alpha \sum_{n=1}^{s-1} b_{s-n} w_{k+n} + (1-\alpha) w_{k+s}, \end{aligned}$$

where $b_k = P[B = k]$, $w_k = P[W = k]$, $k = 1, 2, 3, \ldots$. Here the random variable B is the service time of the arrived customer and W is the amount of work seen by (or waiting time of) the arrived customer. The balance equation for w_0 is rewritten so that the right-hand side is of nearly the same form as the right-hand side of the general balance equation for w_k for $k = 1, 2, \ldots$. The general balance equation can be explained as follows. Consider all possible transitions from a state j to the current state $k \geq 1$. When the system is currently in state $k \geq 1$, the previous frame took s slots and so the total amount of work just before the beginning of that previous frame was at most $k + s$. This implies that $0 \leq j \leq k + s$. The total amount of work that arrived at the beginning of the previous frame plus j, the total amount of work that was already present, minus s must be equal to k to ensure a transition from j to k for $k \geq 1$.

Multiplying both sides of each equation with z^k and summing all equations for $k = 0, 1, 2, \ldots$ leads to the following equation:

$$\begin{aligned}
W(z) &= \alpha \sum_{k=0}^{\infty} \sum_{n=0}^{k} b_{s+n} w_{k-n} z^k + \alpha \sum_{k=0}^{\infty} \sum_{n=1}^{s-1} b_{s-n} w_{k+n} z^k \\
&\quad + (1-\alpha) \sum_{k=0}^{\infty} w_{k+s} z^k + \alpha \sum_{n=0}^{s-1} \sum_{k=1}^{s-1-n} b_k w_n \\
&\quad + (1-\alpha) \sum_{n=0}^{s-1} w_n \\
&= W(z) z^{-s} \alpha B(z) \\
&\quad - z^{-s} \alpha \sum_{k=0}^{s-2} w_k z^k \sum_{n=k+1}^{s-1} b_{s-n} z^{s-n} \\
&\quad + z^{-s}(1-\alpha) \left(W(z) - \sum_{k=0}^{s-1} w_k z^k \right) \\
&\quad + \alpha \sum_{n=0}^{s-1} \sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha) \sum_{n=0}^{s-1} w_n.
\end{aligned}$$

After some algebraic manipulation, the above equation yields the following expression for $W(z)$:

$$W(z) = \frac{\alpha \sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k (z^s - z^{k+n}) + (1-\alpha) \sum_{k=0}^{s-1} w_k (z^s - z^k)}{z^s - \alpha B(z) - (1-\alpha)}$$

In this expression one can see that for $z = 1$ the numerator is zero. Therefore, the numerator can be factorized, leading to the following expression for $W(z)$:

$$\frac{(z-1)\left(\alpha \sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} z^i + (1-\alpha) \sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} z^i \right)}{z^s - \alpha B(z) - (1-\alpha)}.$$

In this expression there are still s unknowns: $w_0, w_1, \dots, w_{s-1}$. Below, it will be explained how these yet unknown probabilities can be found, but first an analytic expression for $\mathbb{E}W = W'(1)$ will be given. It can be derived using either Taylor expansion or using l'Hôpital's rule to give the following:

$$\begin{aligned}
\mathbb{E}W &= \frac{\alpha (\sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} i)}{s - \alpha B'(1)} + \\
&\quad \frac{(1-\alpha)(\sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} i)}{s - \alpha B'(1)} - \\
&\quad \frac{(\frac{s(s-1)}{2} - \alpha \frac{B''(1)}{2})}{s - \alpha B'(1)}.
\end{aligned}$$

As mentioned earlier, the first two moments of the service time distribution of a super customer are needed in order to evaluate this expression in case of the static arrival slot mechanism.

Determination of $w_0, w_1, \dots, w_{s-1}$

We know that $W(z)$ is well-defined for z with $|z| \le 1$. Using Rouché's theorem, it may be shown that the denominator of the final expression for $W(z)$ has s zeros $z_1, z_2, \dots, z_s$ that lie on or within the (complex) unit circle. (We omit the detailed arguments because of space limitations.) For these zeros $W(z)$ is well-defined and consequently the numerator has to be zero for these $z_1, z_2, \dots, z_s$ as well. This gives s equations. It can easily be shown that for $z = 1$ both the numerator and denominator always equal zero, independent of the values of the w_k's and the b_k's. So in fact there are at most $s-1$ equations that give information about the w_k's. But there is also a normalization condition: $W(1) = 1$. When we assume that there are indeed $s-1$ different zeros which lead to $s-1$ different equations, we do have, together with the boundary condition, enough equations to solve for $w_0, w_1, \dots, w_{s-1}$. It might happen that there are less than $s-1$ different zeros, but in that case some zeros have a multiplicity that is larger than one. In this case the numerator must also have that zero with the same multiplicity, which gives us again the right number of equations. Hence we have s equations for s unknowns $w_0, \dots, w_{s-1}$. This set of equations has a unique solution, which follows from the general theory of Markov chains that under the condition of stability, there is a unique stationary distribution and thus also a unique solution $W(z)$.

X. MEAN SERVICE TIME OF INDIVIDUAL CUSTOMERS

The $Geo/G/1$ queue yields results for super customers, i.e., for batches of customers. The waiting time for an individual customer is of course the same as the one for a super customer, but its service time is different. As soon as a customer is successful, it will not wait till the whole contention tree is finished, but will leave immediately. The service time of an individual customer also depends on the order in which the contention tree is processed. In this section we only present the analysis of the mean individual service time under the depth-first order; for more details and results the readers are referred to [3], where it is also shown that the service times for breadth-first and depth-first are very close.

Define the random variable $\tilde{S}(n)$ as the *total delay* of all customers in a tree with n customers, when it is processed in depth-first order, $n \ge 2$. Recall that the random variable $\tilde{B}(n)$ is defined as the *total length* of a tree with n customers in it, $n \ge 2$. By conditioning on the outcome in the first contention slot we obtain the following equation for $\mathbb{E}\tilde{S}(n)$:

$$\begin{aligned}
\mathbb{E}\tilde{S}(n) &= \sum_{\substack{n_1, n_2 \neq 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \cdot \\
&\quad \left(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_2) + \mathbb{E}\tilde{S}(n_3) + (n_2+n_3)\mathbb{E}\tilde{B}(n_1) + n_3 \mathbb{E}\tilde{B}(n_2) \right) \\
&\quad + \sum_{\substack{n_1, n_2 = 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_3) + n_3 \mathbb{E}\tilde{B}(n_1) \right) \\
&\quad + \sum_{\substack{n_1, n_2 \neq 1, n_3 = 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_2) + n_2 \mathbb{E}\tilde{B}(n_1) \right) \\
&\quad + \sum_{\substack{n_1, n_2 = 1, n_3 = 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1) \right) + n, \qquad n \ge 2.
\end{aligned}$$

After some manipulations this can be rewritten as:

$$
\begin{aligned}
\frac{3^n}{n!}\mathbb{E}\tilde{S}(n) \quad = \quad & \sum_{\substack{n_1,n_2,n_3\\ n_1+n_2+n_3=n}} \left(\frac{\mathbb{E}\tilde{S}(n_1)}{n_1!}\frac{1}{n_2!}\frac{1}{n_3!} \right. \\
& \left. + \frac{\mathbb{E}\tilde{S}(n_2)}{n_2!}\frac{1}{n_1!}\frac{1}{n_3!} + \frac{\mathbb{E}\tilde{S}(n_3)}{n_3!}\frac{1}{n_1!}\frac{1}{n_2!} \right) \\
& + n\frac{3^n}{n!} + 3 \sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \frac{1}{n_1!}\frac{1}{n_2!}\frac{1}{n_3!} n_2 \mathbb{E}\tilde{B}(n_1), \\
& n \geq 2.
\end{aligned}
$$

Define the function $Q(x) := \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{S}(n)}{n!} x^n$. After multiplying both sides of the last equation with x^n and summing these equations over n, we find the following result:

$$
\begin{aligned}
Q(3x) \quad = \quad & 3e^{2x}Q(x) + \sum_{n=2}^{\infty} n\frac{3^n}{n!}x^n \\
& + 3\sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!} n_2 \mathbb{E}\tilde{B}_{n_1}.
\end{aligned}
$$

Substituting

$$
\sum_{n=2}^{\infty} n\frac{(3x)^n}{n!} = 3x\left(e^{3x}-1\right)
$$

and

$$
\begin{aligned}
& \sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!} n_2 \mathbb{E}\tilde{B}(n_1) \\
& \quad = \sum_{n_1=0}^{\infty} \frac{\mathbb{E}\tilde{B}(n_1)}{n_1!}x^{n_1} \sum_{n_2=2}^{\infty} n_2\frac{x^{n_2}}{n_2!} \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} \\
& \quad = \tilde{F}(x) \cdot x\left(e^x - 1\right) \cdot e^x,
\end{aligned}
$$

yields

$$
Q(3x) = 3e^{2x}Q(x) + 3x\left(e^{3x}-1\right) + 3xe^x\left(e^x-1\right)\tilde{F}(x).
$$

This equation can be solved by iteration. This leads to:

$$
\begin{aligned}
Q(3x) \quad = \quad & \sum_{j=0}^{\infty} 3^j e^{3x(1-\frac{1}{3^j})} \frac{3x}{3^j} \left(e^{\frac{3x}{3^j}} - 1\right) \\
& + \sum_{j=0}^{\infty} 3^j e^{3x(1-\frac{1}{3^j})} \frac{3x}{3^j} e^{\frac{x}{3^j}} \left(e^{\frac{x}{3^j}} - 1\right) \tilde{F}(\frac{x}{3^j}) \\
= \quad & \sum_{j=0}^{\infty} 3xe^{3x(1-\frac{1}{3^j})} \left(e^{\frac{3x}{3^j}} - 1\right) \\
& + \sum_{j=0}^{\infty}\sum_{i=0}^{\infty} 3^{i+1} x e^{x(3-\frac{1}{3^j}-\frac{1}{3^{i+j}})} \, . \\
& \quad \left(e^{\frac{x}{3^j}} - 1\right)\left(e^{\frac{x}{3^{i+j}}} - \frac{x}{3^{i+j}} - 1\right).
\end{aligned}
$$

Hence, $Q(x)$ can be readily computed by using this infinite sum (which converges very fast). Now we determine the mean individual service time. Let the random variable $\tilde{S}$ denote the total delay in an *arbitrary tree*. By conditioning on the size of the tree we obtain:

$$
\begin{aligned}
\mathbb{E}\tilde{S} \quad & = \quad e^{-\lambda(s)}\cdot 0 + \lambda(s)e^{-\lambda(s)}\cdot 1 + \sum_{n=2}^{\infty} e^{-\lambda(s)}\frac{(\lambda(s))^n}{n!}\mathbb{E}\tilde{S}(n) \\
& = \quad e^{-\lambda(s)}\left(\lambda(s) + Q\left(\lambda(s)\right)\right).
\end{aligned}
$$

The mean service time of an arbitrary individual customer, denoted by $\mathbb{E}\tilde{S}_{ind}$, can be obtained by dividing the mean total internal delay of an arbitrary tree by the mean number of customers in a tree. So the mean service time of an arbitrary individual customer satisfies the following relation:

$$
\begin{aligned}
\mathbb{E}\tilde{S}_{ind} \quad & = \quad \frac{\mathbb{E}\tilde{S}}{\lambda(s)} \\
& = \quad e^{-\lambda(s)}\frac{\lambda(s) + Q\left(\lambda(s)\right)}{\lambda(s)}.
\end{aligned}
$$

Now we nearly have all ingredients to compute the mean waiting time and the mean sojourn time of an individual customer (or user). So far we ignored the presence of arrival slots (i.e. batches of customers arrive instantaneously at the beginning of each frame of s slots). To compute the waiting times and sojourn times correctly, we have to insert the arrival slots again, and as a consequence, waiting times and sojourn times will become longer. This translation to the situation with arrival slots can be done by using the generating functions of the waiting time and the sojourn time; for details the readers are referred to [3].

XI. Waiting time and sojourn time results

In this section we compare the performance of the three contention resolution mechanisms. The performance characteristics of interest are the mean and variance of the waiting time and the sojourn time of individual customers. The comparison is based on simulation.

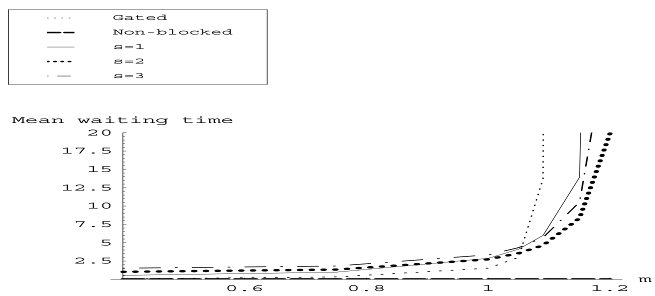

Fig. 8. Estimated mean waiting time as a function of μ.

When we examine Figure 8, we see that the gated mechanism performs very well in the light-traffic zone. In this regime, the mechanism with $s = 1$ shows good performance as well. This is intuitively clear because we have small trees or no trees generated in this regime. In the heavy-traffic regime, the capacity of the system is eventually the important factor that determines the performance of the system. Therefore, we see that the mechanism with $s = 2$ outperforms all the other mechanisms, except the non-blocked mechanism, in which individual customers experience, by definition, a waiting time

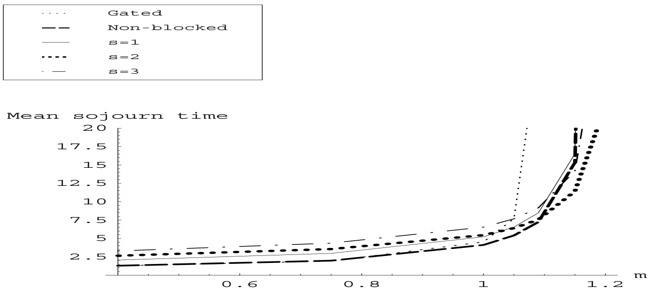

Fig. 9. Estimated mean sojourn time as a function of μ.

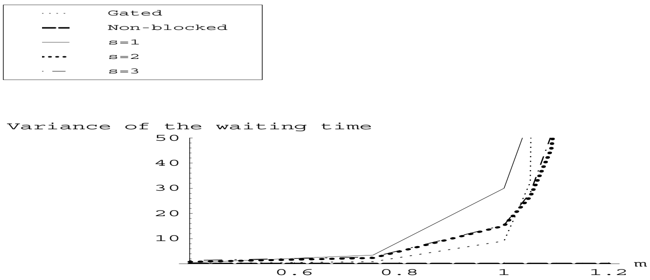

Fig. 10. Estimated variance of the waiting time as a function of μ.

of zero slots. The most interesting comparison between the different contention resolution procedures is made with respect to the mean sojourn time (of individual customers). In Figure 9 we see again that the non-blocked and gated mechanisms are superior to the static arrival slot mechanism in light traffic. In heavy traffic, the mechanisms are ordered with respect to their capacity. An interesting remark can be made on the variance of the service time in the gated mechanism compared with the static arrival slot mechanisms. The corresponding graph is given in Figure 10. In this graph, we see a considerably smaller variance for the gated mechanism as compared with the arrival slot mechanisms in the light-traffic zone. This can be explained as follows. In the gated mechanism, the number of customers that are involved in a tree is small most of the time. In the arrival slot mechanisms, the arrivals are accumulated over a period of $s+1$ slots. In light traffic, this period of $s+1$ slots will be longer than most gate periods. Consequently, there will be more variance in the number of customers involved in a tree, leading to larger variances of the service time. We see in fact a similar behavior as in the corresponding

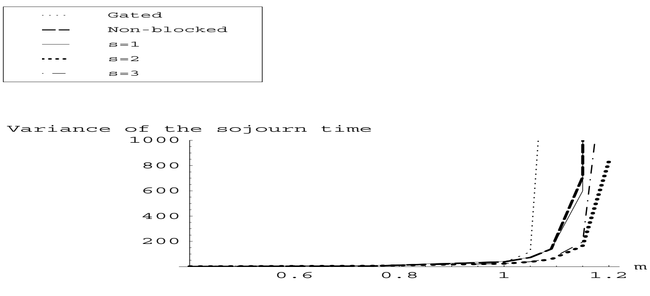

Fig. 11. Estimated variance of the sojourn time as a function of μ.

mean sojourn time graph. In the light-traffic zone, the gated and non-blocked mechanism outperform the static arrival slot mechanism. When the intensity μ increases, the capacity of the system becomes the dominant factor and determines the performance with respect to the variance of the sojourn time completely. This is confirmed in Figure 11.

XII. Conclusion

In this paper we proposed a novel mechanism for efficient contention resolution, and compared this algorithm with existing procedures available in the standards. It is demonstrated that the channel capacity is increased, while the delay of processing a NT's request is considerably lowered in a heavy-traffic regime. The main reason for the performance improvement is that there are small trees in the proposed system compared to the large trees that are there in the existing mechanisms under heavy-traffic conditions. The delay in a light-traffic regime can be reduced if we do not 'waste' the arrival slot when there are no arrivals. This is the key feature of the dynamic arrival slot mechanism which is described in greater detail in [3].

References

[1] N. Abramson, "The ALOHA system – Another alternative for computer communications", *Proc. 1970 Fall Joint Comput. Conf. AFIPS Conf.*, vol. 37, 1970.

[2] T. Berger, "The Poisson multiple-access conflict resolution problem", in *Multi-user communication systems*, G. Longo, Ed., CISM course and Lecture Notes 265 (Springer Verlag, New York, 1981), pp. 1–27.

[3] M. X. van den Broek, "On Contention Resolution Procedures – Queueing Analysis and Simulation", Master Thesis, Department of Mathematics & Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, 2001.

[4] J. I. Capetanakis, "Tree algorithms for packet broadcast channels", *IEEE Trans. Inform. Theory*, vol. 25, pp. 505–515, 1979.

[5] DVB, "Digital Video Broadcasting (DVB); DVB interaction channel for Cable-TV distribution systems (CATV)", working draft (version 2), March 1999, based on European Telecommunications Standard.

[6] R. G. Gallager, "A perspective on multiple access channels", *IEEE Trans. Inform. Theory*, vol. 31, pp. 124–142, 1985.

[7] B. Hajek, N. B. Likhanov, and B. S. Tsybakov, "On the delay in a multiple-access system with large propagation delay", *IEEE Trans. Inform. Theory*, vol. 40, pp. 1158–1166, 1994.

[8] IEEE 802.14 WG, "Cable-TV access method and physical layer specification", Draft 3, Revision 3, Oct. 1998.

[9] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part 1: CSMA modes and their throughput-delay characteristics", *IEEE Trans. Commun.*, vol. 23, pp. 1400–1416, 1975.

[10] J. L. Massey, "Collision resolution algorithms and random access algorithms", in *Multi-user communication systems*, G. Longo, Ed., CISM course and Lecture Notes 265 (Springer Verlag, New York, 1981), pp. 73–131.

[11] MCNS Holdings, "Data-over-cable service interface specifications, radio frequency interface specification, Ref. Nr. SP-RFIv1.1-PI01-990226, Feb. 1999.

[12] G. Ruget, "Some tools for the study of channel sharing algorithms", in *Multi-user communication systems*, G. Longo, Ed., CISM course and Lecture Notes 265 (Springer Verlag, New York, 1981), pp. 201–231.

[13] N. Sai Shankar, "A distributed queueing model for contention resolution Protocol in HFC access networks", in *Proc. Int. Conf. Inf. Comm. Signal Proc.*, Singapore, 2001.

[14] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback", *Probl. Info. Trans.*, vol. 14, pp. 259–280, 1978.

[15] R. W. Wolff, "Stochastic Process and the Theory of Queues", Prentice Hall, New York, 1989.