SICTA: A 0.693 Contention Tree Algorithm Using Successive Interference Cancellation

Yingqun Yu and Georgios B. Giannakis
Department of Electrical and Computer Engineering
University of Minnesota
200 Union Street SE
MN 55455, USA
{yyu, georgios}@ece.umn.edu

Abstract— **Contention tree algorithms have provable stability properties, and are known to achieve stable throughput as high as 0.487 for the infinite population Poisson model. A common feature in all these random access protocols is that collided packets at the receive-node are always discarded. In this paper, we derive a novel tree algorithm (TA) that we naturally term SICTA because it relies on successive interference cancellation to resolve collided packets. Performance metrics including throughput and delay are analyzed to establish that SICTA outperforms existing contention tree algorithms reaching 0.693 in stable throughput.**

I. Introduction

The medium access problem is present in nearly all communication and computer networks, especially when the channel is inherently broadcast and should be shared by many nodes, as in wired and wireless local area networks (LANs), wireless cellular networks (GSM, GPRS, 3G) and hybrid fiber coaxial networks (HFCs). Since multiple nodes may access the single channel simultaneously, the central problem is to coordinate the sending and receiving nodes to achieve the effective utilization. Depending on the channel types and source traffic characteristics, various solutions to the medium access problem have been proposed.

Fixed resource allocation protocols are contention-free because orthogonal or near orthogonal subchannels are assigned to each user. Typical examples are FDMA, TDMA and CDMA, which are implemented in the first, second and third generations of wireless cellular networks, respectively [1]. The main limitations of such protocols are low channel utilization and long access delay, when there is a large population and the data traffic is bursty [2]. For such cases, random access protocols are certainly more appropriate.

Broadly speaking, there are three categories of random multiple-access protocols: ALOHA [3], carrier-sense multiple access (CSMA) [4], and collision-resolution algorithms (CRA) [5], [6]. The ALOHA and CSMA protocols are common practice for bursty, computer generated traffic. However, they suffer from stability problems and low throughput. In comparison, CR protocols based on contention tree algorithms have provable stability properties; see e.g., [18]. The well-known First-come-first-serve (FCFS) tree algorithm is the most efficient one reaching throughput 0.487, when the traffic is Poisson distributed [8], [9], [10]; see also [18] for a thorough review. In recent years, with the interest for HFC and wireless broadband access networks growing, tree algorithms have received revived attention by several standards, including DAVIC/DVB [12], IEEE 802.14 [13] and DOCSIS 1.1 [14]. New channel access mechanisms for tree algorithms are explored to improve the performance under heavy traffic [15].

Note that all random access schemes rely on re-transmissions in order to resolve current collisions as well as minimize future collisions. Collided packets are typically discarded, which reduces throughput considerably. In contrast, two recent protocols, named Network Diversity Multiple Access (NDMA) and Blind NDMA (B-NDMA), do not discard collided packets but rely on proper re-transmissions and signal separation principles to resolve collisions [19], [20], [21].

Inspired by [19], [20], we derive here a novel contention tree algorithm, that we term SICTA, which also retains collided packets. SICTA is rooted on the simple idea that interference contains information about the packets we wish to resolve. Different from NDMA and B-NDMA which require estimations of the number of users involved in a collision and complex signal separation techniques, SICTA exploits the conventional contention tree structure and simply employs successive interference cancellation (SIC) to resolve collisions. In a standard tree algorithm (STA), all packets are resolved in an orderly one-by-one fashion according to the underlying tree structure. This nice property lends itself naturally to SIC, provided that collided packets are not discarded. To illustrate the underlying idea, consider the motivating example depicted in Fig. 1. Let $\mathbf{Y}_t$ denote the received signal vector at the end of slot t. From the received signal vector at the second slot, $\mathbf{Y}_2$, the receiver decodes the packet A, and recovers the transmitted signal vector $\mathbf{X}_A$ corresponding to the packet A. Subsequently, the interference $\mathbf{X}_A$ from packet A to packet B in the first slot is cancelled to obtain:

$$\widetilde{\mathbf{Y}}_1 = \mathbf{Y}_1 - \mathbf{X}_A. \tag{1}$$

Based on $\widetilde{\mathbf{Y}}_1$, the packet B can be recovered as well. Notice that unlike existing tree algorithms that require 3 slots to resolve this collision, only 2 slots suffice here.

The main advantage of SICTA is that it improves the throughput of a standard binary tree algorithm with gated

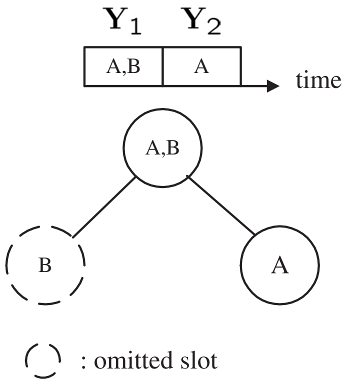

Fig. 1. A motivating SICTA example

access from 0.346 to 0.693, thus outperforming all known contention tree algorithms, including the celebrated 0.487 FCFS one.

The rest of the paper is organized as follows. Section II describes SICTA in detail. Section III analyzes statistics of the collision resolution interval (CRI), and derives the throughput. Section IV deals with delay analysis and methods to compute the average packet delay. Numerical results are given in Section V, and conclusions are summarized in Section VI.

II. Novel Random-Access Protocol

In this section, we develop our contention tree algorithm. After detailing the channel model, we outline the two basic CRAs, namely the standard tree algorithm (STA), and the modified tree algorithm (MTA), before focusing on the development of our SICTA.

We assume the following channel model for multiple access:

i) Infinite Population: An infinite number of independent users are transmitting to a common receiver packets of length equal to one time unit (slot), over a slotted-time channel.
ii) Poisson Arrivals: The packet arrival process is Poisson distributed with overall rate λ, and each packet arrives to a “new” user that has never been assigned a packet before.
iii) Collision or Perfect Reception: If only one user sends a packet during a certain slot, it is received with no errors. However, if more users send packets in the same slot, then collision occurs and no packet information can be extracted from this single collision. In STA and MTA, the collided packets are discarded, while in SICTA they are saved for future reuse.
iv) $0/k/e$ Immediate Feedback: By the end of each slot, users are informed of the feedback from the receiver immediately and errorlessly. The feedback is one of:
    a) idle (0): when no packet transmission is taking place,
    b) number of identified slots (k), where k is the number of decoded packets plus the number of slots identified as being left-idle – a notion that will become clear later; or,
    c) conflict (e): when no packet reception is successful.

For MTA, the feedback is ternary $0/1/e$, since only one ($k = 1$) successful packet reception is possible per time slot. STA on the other hand, requires only binary feedback: collsion/non-collision; that is, it does not differentiate between the feedback 1 and 0.

To form a complete random access protocol, a CRA must be combined with a distributed channel-access algorithm (CAA), which specifies when new packets may join a CRA. In this paper, we adopt a gated CAA, where new packets are transmitted in the first available slot after previous conflicts are resolved. The time interval from the slot where an initial collision occurs up to and including the slot in which all senders recognize that all packets involved in the collision have been successfully received, is called a collision resolution interval (CRI). Thus, new arrivals are inhibited from transmission during the CRI.

A. STA and MTA

In the gated access STA, new packets arriving during the previous CRI join the next CRI and are transmitted in the first available slot. If the receiver feeds back an idle or success, the CRI ends at this point. However, if the receiver feeds back a collision, each user tosses a two-sided coin and joins the first (right) subset with probability p or the second (left) subset with probability $1 - p$. The initial collision is resolved when both of these subsets are resolved and the first subset is always resolved first. This procedure continues recursively until all packets are received successfully. The best way to visualize this CR in STA is to use a binary tree with the root being the first slot of the CRI. The rooted binary tree structure in Fig. 2 represents a particular pattern of idles, successes and collisions resulting from such a sequence of splits. In this example, 7 slots are required to resolve the collisions of three packets A, B, and C. Let l_n be the CRI length given that n packets initially collide. For STA, we have

$$l_n = 1 + l_i + l_{n-i}, \quad n \geq 2, \tag{2}$$

where i is the number of users joining the right subset. It is known that fair splitting with $p = 1/2$ is optimal and results in 0.346 throughput [11], [16].

Notice that in Fig. 2, the collision in slot 3 is followed by an idle in slot 4, generating a deterministic collision in slot 5, since all collided packets involved in slot 3 are assigned to the second subset. An improvement results if we do not transmit over this second subset and proceed directly to the next level of the binary tree. This algorithm is known as modified tree algorithm (MTA) [11]. As it is shown in Fig. 3, only 6 slots are required to resolve the collision for the same tree structure of Fig. 2. Using the same definition of l_n as in (2), we have

$$l_n = \begin{cases} 1 + l_i + l_{n-i}, & \text{if } 1 \leq i \leq n \\ l_i + l_{n-i} (= 1 + l_n), & \text{if } i = 0 \end{cases}, \quad n \geq 2. \tag{3}$$

Relative to (2), the slot for transmitting the left subset is omitted when the right subset is empty ($i = 0$).

This simple modification increases the throughput from 0.346 to 0.375 [11], [16]. However, for binary MTA a biased

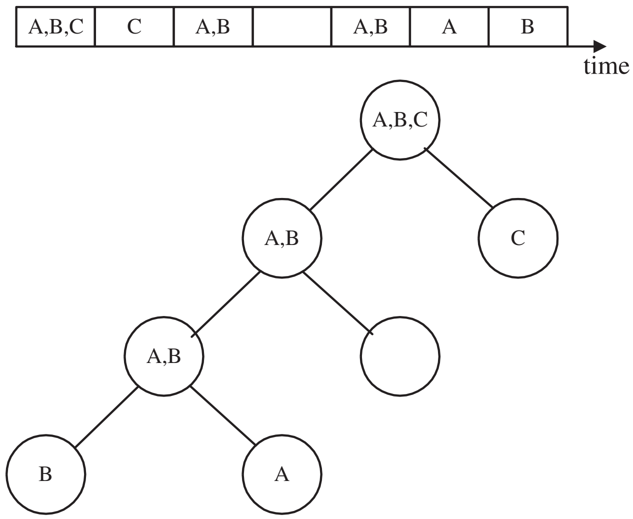

Fig. 2. An STA example

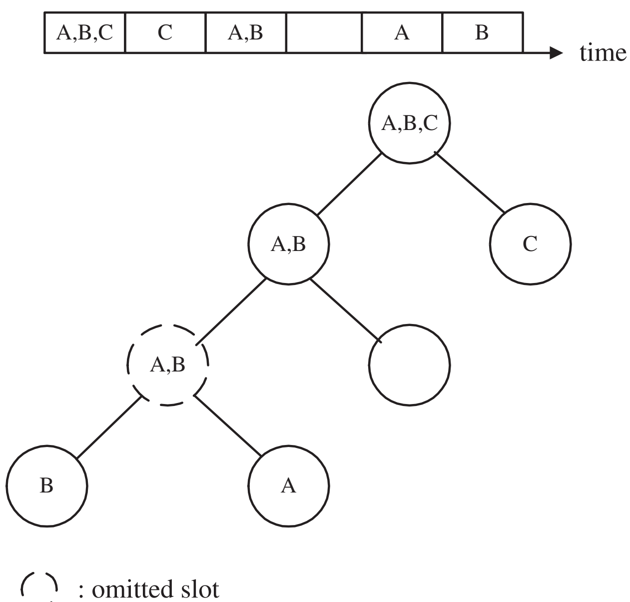

Fig. 3. An MTA Example: a collision followed by an idle slot indicates that the dashed subset contains two or more packets.

splitting with $p = 0.582$ is optimal, and results in 0.381 throughput.

B. SICTA

Collided packets are discarded in STA and MTA with no attempt to extract pertinent packet information. In contrast, the protocol we will detail here retains those collided packets for future reuse. The simple example of Fig. 1 has shown how to use SIC to reduce the number of transmission slots, which clearly offers the potential to improve throughput. A key observation which extends to more complicated scenarios is that the first slot in the left subtree can be omitted. The reason is that after the receiver decodes all packets in the right subtree, they can be cancelled from the received signals in the root slot. Therefore, we have

$$l_n = 1 + l_i + l_{n-i} - 1 = l_i + l_{n-i}, \quad n \geq 2. \tag{4}$$

Compared with (3), eq. (4) holds for any splitting, not just for the $i = 0$ case.

A more complicated example is shown in Fig. 4, which uses the same tree structure of Fig. 2, but three slots of transmission are now omitted. Since only packet C is transmitted in the second slot, it is successfully decoded from $\mathbf{Y}_2$ and is subtracted from $\mathbf{Y}_1$ to obtain: $\widetilde{\mathbf{Y}}_1 = \mathbf{Y}_1 - \mathbf{X}_C$. The signal vector $\widetilde{\mathbf{Y}}_1$ contains information about packets A and B, but the receiver can recover neither one of the two from their superposition; hence, it feeds back only one success ($k = 1$) to users. It is clearly unnecessary for A and B to transmit in the third slot, because the receiver already knows $\widetilde{\mathbf{Y}}_1$. Therefore, the algorithm proceeds directly to the next level of the tree and nothing is transmitted in slot 3 according to the tree. As in MTA, a collision slot followed by an idle slot means that the next slot can be skipped. Finally, after the receiver recovers $\mathbf{X}_A$ at the end of slot 4, the packet B can be recovered from $\widetilde{\widetilde{\mathbf{Y}}}_1 = \widetilde{\mathbf{Y}}_1 - \mathbf{X}_A = \mathbf{Y}_1 - \mathbf{X}_C - \mathbf{X}_A$; and the receiver feeds back $k = 2$.

The third example is shown in Fig. 5. After decoding the packet A at the end of the 4th slot, B and C can be recovered. The receiver identifies an idle left leaf as shown in the tree by subtracting $\mathbf{X}_A$ and $\mathbf{X}_B$ from $\mathbf{Y}_2$, and checking the energy of $\mathbf{Y}_2 - \mathbf{X}_A - \mathbf{X}_B$. Since three packets plus one left idle slot are identified, the receiver feeds back $k = 4$. Now the meaning of the number of identified slots should be clear: it indicates the number of successfully decoded packets plus the number of left-slots identified as being idle.

Next, we present an algorithmic description of SICTA. For each packet involved in a CRI, there is a level denoted as D_t, where t is the slot index. Initially, $D_1 = 0$. At the end of slot t, each user updates its level value as follows:

i) If feedback = e and $D_t > 0$, then

$$D_{t+1} = D_t + 1. \tag{5}$$

ii) If feedback = e and $D_t = 0$, then

$$D_{t+1} = \begin{cases} 0, & \text{with probability } p \\ 1, & \text{with probability } 1-p \end{cases} \tag{6}$$

where p is the probability to join the right subset.

iii) If feedback = 0 and $D_t > 1$, then

$$D_{t+1} = D_t. \tag{7}$$

iv) If feedback = 0 and $D_t = 1$, then

$$D_{t+1} = \begin{cases} 0, & \text{with probability } p \\ 1, & \text{with probability } 1-p \end{cases}. \tag{8}$$

iv) If feedback = k ($k \geq 1$), then

$$D_{t+1} = D_t - k. \tag{9}$$

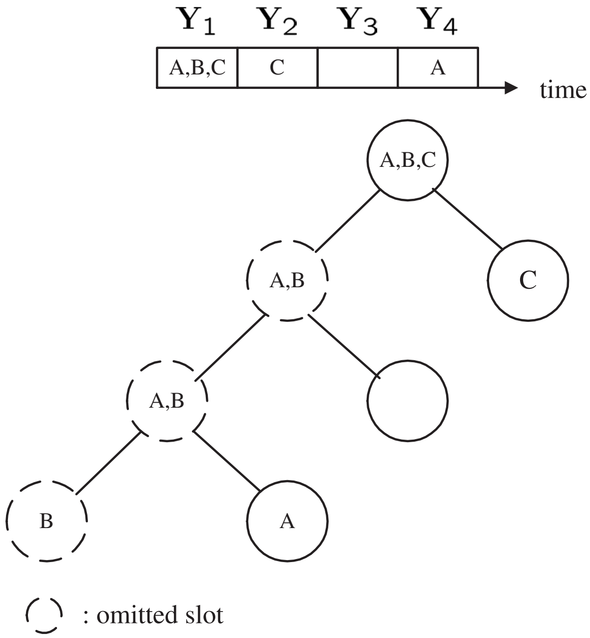

Fig. 4. SICTA Example 2: after decoding A, the receiver recovers B, and feeds back $k = 2$.

If $D_{t+1} < 0$, then the packet is successfully received and quits the remaining collision resolution procedure. If $D_{t+1} = 0$, then

$$D_{t+1} = \begin{cases} 0, & \text{with probability } p \\ 1, & \text{with probability } 1-p \end{cases}. \tag{10}$$

At the beginning of slot $t+1$, a user transmits its packet if and only if $D_{t+1} = 0$.

III. CRI Length Statistics and Throughput

In this section, we first derive the probability generating function (PGF) $Q_n(z)$ of the random variable l_n, the CRI length given that n packets initially collide in our gated access SICTA. This PGF can be used to recursively compute the first moment $L_n := \mathbb{E}\{l_n\}$ of l_n as well as higher moments. To obtain a direct non-recursive expression of L_n, we derive and solve a functional equation that will recur throughout the paper. An upper bound on L_n will be obtained and used subsequently to derive a sufficient condition for stability.

A. PGF and Average CRI Length

The specification of the binary SICTA yields the following relation for l_n (see (4)):

$$l_n = \begin{cases} 1, & \text{if } n = 0, 1 \\ l_i + l_{n-i}, & \text{if } n \geq 2, \end{cases} \tag{11}$$

where i is the number of nodes joining the right subset. Define the PGF as

$$Q_n(z) := \sum_{k=0}^{\infty} \mathbb{P}_r\{l_n = k\} z^k = \mathbb{E}\{z^{l_n}\}. \tag{12}$$

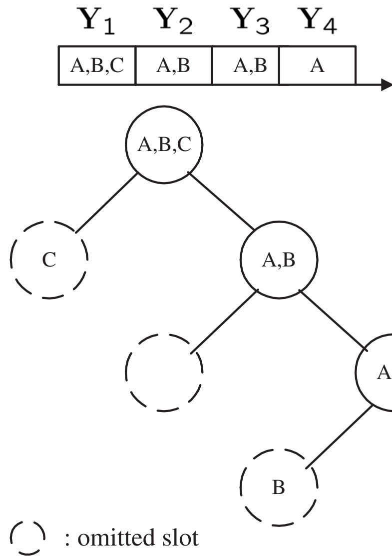

Fig. 5. SICTA Example 3: after decoding A, the receiver recovers B and C, identifies a left idle slot, and feeds back $k = 4$.

It is clear that

$$Q_0(z) = Q_1(z) = z. \tag{13}$$

By further conditioning i on the expectation of the right-hand side (RHS) of (12), we obtain for $n \geq 2$

$$\begin{aligned} Q_n(z) &= \mathbb{E}\{\mathbb{E}\{z^{l_n}|i\}|n\} \\ &= \mathbb{E}\{\mathbb{E}\{z^{l_i + l_{n-i}}|i\}|n\} \\ &= \sum_{i=0}^{n} B_{n,i} Q_i(z) Q_{n-i}(z), \end{aligned} \tag{14}$$

where

$$B_{n,i} = \binom{n}{i} p^i (1-p)^{n-i} \tag{15}$$

is the binomial probability of an $(i, n-i)$ split.

Differentiating (14) with respect to z and setting $z = 1$, yields the recursion

$$L_n = \sum_{i=0}^{n} B_{n,i}(L_i + L_{n-i}), \ n \geq 2, \tag{16}$$

with initial conditions

$$L_0 = L_1 = 1. \tag{17}$$

Therefore, L_n can be computed using the iteration:

$$L_n = \frac{\sum_{i=0}^{n-1}(B_{n,i} + B_{n,n-i})L_i}{1 - p^n - (1-p)^n}, \ \ n \geq 2. \tag{18}$$

However, we are interested in a closed-form expression for L_n, which will facilitate our throughput analysis. To this end, we assume that the initial collision size n is Poisson distributed with mean x. Upon defining the PGF

$$Q(x, z) := \sum_{n=0}^{\infty} Q_n(z) e^{-x} \frac{x^n}{n!}. \tag{19}$$

and substituting (13) and (14) into (19), we have

$$\begin{aligned}Q(x,z) &= z(1+x)e^{-x} + \sum_{n=2}^{\infty}\sum_{i=0}^{n} B_{n,i}Q_i(z)Q_{n-i}(z)e^{-x}\frac{x^n}{n!}\\ &= z(1+x)e^{-x} - z^2(1+x)e^{-x}\\ &\quad + \sum_{n=0}^{\infty}\sum_{i=0}^{n} B_{n,i}Q_i(z)Q_{n-i}(z)e^{-x}\frac{x^n}{n!}.\end{aligned} \tag{20}$$

The last sum on the RHS of (20) can be rewritten as

$$\begin{aligned}&\sum_{i=0}^{\infty}\sum_{n=i}^{\infty}\frac{n!}{i!(n-i)!}p^i(1-p)^{n-i}Q_i(z)Q_{n-i}(z)e^{-x}\frac{x^n}{n!}\\ &= \sum_{i=0}^{\infty}Q_i(z)e^{-px}\frac{(px)^i}{i!}\sum_{n=i}^{\infty}Q_{n-i}(z)e^{-(1-p)x}\frac{((1-p)x)^{n-i}}{(n-i)!}\\ &= Q(px,z)Q((1-p)x,z).\end{aligned} \tag{21}$$

Substituting back to (20), we obtain a functional equation for $Q(x,z)$

$$Q(x,z) = Q(px,z)Q((1-p)x,z) + (z-z^2)(1+x)e^{-x}. \tag{22}$$

Differentiating the latter and setting $z=1$ yields the average CRI length $L(x)$ as the solution of

$$L(x) = L(px) + L((1-p)x) - (1+x)e^{-x}. \tag{23}$$

B. Closed-Form Expression of L_n

Both (22) and (23) can be solved using the power series technique. Let us first solve (23) and postpone (22) to a later section. The power series representation

$$L(x) = \sum_{n=0}^{\infty}\alpha_n x^n \ , \tag{24}$$

can be re-written as

$$L(x) = \frac{\partial Q(x,z)}{\partial z}|_{z=1} = e^{-x}\sum_{n=0}^{\infty}L_n\frac{x^n}{n!}, \tag{25}$$

from which it follows directly that

$$\sum_{n=0}^{\infty}L_n\frac{x^n}{n!} = e^x\sum_{n=0}^{\infty}\alpha_n x^n = (\sum_{n=0}^{\infty}\frac{x^n}{n!})(\sum_{n=0}^{\infty}\alpha_n x^n). \tag{26}$$

Equating coefficients of x^n on both sides of (26), yields

$$\frac{L_n}{n!} = \sum_{i=0}^{n}\frac{\alpha_i}{(n-i)!}. \tag{27}$$

Eq. (27) expresses L_n as a weighted sum of α_i, for $i = 0,\ldots,n$. To find L_n, it suffices to find α_n, which can be done by solving the functional equation (23).

Substituting (24) into (23) and expanding e^{-x} in a power series, we have

$$\begin{aligned}\sum_{n=0}^{\infty}\alpha_n x^n &= \sum_{n=0}^{\infty}\alpha_n[p^n + (1-p)^n]x^n\\ &\quad - \sum_{n=0}^{\infty}\frac{(-1)^n x^n(1+x)}{n!}.\end{aligned} \tag{28}$$

Solving (28) for α_n, we obtain [1]

$$\begin{aligned}\alpha_0 &= 1,\\ \alpha_1 &= 0,\\ \alpha_n &= \frac{(n-1)}{[1-p^n-(1-p)^n]}\cdot\frac{(-1)^n}{n!}.\end{aligned} \tag{29}$$

Now we are ready to obtain a non-recursive expression for L_n. Substituting (29) into (27) leads to

$$L_n = 1 + \sum_{i=2}^{n}\binom{n}{i}\frac{(i-1)(-1)^i}{1-p^i-(1-p)^i},\ n \geq 2. \tag{30}$$

There is a simple connection between STA and SICTA. Let l'_n to denote the CRI length given that n packets initially collide in STA. With $L'_n := \mathbb{E}\{l'_n\}$, it is known that [16]

$$\begin{aligned}L'_0 &= L'_1 = 1,\\ L'_n &= 1 + \sum_{i=0}^{n}B_{n,i}(L'_i + L'_{n-i}),\ n \geq 2,\end{aligned} \tag{31}$$

and that L'_n is expressible in closed-form as

$$L'_n = 1 + \sum_{i=2}^{n}\binom{n}{i}\frac{2(i-1)(-1)^i}{1-p^i-(1-p)^i},\ n \geq 2. \tag{32}$$

From (30) and (32), we easily obtain a relationship between L_n and L'_n:

$$L'_n - 1 = 2(L_n - 1). \tag{33}$$

Actually, there is another way to compute L_n if we prove that (33) holds first, and then compute L_n since L'_n is known. We can start from (16), (17), (31) and use induction to prove (33). As this is rather straightforward, its proof is omitted for brevity.

It is easy to show that each sum on the RHS of (30) is minimized by setting $p = 1/2$. Therefore, similar to the binary STA, fair splitting is also optimal for the binary SICTA. Hereafter, we only consider the case $p = 1/2$. For fair splitting, there is a known linear upper bound for L'_n [11], [16]

$$L'_n \leq \alpha n + 1,\ \forall n \geq 0, \tag{34}$$

where $\alpha \approx 2.886$. Thanks to (33), a similar linear upper bound becomes available for L_n

$$L_n \leq \frac{\alpha}{2}n + 1,\ \forall n \geq 0. \tag{35}$$

[1]Notice α_1 is not determined from the power series. It is easy to see that $\alpha_1 = 0$ from (27), since $L_1 = \alpha_0 + \alpha_1$ and $L_1 = 1$.

Furthermore, this bound generates a bound on $L(x)$:

$$\begin{aligned} L(x) &= \sum_{n=0}^{\infty} L_n e^{-x} \frac{x^n}{n!} \\ &\le \sum_{n=0}^{\infty} [\frac{\alpha}{2} n + 1] e^{-x} \frac{x^n}{n!} \\ &= \frac{\alpha}{2} x + 1, \quad \forall x \ge 0. \end{aligned} \tag{36}$$

C. Throughput

Now we are ready to study the stability property of SICTA based on Pakes' lemma which provides sufficient conditions for the ergodicity of a Markov chain [2]. We consider a system with an infinite population generating Poisson arrivals with mean λ. Let c_i be the length of the i-th CRI, $i = 0, \ldots, \infty$. For gated access SICTA, c_{i+1} depends only on the number of packets at the beginning of the $(i+1)$-st CRI, denoted as s_{i+1}, which is the number of new packets arriving during the i-th CRI. Conditioning on the event that $c_i = n$, s_{i+1} is Poisson distributed with mean λn due to Poisson arrivals:

$$\mathbb{P}_r\{s_{i+1} = k | c_i = n\} = e^{-\lambda n} \frac{(\lambda n)^k}{k!}. \tag{37}$$

We observe that $\{c_{i+1}, i = 0, \ldots, \infty\}$ forms a homogeneous Markov chain, for which we assume without loss of generality that is also irreducible and aperiodic with drift

$$d_n = \mathbb{E}\{c_{i+1} - c_i | c_i = n\}. \tag{38}$$

According to Pakes' lemma, the Markov chain is ergodic if its drift satisfies

i) $|d_n| < \infty \quad \forall n$, and
ii) $\limsup_{n \to \infty} d_n < 0$.

Since c_{i+1} is Poisson distributed with mean λn given $c_i = n$, its drift is

$$\begin{aligned} d_n &= \mathbb{E}\{c_{i+1} | c_i = n\} - n \\ &= \sum_{k=0}^{\infty} L_k e^{-\lambda n} \frac{(\lambda n)^k}{k!} - n \\ &= L(\lambda n) - n. \end{aligned} \tag{39}$$

From the linear upper bound (36), we have

$$d_n \le (\frac{\alpha}{2}\lambda - 1)n + 1. \tag{40}$$

Clearly, both conditions of Pakes' lemma hold for $\lambda < 2/\alpha \approx 0.693$. Therefore, our gated-access SICTA is stable if $\lambda < 0.693$, which is twice that of STA.

The main advantage of SICTA is its high stable throughput. Indeed, the best contention TA is the FCFS algorithm, which offers a stable throughput of 0.487 [2]. On the other hand, it has been proved that all conventional random access algorithms, including contention tree algorithms, are unstable for $\lambda > 0.568$ [7]. SICTA not only outperforms FCFS, but also exceeds the limit of the conventional random access algorithms.

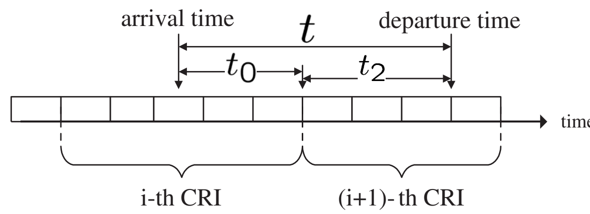

Fig. 6. Two Delay Components: t_0 and t_2

IV. Delay Analysis

In this section, we explore the delay properties of SICTA. Specifically, we derive the average packet delay. The technique can be extended to obtain higher moments of delay, but is omitted due to space limitations.

Our approach mimics the one used in [17] to compute the delay for the gated access binary STA and MTA. Suppose that a randomly chosen "tagged" packet arrives during the i-th CRI and leaves the system during the $(i+1)$-st CRI. The total delay t experienced by this "tagged" packet is partitioned into the following two components as depicted in Fig. 6:

t_0: the *initial delay* required before joining the $(i+1)$-st CRI, which measures the residual life of the i-th CRI at the moment of its arrival.

t_2: the *collision resolution time* for the "tagged" packet to leave the system, which covers the rest of the time from the first transmission of the "tagged" packet until its resolution.

We begin by deriving the distribution of the CRI length in the steady state:

$$\overrightarrow{\pi} = (\pi_1, \pi_2, \pi_3, \ldots), \tag{41}$$

where π_j is the probability that a random chosen CRI will be j slots long. As pointed out in the pervious section, under the infinite population Poisson model, the CRI lengths $\{c_{i+1}, i = 0, 1, \ldots, \infty\}$ form a homogenous Markov chain. For this reason, the steady-state distribution can be found by solving the usual global balance equation

$$\overrightarrow{\pi} = \overrightarrow{\pi}\mathbf{P} \tag{42}$$

under the constraint

$$\sum_{j=1}^{\infty} \pi_j = 1 \ , \tag{43}$$

where the infinite-size matrix $\mathbf{P}$ represents the transition probability matrix of the Markov chain and its (i, j)-th entry $P_{i,j}$ is the probability that a CRI of length i is followed by a CRI of length j. We will derive $P_{i,j}$ by solving (22) later.

For now, we use the fact that we have Poisson packet arrivals and apply standard renewal-theoretic results to obtain $\widetilde{\pi}_n$, the probability that the "tagged" packet joins the system during a CRI of length n:

$$\widetilde{\pi}_n = \frac{n\pi_n}{\sum_{n=1}^{\infty} n\pi_n}. \tag{44}$$

If we condition on the event that the "tagged" packet arrives during a CRI of length n, then t_0 has a uniform distribution over the interval $[0, n)$, and the distribution of t_2 should be conditioned on the fact that the "tagged" packet will be competing with a group of other packets having a Poisson distribution with mean λn [c.f. (37)]. In addition, the two conditional distributions are independent and the distribution of the sum $t = t_0 + t_2$ factors into the product of their marginal distributions. Therefore, the Laplace transform of the packet delay can be written immediately in the form:

$$D^*(s) = \sum_{n=0}^{\infty} \widetilde{\pi}_n \cdot U^*(n, s) \cdot G(\lambda n, e^{-s}), \tag{45}$$

where

$$U^*(n, s) = \frac{1 - e^{-sn}}{ns}$$

is the Laplace transform of the uniform distribution over $[0, n)$ with mean $n/2$ and variance $n^2/12$; and $G(x, z)$ is the PGF of t_2, given that the "tagged" packet is competing with Poisson traffic having mean x, as described later. Once we obtain $D^*(s)$, we can compute all the moments of t using standard techniques. In this paper, we are only interested in the mean of the packet delay:

$$\overline{T} = \mathbb{E}\{t_0\} + \mathbb{E}\{t_2\} = \sum_{n=0}^{\infty} \widetilde{\pi}_n \cdot [\frac{n}{2} + T_{2|n}], \tag{46}$$

where $T_{2|n}$ is the conditional mean of t_2, the time spent in the next CRI given that the "tagged" packet joins a CRI with length n.

In the following, we delve further $\overrightarrow{\pi}$ and t_2.

A. Steady-state Distribution of CRI length

To obtain the steady-state distribution of the CRI length, we need to solve (42) and (43). First we shall find $\mathbf{P}$, whose (i, j)-th entry $P_{i,j}$ is the probability that a CRI of length i is followed by a CRI of length j. Since the length of the next CRI depends only the number of initial contenders which arrive during the previous CRI, it follows from (37) that

$$\begin{aligned} P_{i,j} &= \sum_{n=0}^{\infty} \mathbf{P}_r\{l_n = j\} e^{-\lambda i} \frac{(\lambda i)^n}{n!} \\ &= q_j(\lambda i), \end{aligned} \tag{47}$$

where the function $q_j(x)$ is defined as

$$q_j(x) := \sum_{n=0}^{\infty} \mathbf{P}_r\{l_n = j\} e^{-x} \frac{x^n}{n!}, \;\; j = 0, \ldots, \infty. \tag{48}$$

It turns out that $\{q_j(x)\}_{j=0}^{\infty}$ has a close relation with the PGF $Q(x, z)$ defined in (19). Substituting (12) into (19) and switching the summation order leads to

$$\begin{aligned} Q(x, z) &= \sum_{n=0}^{\infty} [\sum_{j=0}^{\infty} \mathbb{P}_r\{l_n = j\} z^j] e^{-x} \frac{x^n}{n!} \\ &= \sum_{j=0}^{\infty} [\sum_{n=0}^{\infty} \mathbb{P}_r\{l_n = j\} e^{-x} \frac{x^n}{n!}] z^j \\ &= \sum_{j=0}^{\infty} q_j(x) z^j. \end{aligned} \tag{49}$$

Therefore, $\{q_j(x)\}$ are the exact coefficients of the power series expansion of $Q(x, z)$ over z, which can be obtained by solving (22) with $p = 1/2$.

Substituting the power series representation (49) into (22), we have

$$\sum_{j=0}^{\infty} q_j(x) z^j = [\sum_{j=0}^{\infty} q_j(\frac{x}{2}) z^j]^2 + (z - z^2)(1 + x) e^{-x}, \tag{50}$$

which yields the solution after equating the coefficients of z^j on both sides

$$\begin{aligned} &q_0(x) = 0, \\ &q_1(x) = (1 + x) e^{-x}, \\ &q_2(x) = \frac{x^2}{4} e^{-x}, \\ &q_j(x) = \sum_{i=1}^{j-1} q_i(\frac{x}{2}) q_{j-i}(\frac{x}{2}), \;\; j \geq 3. \end{aligned} \tag{51}$$

Eq. (51) offers a recursive means of computing $q_j(x)$.

We are now ready to compute $\mathbf{P}$ from (47). To solve (42) and (43) numerically, we need to truncate the infinite-dimensional $\overrightarrow{\pi}$ to a finite size.

B. Collision Resolution Delay t_2

After the steady-state distribution, the next step is to compute the conditional mean of t_2 given that the "tagged" packet joins a CRI of length n, namely $T_{2|n}$.

The collision resolution time t_2 is the duration that a "tagged" packet spends in the CRI from the first transmission of the "tagged" packet until it is successfully resolved. Let $t_{2,m}$ denote the time the "tagged" packet spends in the CRI, when there are m other packets competing with the "tagged" packet for the channel at the beginning. Due to Poisson arrivals, m should be Poisson distributed with mean λn when the "tagged" packet joins a CRI of length n; therefore, we have

$$\begin{aligned} T_{2|n} &= \sum_{m=0}^{\infty} \mathbb{E}\{t_{2,m}\} e^{-\lambda n} \frac{(\lambda n)^m}{m!} \\ &= \sum_{m=0}^{\infty} \sum_{k=1}^{\infty} k \mathbb{P}_r\{t_{2,m} = k\} e^{-\lambda n} \frac{(\lambda n)^m}{m!} \\ &= T_2(\lambda n), \end{aligned} \tag{52}$$

where the function $T_2(x)$ is defined as

$$T_2(x) := \sum_{m=0}^{\infty} \sum_{k=1}^{\infty} k \mathbb{P}_r\{t_{2,m} = k\} e^{-x} \frac{x^m}{m!}. \tag{53}$$

With the goal of computing $T_2(x)$, we notice that if the "tagged" packet joins the first (right) subset (sub-CRI) after the first collision, we have

$$t_{2,m} = 1 + t_{2,i},\ m \geq 1, \tag{54}$$

where i refers to the number of other packets joining the right sub-CRI; if the "tagged packet" packet joins the second (left) subset (sub-CRI), we have

$$\begin{aligned} t_{2,m} &= 1 + l_i + t_{2,m-i} - 1,\ m \geq 1 \\ &= l_i + t_{2,m-i}. \end{aligned} \tag{55}$$

Let us define the PGF of $t_{2,m}$ as

$$G_{m+1}(z) := \sum_{k=0}^{\infty} \mathbb{P}_r\{t_{2,m} = k\} z^k, \tag{56}$$

and

$$\begin{aligned} G^{(s)}_{m+1}(z) :&= \sum_{k=0}^{\infty} \mathbb{P}_r\{t_{2,m} = k | \text{it joins sub-CRI } s\} z^k, \\ & m \geq 1, \end{aligned} \tag{57}$$

where s is 0 or 1, for the first and second sub-CRI, respectively. It is clear that $t_{2,0} = 1$, which means

$$G_1(z) = z. \tag{58}$$

It follows from (54) and (55) that

$$G^{(0)}_{m+1}(z) = z \sum_{i=0}^{m} B_{m,i} G_{i+1}(z),\ m \geq 1, \tag{59}$$

and

$$G^{(1)}_{m+1}(z) = \sum_{i=0}^{m} B_{m,i} Q_i(z) G_{m-i+1}(z),\ m \geq 1. \tag{60}$$

In fair splitting, the "tagged" packet joins both sub-CRIs with equal probability; thus, we have

$$\begin{aligned} G_{m+1}(z) &= \frac{1}{2} G^{(0)}_{m+1}(z) + \frac{1}{2} G^{(1)}_{m+1}(z), \\ &= \frac{1}{2} z \sum_{i=0}^{m} B_{m,i} G_{i+1}(z) \\ &\quad + \frac{1}{2} \sum_{i=0}^{m} B_{m,i} Q_i(z) G_{m-i+1}(z), \\ &\quad m \geq 1. \end{aligned} \tag{61}$$

Following the technique used in Section III, we define

$$G(x,z) := \sum_{m=0}^{\infty} G_{m+1}(z) e^{-x} \frac{x^m}{m!}. \tag{62}$$

Substituting (61) into (62) yields

$$\begin{aligned} G(x,z) = &\sum_{m=0}^{\infty} \frac{1}{2} z \sum_{i=0}^{m} B_{m,i} G_{i+1}(z) e^{-x} \frac{x^m}{m!} \\ &+ \sum_{m=0}^{\infty} \frac{1}{2} \sum_{i=0}^{m} B_{m,i} Q_i(z) G_{m-i+1}(z) e^{-x} \frac{x^m}{m!} \\ &+ z e^{-x} - \frac{1}{2} z^2 e^{-x} - \frac{1}{2} z^2 e^{-x}. \end{aligned} \tag{63}$$

The first term in (63) can be rewritten as

$$\begin{aligned} &\frac{1}{2} z \sum_{i=0}^{\infty} e^{-\frac{x}{2}} \frac{(x/2)^i}{i!} \sum_{m=i}^{\infty} G_{i+1}(z) e^{-\frac{x}{2}} \frac{(x/2)^{m-i}}{(m-i)!} \\ &\qquad = \frac{1}{2} z G(\frac{x}{2}, z). \end{aligned} \tag{64}$$

Similarly, we re-write the second term in (62) as

$$\begin{aligned} &\frac{1}{2} \sum_{i=0}^{\infty} Q_i(z) e^{-\frac{x}{2}} \frac{(x/2)^i}{i!} \sum_{m=i}^{\infty} G_{m-i+1}(z) e^{-\frac{x}{2}} \frac{(x/2)^{m-i}}{(m-i)!} \\ &\qquad = \frac{1}{2} Q(\frac{x}{2}, z) G(\frac{x}{2}, z). \end{aligned} \tag{65}$$

Combining (64) and (65), we arrive at

$$G(x,z) = \frac{1}{2} z G(\frac{x}{2}, z) + \frac{1}{2} Q(\frac{x}{2}, z) G(\frac{x}{2}, z) + (z - z^2) e^{-x}. \tag{66}$$

Notice that

$$T_2(x) = \frac{\partial G(x,z)}{\partial z}|_{z=1}. \tag{67}$$

Differentiating (66) and setting $z = 1$, we obtain

$$T_2(x) = \frac{1}{2} + T_2(\frac{x}{2}) + \frac{1}{2} L(\frac{x}{2}) - e^{-x}, \tag{68}$$

where $L(x)$ is given by (24).

Once again, we use the power series method to solve this functional equation. Assuming

$$T_2(x) = \sum_{k=0}^{\infty} \theta_k x^k\ , \tag{69}$$

and substituting it into (68) yields the solution

$$\begin{aligned} &\theta_0 = 1, \\ &\theta_k = \frac{2^{-k}}{2(1-2^{-k})} \alpha_k - \frac{1}{1-2^{-k}} \cdot \frac{(-1)^k}{k!},\ k \geq 1, \end{aligned} \tag{70}$$

where $\{\alpha_k\}$ are given by (29).

V. Numerical Results

We use the technique of [17] to numerically calculate the steady-state distribution of CRI lengths and the average delay. We omit the details but present the final results. Fig. 7 illustrates throughput versus average packet delay for STA, MTA, SICTA all with gated access, and FCFS. Table I summarizes the calculated results, where the data for STA and MTA are from [17] and the data for FCFS are from [18]. It is clear that SICTA's delay curve stays always below others.

VI. Conclusions

In this paper, we derived a novel contention tree algorithm, that we termed SICTA, since it is based on successive interference cancellation. Its main advantage is high throughput, reaching 0.693 - the highest among all contention tree algorithms. The reason for the performance improvement is the novel idea of having collided packets used to extract packet information relying on successive interference cancellation and the tree structure of the collision resolution algorithm. We also

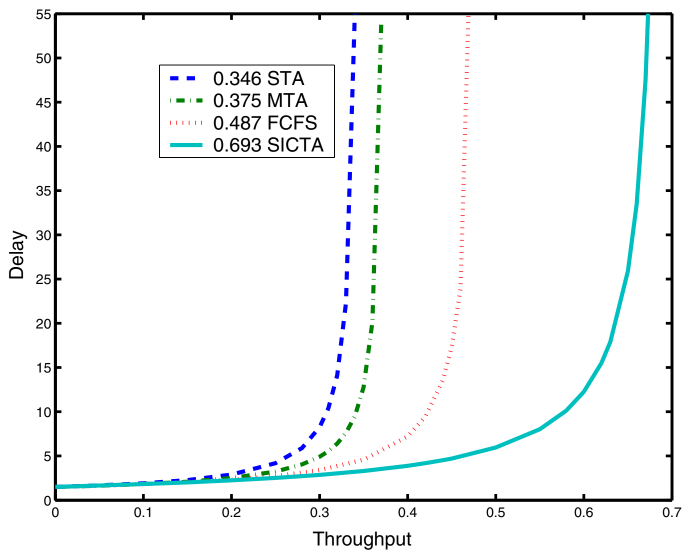

Fig. 7. Delay vs. Throughput

TABLE I
PACKET DELAY FOR STA, MTA, FCFS AND SICTA

λ	STA	MTA	λ	FCFS	λ	SICTA
0.10	1.909	1.825	0.10	1.81	0.1	1.8244
0.20	2.896	2.502	0.20	2.31	0.20	2.245
0.25	4.184	3.238	0.25	2.73	0.30	2.861
0.30	8.246	4.912	0.30	3.40	0.40	3.887
0.31	10.38	5.543	0.35	4.58	0.50	5.951
0.32	14.11	6.392	0.40	7.22	0.60	12.25
0.33	22.28	7.603	0.42	9.43	0.62	15.51
0.34	55.11	9.471	0.44	13.57	0.63	17.89
0.35	–	12.76	0.45	17.34	0.65	25.91
0.36	–	20.17	0.46	23.92	0.66	33.48
0.37	–	54.13	0.48	92.91	0.67	47.51

derived methods for calculating the CRI distribution in steady-state, the CRI length, and the packet delay. Although only the first moment was given, the technique can be extended to obtain higher order moments as well. Throughout the paper, we assumed that the interference cancellation is perfect and the receiver has infinite storage. Our future work will address the effect of error propagation and the finite storage limit on the overall system performance.

ACKNOWLEDGMENTS

This work was supported in part by the ARL/CTA Grant No. DAAD19-01-2-0011

REFERENCES

[1] I. F. Akyildiz, J. McNair, L. C. Martorell, R. Puigjaner and Y. Yesha, "Medium access control protocols for multimedia traffic in wireless networks," *IEEE Network*, vol. 13, no. 4, pp. 39–47, July/Augest 1999.
[2] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1992.
[3] N. Abramson, "The ALOHA system - another alternative for computer communications," in *Proc. of Fall Joint Comput. Conf. AFIPS Conf.*, vol. 37, 1970.
[4] L. Kleinrock and F.A. Tobagi, "Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput/delay characteristics," *IEEE Trans. on Commun.*, vol. 23, no. 12, pp. 1400–1416, Sept. 1975.
[5] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. on Inform. Theory*, vol. 25, no. 4, pp. 505–515, Sept. 1979.
[6] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Problemy Peredachi Informatsii*, vol. 14, no. 4, pp. 32–59, Oct.-Dec. 1978.
[7] B. S. Tsybakov and N. B. Likhanov, "Upper bound on the capacity of a random multiple access system," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 246–236, Jan. 1988.
[8] R. Gallager, "A perspective on multiaccess channels," *IEEE Trans. on Information Theory*, vol. 27, no. 3, pp. 124–142, Mar. 1982.
[9] B. S. Tsybakov and V. A. Mikhailov, "Random multiple packet access: part-and-try algorithm," *Problemy Peredachi Informatsii*, vol. 16, no. 4, pp. 65–79, Oct.-Dec. 1980.
[10] G. Ruget, "Some tools for the study of channel sharing algorithms," in *Multiuser Communication Systems 265*, ed. G. Longo, CISM Course and Lecture Notes, Spriger, New York, pp. 201–231, 1981.
[11] J. L. Massey, "Collision resolution algorithm and random access communications," in *Multiuser Communication Systems*, ed. G. Longo, CISM Course and Lecture Notes 265, Spriger, New York, pp. 73–131, 1981.
[12] 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.
[13] IEEE 802.14 WG, "Cable-TV access method and physical layer specification," Draft 3, Revison 3, Oct. 1998.
[14] MCNS Holdings, "Data-over-cable service interface specifications, radio frequency interface specification," Ref. SP-RFIv1.1-PI01-990226, Feb. 1999.
[15] M. X. van den Broek, I.B.J.F. Adan, N. Sai Shankar and S. C. Borst, "A novel mechanism for contention resolution in HFC networks," *Proc. of INFOCOM Conf.*, San Francisco, CA, pp. 979–989, March 30-April 3, 2003.
[16] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*, Springer-Verlag, New York, 1990.
[17] M. L. Molle and A. C. Shih, "Computation of the packet delay in Massey's standard and modified tree conflict resolution algorithms with gated access," *Technical Report CSRI-264*, Computer Systems Research Insitute, University of Toronto, Toronto, Canada, Feb. 1992.
[18] M. L. Molle and G. C. Polyzos, "Conflict resolution algorithms and their performance analysis," *Technical Report CS93-300*, Department of Computer Science and Engineering, University of California, San Diego, July 1993.
[19] M. K. Tsatsanis, R. Zhang and S. Banerjee, "Network-assisted diversity for random access wireless networks," *IEEE Trans. on Signal Processing*, vol. 48, no. 3, pp. 702–711, March 2000.
[20] R. Zhang, N. D. Sidiropoulos and M. K. Tsatsanis, "Collision resolution in packet radio networks using rotational invariance techniques," *IEEE Trans. on Commun.*, vol. 50, no. 1, pp. 146–155, Jan. 2002.
[21] G. Dimic, N. D. Sidiropoulos and L. Tassiulas, "Wireless networks with retransmission diversity access mechanisms: stable throughput and delay properties," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2019–2030, Aug. 2003.