Multiple Access Control with Intelligent Bandwidth Allocation for Wireless ATM Networks

Maria C. Yuang, *Member, IEEE,* and Po L. Tien

***Abstract*—Two major challenges pertaining to wireless asynchronous transfer mode (ATM) networks are the design of multiple access control (MAC), and dynamic bandwidth allocation. While the former draws more attention, the latter has been considered nontrivial and remains mostly unresolved. In this paper, we propose a new intelligent multiple access control system (IMACS) which includes a versatile MAC scheme augmented with dynamic bandwidth allocation, for wireless ATM networks. IMACS supports four types of traffic—CBR, VBR, ABR, and signaling control (SCR). It aims to efficiently satisfy their diverse quality-of-service (QoS) requirements while retaining maximal network throughput. IMACS is composed of three components: multiple access controller (MACER), traffic estimator/predictor (TEP), and intelligent bandwidth allocator (IBA). MACER employs a hybrid-mode TDMA scheme, in which its contention access is based on a new dynamic-tree-splitting (DTS) collision resolution algorithm parameterized by an optimal splitting depth (SD). TEP performs periodic estimation and on-line prediction of ABR self-similar traffic characteristics based on wavelet analysis and a neural-fuzzy technique. IBA is responsible for *static* bandwidth allocation for CBR/VBR traffic following a closed-form formula. In cooperation with TEP, IBA governs *dynamic* bandwidth allocation for ABR/SCR traffic through determining the optimal SD. The optimal SD's under various traffic conditions are postulated via experimental results, and then off-line constructed using a back propagation neural network (BPNN), being used on-line by IBA. Consequently, with dynamic bandwidth allocation, IMACS offers various QoS guarantees and maximizes network throughput irrelevant to traffic variation.**

***Index Terms*—Bandwidth allocation, collision resolution algorithm, multiple access control (MAC), neural-fuzzy technique, quality-of-service (QoS), self-similar traffic, wireless asynchronous transfer mode networks (WATM).**

I. Introduction

WITH THE rapid proliferation of personal communication services provided to multimedia portable computers, wireless access to existing networks has emerged as a significant concern [1]. Essentially, wireless ATM [2] has been envisioned as a potential framework for next-generation wireless networks capable of supporting integrated multimedia services with a wide range of service rates and different quality-of-service (QoS) requirements. Expected supported services include constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and signaling control (SCR) for CBR/VBR traffic. Two major challenges pertaining to such wireless ATM networks are the design of multiple access control (MAC), and dynamic bandwidth allocation.

Existing MAC schemes, such as time-division multiple access (TDMA) [2]–[5] and code-division multiple access (CDMA) [4], [6], [7], exhibit various performance merits and weaknesses. This paper, taking advantage of CDMA features, mainly focuses on the design of a TDMA-based MAC protocol. Generally, compared to solely reservation-based or contention-based TDMA, the combination of reservation-based and contention-based, namely the hybrid-mode TDMA [8]–[10] has been considered most promising. In essence, the reservation-access mode is indubitably advantageous for guaranteed services, such as CBR/VBR traffic. The contention-access mode, on the other hand, is beneficial to the best effort and access-delay-sensitive traffic, such as ABR and SCR traffic, respectively. While the former mode has been considerably explored in the literature, the latter mode, especially the design of collision resolution [5], becomes one of the major focuses of this paper.

Existing collision resolution algorithms are either distributed-oriented [11] or centralized-oriented [12], [13]. In the distributed-oriented algorithm, each backlogged station probabilistically computes the backoff time interval for the subsequent retransmission based on the ALOHA protocol. This algorithm [11] was shown to achieve high utilization via simulation. On the other hand, in centralized-oriented algorithms, the central station resolves collisions in a deterministic and FCFS manner. The examples obtaining the most merit are tree-splitting algorithms [5], [12]. They can be further classified as being exhaustive [12] or static [13]. Exhaustive tree-splitting algorithms defer new transmissions until all previously collided packets have been resolved. These algorithms ensure FCFS transmissions, but unfortunately suffer from throughput degradation and occasional drastic increases in delay for other traffic. In contrast, the static tree-splitting algorithm resumes new transmissions when the number of tree splittings reaches the predetermined, fixed splitting depth (SD). This algorithm offsets the drawbacks of exhaustive splitting algorithms. Nevertheless, the engagement of a single SD can be impractical for networks undergoing traffic fluctuation. The first goal of this paper is to propose a new dynamic tree-splitting collision resolution algorithm using an optimal SD.

With regard to bandwidth allocation, there are two prevailing classes of mechanisms—static allocation and dynamic allocation. A significant static allocation application is admission control [14], which is beyond the scope of this paper. Based on static

Manuscript received October 1, 1999; revised March 1, 2000. This work was supported in part by the MOE Program of Excellence Research under Contract 89-E–FA04-1-4-X89008, and in part by the National Science Council, ROC, under Grant NSC 89-2219-E-009-001.

The authors are with the Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu 30050, Taiwan, ROC (e-mail: mcyuang@csie.nctu.edu.tw; tbl@csie.nctu.edu.tw)

Publisher Item Identifier S 0733-8716(00)07138-9.

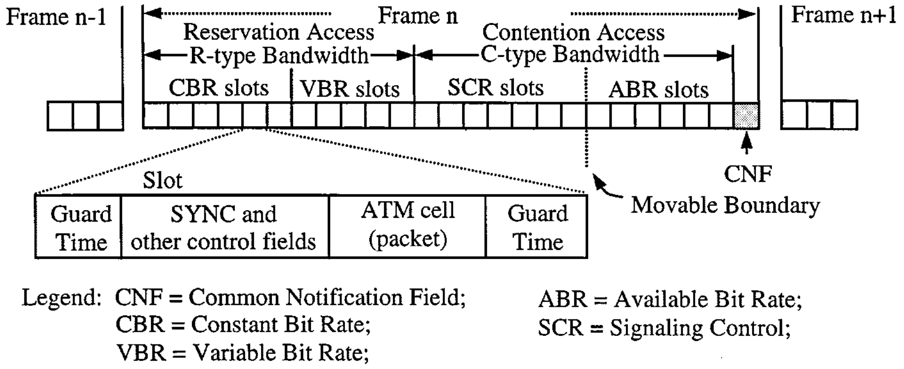


Fig. 1. Frame and slot structures.

allocation, PRMA and companions [8] provided QoS guarantees for traditional CBR voice traffic only. The dynamic allocation mechanisms proposed in [15] and [16] managed efficient bandwidth usage particularly for VBR or CBR traffic, unfortunately, with complete disregard for SCR traffic. DQRUMA [17] further offered minimum delay guarantee for SCR traffic, but discounted differentiated services between VBR and ABR traffic. PRMA/DA [9] governed dynamic bandwidth allocation among CBR, VBR, and SCR traffic, however, at the expense of a noticeable decrease in network throughput. Ultimately, the second goal of this paper is to provide efficient static and dynamic allocation for the four aforementioned services while retaining maximal network throughput.

In this paper, we propose an intelligent division multiple control system (IMACS) for wireless ATM networks, supporting CBR, VBR, ABR, and SCR traffic types. IMACS is composed of three components: multiple access controller (MACER), traffic estimator/predictor (TEP), and intelligent bandwidth allocator (IBA). MACER employs a hybrid-mode TDMA scheme, incorporating reservation access and contention access governing the CBR/VBR and ABR/SCR traffic, respectively. In particular, this contention access is based on a new dynamic-tree-splitting (DTS) collision resolution algorithm using an optimal splitting depth (SD). Based on wavelet analysis and a neural-fuzzy technique, TEP performs periodic estimation and on-line prediction of ABR self-similar traffic characteristics. IBA is responsible for the static allocation of reservation bandwidth to VBR and CBR on a call basis. In cooperation with TEP, IBA also governs the dynamic allocation of contention bandwidth by determining the optimal SD, aiming to balance the tradeoff between ABR throughput and SCR blocking probability. Finally, experimental results postulate the optimal SD as a complex function of ABR mean, variance, the Hurst parameter, and SCR mean. These results are off-line trained and constructed using a back propagation neural network (BPNN), which is efficiently used on-line by IBA.

Thus, the major contribution of this paper is summarized as follows.

- MACER performs a hybrid-mode TDMA scheme with contention access based on a new dynamic-tree-splitting (DTS) collision resolution algorithm.
- TEP performs on-line ABR self-similar traffic prediction using a self-constructing neural-fuzzy inference network.
- IBA provides static allocation for VBR traffic via a closed form formula, and dynamic allocation for ABR and SCR traffic by determining the optimal SD parameter.

The remainder of this paper is organized as follows. Section II presents the architecture of IMACS. Section III describes the MACER operation, including its MAC scheme and the DTS collision resolution algorithm. Section IV outlines the TEP logic. Section V provides throughput analyses and experimental results on which IBA is based for optimal-SD determination. Finally, concluding remarks are given in Section VI.

II. The IMACS Architecture

IMACS operates in the base station (BS) of an infrastructure-based wireless ATM network [2]. The medium bandwidth is divided into two separate channels: uplink and downlink. The uplink channel transfers information from mobile terminals (MT's) to the BS, based on a new hybrid TDMA scheme described in the next section. The downlink channel typically broadcasts information and acknowledges previous transmissions made on the uplink channel. This operation is beyond the scope of this paper. Furthermore, time on the uplink channel is divided into a contiguous sequence of fixed-size TDMA frames (see Fig. 1).

Each frame is further subdivided into a fixed number of slots to be dynamically allocated to four ATM-traffic classes: CBR, VBR, ABR, and SCR. As was mentioned, while CBR and VBR traffic are governed by reservation access using reservation (R)-type bandwidth, ABR and SCR traffic are controlled by contention access using contention (C)-type bandwidth. Each slot contains a data packet or, more specifically, an ATM cell, other than guard times, sync, and other control fields [3]. Notice that, with guard times provided, the propagation delay between the BS and MT's can be ignored. This in turn allows acknowledgment for all packet transmissions made in the current slot to be available to all MT's prior to the beginning of the next slot.

Most significantly, the network is assumed to use phase-shift keying (PSK)-based encoding equipped with simple CDMA capability [18], namely pseudo-code sequence generation. Essentially, all MT's with ABR packets in their buffers are required to inform the BS through placing different code sequences at the last slot of each frame, called the common notification field (CNF). Due to orthogonality and phase

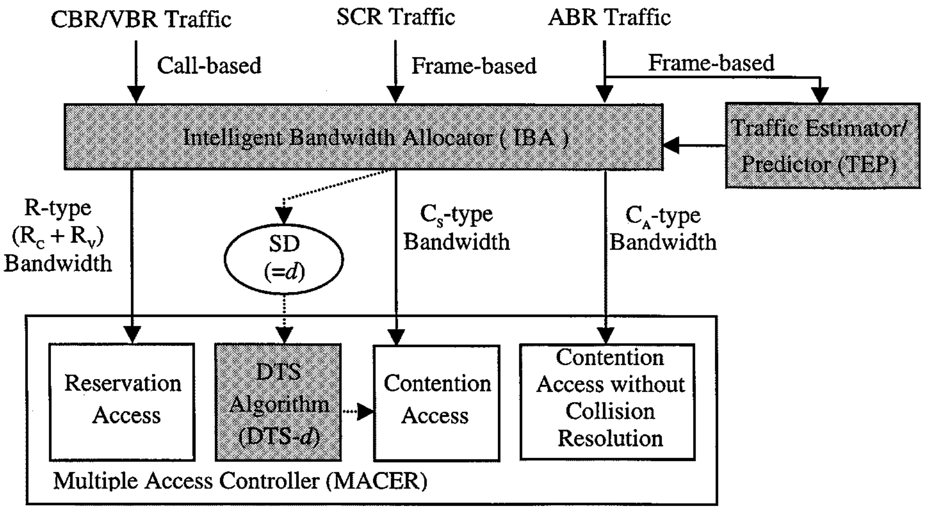

CBR/VBR Traffic
SCR Traffic
ABR Traffic
Call-based
Frame-based
Frame-based
Intelligent Bandwidth Allocator (IBA)
Traffic Estimator/ Predictor (TEP)
R-type ($R_C + R_V$) Bandwidth
SD (=d)
C_S-type Bandwidth
C_A-type Bandwidth
Reservation Access
DTS Algorithm (DTS-d)
Contention Access
Contention Access without Collision Resolution
Multiple Access Controller (MACER)

Fig. 2. IMACS architecture.

differences [18] of CDMA, the BS is able to identify the total number of different codes, which corresponds to the total number of active MT's during the last frame. This information is made available by TEP for the on-line traffic estimation and prediction described in Section IV.

IMACS is composed of three major components (see Fig. 2): multiple access controller (MACER), traffic estimator/predictor (TEP), and intelligent bandwidth allocator (IBA). It supports four types of traffic—CBR, VBR, ABR, and SCR. IMACS has been designed to satisfy delay guarantees for CBR/VBR traffic while offering minimal access delay for ABR and SCR traffic. Accordingly, MACER employs a reservation-based access protocol for CBR and VBR traffic making use of a fixed amount of R_C-type and R_V-type bandwidth ($R_C + R_V = R$) (in slots), respectively. By contrast, for SCR and ABR traffic, MACER adopts a contention-based access protocol using C_S-type and C_A-type bandwidth ($C_S + C_A = C$) (in slots), respectively. In particular, due to the access-delay-sensitive nature, SCR traffic is particularly governed by contention access using the DTS collision resolution algorithm parameterized by the optimal SD, denoted as DTS-d, if SD $= d$.

IBA then takes responsibility for the static allocation of R-type bandwidth on a call basis and the dynamic allocation of C-type bandwidth on a frame basis. The major focus has been the dynamic allocation of C_S-type and C_A-type bandwidth through determining the optimal SD, aiming at satisfying the minimum ABR throughput and acceptable SCR blocking probability, while retaining maximal aggregate throughput. On behalf of IBA, TEP performs periodic estimation and on-line prediction of ABR traffic characteristics based on past CNF values. Provided with ABR load information in the CNF and the SCR blocking probability requirement, IBA determines the optimal SD prior to every subsequent frame. Once the optimal SD is identified, C_S bandwidth is determined. The remaining bandwidth (C_A) is then allocated to ABR traffic.

III. Multiple Access Controller (MACER)

MACER employs reservation access for CBR and VBR traffic and contention access for SCR and ABR traffic. Specifically, CBR and VBR traffic are statically allocated with fixed amounts of bandwidth (R_C and R_V) for an entire call, satisfying the duty cycle and maximum end-to-end delay requirements, respectively. Due to the allocation simplicity for CBR traffic, further detail is omitted here. In this section, we focus on reservation access for VBR traffic, and contention access, particularly the DTS collision resolution algorithm for SCR traffic.

A. Reservation Access

VBR traffic is assumed to be controlled through a leaky-bucket (ρ, σ) regulator [19], where ρ is the mean leaky rate, and σ is the maximum bucket size, as shown in Fig. 3. Accordingly, a VBR traffic source can be characterized by three parameters (ρ, σ, $D_{\max}$), where $D_{\max}$ is the maximum tolerable end-to-end delay. In the sequel, we derive the minimum bandwidth R_V for VBR traffic satisfying a given end-to-end delay bound, $D_{\max}$. Let $A(s, t)$ denote the total number of packets arriving in a time interval $(s, t]$, and $S(u, v)$ the packets served within the time interval $(u, v]$. Since arriving packets must conform to the (ρ, σ) regulator, $A(0, t) \leq \lceil \rho \cdot t + \sigma \rceil$, where $\lceil\ \rceil$ denotes the ceiling function. Let $\overline{G}$ denote the maximum signaling delay for the establishment of a VBR connection. First, $D_{\max}$ can be given as

$$D_{\max} = \sup_t \{d \geq 0: A(0, t) = S(\overline{G}, t + d)\}. \tag{1}$$

Notice that $S(u, v) = (v - u) \times R_V$. Equation (1) becomes $D_{\max} = \sup_t\{d: 0 \leq d \leq (\lceil \rho \cdot t + \sigma \rceil / R_V) + \overline{G} - t\}$. Since condition $\rho \leq R_V$ must be satisfied, it follows that $D_{\max} = (\lceil \sigma \rceil / R_V) + \overline{G}$, which results in the fixed bandwidth to be allocated to VBR:

$$R_V = \max\left(\frac{\lceil \sigma \rceil}{D_{\max} - \overline{G}}, \rho\right). \tag{2}$$

B. Contention Access

As was previously stated, the contention access protocol is augmented with a DTS collision resolution algorithm. The DTS-d algorithm (if SD $= d$) is described as follows. In

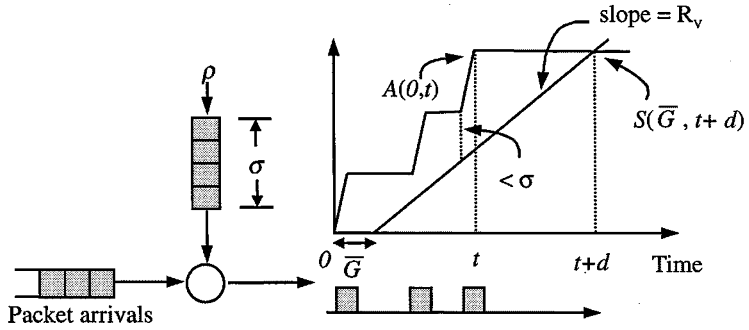


Fig. 3. VBR source traffic model and variables.

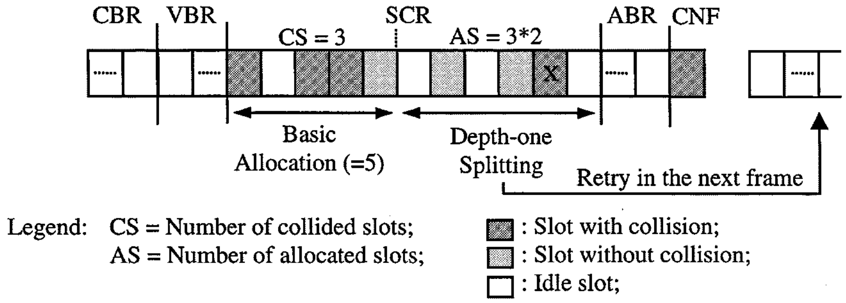


Fig. 4. An example: DTS collision resolution with SD = 1 (DTS-1).

each frame, SCR traffic is initially allocated with the least amount of bandwidth, called the basic allocation (in slots). First, slots from the basic allocation are randomly accessed. Should collisions occur and the number of splitting is less than d, twice as many as the number of collided slots are allocated at the next splitting level. This process repeats until either there is no collision or the number of splitting levels has reached d. All unresolved transmissions then back off in the next frame. It is worth noticing that SCR call requests are not considered blocked until the number of frame backoffs exceeds a predefined threshold, called the retry count (RC).

In Fig. 4, we illustrate an example of the DTS-1 algorithm using 5-slot basic allocation. In the example, due to the presence of 3-slot collisions in the basic allocation, a number of 6 (3 × 2) slots are allocated at the next splitting level. Collision resolution terminates after the depth-one splitting and the unresolved slot (marked "X") will back off in the subsequent frame.

IV. Traffic Estimator/Predictor (TEP)

On behalf of IBA, TEP is responsible for the periodic estimation of the Hurst parameter (denoted as H), and the prediction of the short-term mean and variance of ABR traffic. Specifically, H is periodically estimated based on wavelet analysis [20], [21]. The short-term mean and variance for the subsequent frame are predicted by means of an on-line neural-fuzzy approach [22]. Since the prediction of the variance can be similarly applied, in the sequel we describe the estimation of H and prediction of the short-term mean number of active MT's.

A. Wavelet-Based Traffic Estimation

A self-similar process [23], [24] can be characterized by H, a key measure of self-similarity. Namely, a process $X = \{X_k\colon k = 0, 1, 2, \ldots\}$ is said to be self-similar with parameter $H(0.5 < H < 1)$ if

$$\mathrm{var}(X^{(m)}) = \frac{\mathrm{var}(X)}{m^{2(1-H)}},$$

and

$$r_{X^{(m)}}(k) = r_X(k), \qquad m = 1, 2, \ldots, \tag{3}$$

where $X_k^{(m)} = (1/m)\sum_{i=km-(m-1)}^{km} X_i$, and var and r denote the variance and autocorelation functions, respectively. Considering the multiresolutional wavelet decomposition [21] of a sample function $X(k)$: $X(k) = approx_J(k) + \sum_{m=1}^{J} \sum_n d_{m,n}\psi_{m,n}(k)$, where $approx_J(k)$ represents the approximation of $X(k)$ at the Jth level decomposition, $\psi_{m,n}$ is the orthonormal mother wavelet at resolution m, and coefficient $|d_{m,n}|^2$ measures the amount of energy in the analyzed process at resolution m. Define $\Gamma_m = (1/c_m)\sum_n |d_{m,n}|^2$, where c_m is the number of wavelet coefficients at resolution m. Notice that an important property [20], [21] of self-similar traffic is related to the behavior of the power spectral density at low frequencies: $\Gamma(f) \propto (1/f^{2H-1})$ as $f \rightarrow 0$. We thus obtain the relationship between the amount of energy associated with different resolution planes: $\Gamma_{\mathrm{m}} = 2^{\mathrm{m(2H-1)}}\Gamma_0$. Hence, H can simply be estimated from the slope $(2H - 1)$ of the best-fitting straight line of function $\log(\Gamma_m)$ versus the resolution level, m.

In Fig. 5, we illustrate the estimation of traffic H = 0.8. We discovered that satisfactory estimation requires as few as 10 resolution levels of decomposition, i.e., 2^{10} CNF values. In other words, H can be estimated per every 1024 frames. This fact justifies the viability of frequent estimation of H.

B. Neural-Fuzzy On-Line Traffic Prediction (NFTP)

NFTP performs on-line traffic prediction based on a self-constructing neural-fuzzy inference network [22]. It is involved

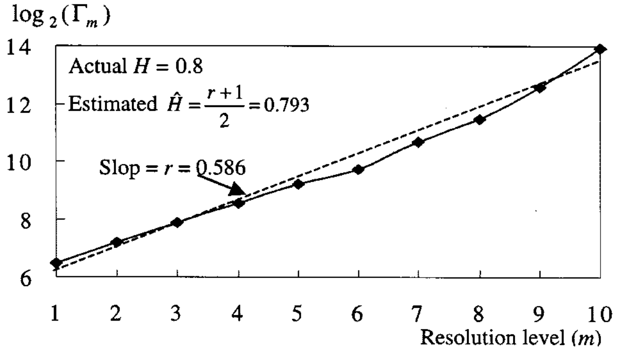


Fig. 5. Hurst-parameter estimation—wavelet analysis.

in two phases of learning: structure and parameter learning. The structure-learning phase determines the structure of fuzzy if–then rules, and the parameter-learning phase tunes the coefficients of the rules adapting to the input traffic dynamics. Unlike existing neural-fuzzy models using sequential learning, NFTP performs the structure and parameter learning in parallel. This makes NFTP advantageous for fast on-line prediction.

NFTP is a six-layer network taking on a number of input nodes and one output node, as shown in Fig. 6. Initially, there are no rules in the network other than input nodes (layer 1) and an output node (layer 6). Upon receiving on-line training data, the structure-learning process proceeds by dynamically self-constructing fuzzy if–then rules (layer 3) according to an input–output clustering-based space-partitioning algorithm [22]. Once a new rule is generated, the centers and widths of the corresponding set of Gaussian membership functions (layer 2 and layer 5) are assigned. The output of a layer 3 node corresponds to the firing strength of the corresponding fuzzy rule, which is in turn normalized in layer 4. Consequently, the predicted output value, y, is given as

$$\begin{aligned} &y = \sum_i y_i, \qquad i = \text{fuzzy rule index, and} \\ &\text{Fuzzy rule } i\text{:} \quad \underline{\text{If}}\ x_1 \text{ is } A_{i1}\ \underline{\text{and}}\ \ldots\ \underline{\text{and}}\ x_n \text{ is } A_{in}, \\ &\qquad\qquad\quad \underline{\text{Then}}\ y_i = f_i m_i \end{aligned} \tag{4}$$

where

y_i contribution of fuzzy rule i to the predicted output value;

x_j jth input value;

A_{ij} jth membership function of fuzzy rule i;

f_i normalized firing strength of fuzzy rule i;

m_i center of the membership function in layer 5 connected to fuzzy rule i.

Meanwhile, in the parameter-learning process, the centers and widths of input membership functions (layer 2) are dynamically adjusted based on the least mean squares (LMS) algorithm [22], whereas those of output membership functions (layer 5) are tuned using the back propagation algorithm [25].

Fig. 6 illustrates an NFTP network with three inputs. This network predicts the future CNF value ($\hat{N}_4$), which corresponds to *the mean number of active MT's in the subsequent frame*, based on three input values taken from three most-recent CNF values (denoted as N_i, $i = 1$ to 3). At the end of each frame, in addition to predicting the CNF value of the next frame, NFTP also performs the learning operation described above. This is indicated in Fig. 6 by the arrowed link pointing from the CNF of Frame 4 to the NFTP output node.

We experimented on two different NFTP structures using different types of inputs, respectively, via simulation. In the first structure, called *CNF-based NFTP*, the inputs are taken directly from a set of different numbers of past CNF values (N_i), ranging from 4 to 24, similar to what is shown in Fig. 6. In the second structure, referred to as *CNF-correlation-based NFTP*, we adopted *exponential-averaging k-lag correlation* of CNF values as inputs. Specifically, taking an example of NFTP with four inputs x_k, $k = 1$ to 4, at the end of the ith frame, x_k will be set as the k-lag correlation $\hat{C}_i$ defined as: $\hat{C}_i = \lambda C_i + (1 - \lambda)\hat{C}_{i-1}$, where $C_i = N_i \times N_{i-k}$, and λ is the smoothing constant ($0 < \lambda < 1$). With this structure, we also carried out 4 to 24 different numbers of inputs. In this simulation, we on-line predicted a set of 200 frames, using both structures of NFTP. All parameters used in the simulation are summarized in Table I. In addition, the performance of NFTP is evaluated in terms of its prediction precision (error rate), time complexity, and space complexity. The error rate was computed as the normalized average deviation between the actual and predicated CNF values. The space complexity was given in terms of the total number of fuzzy rules generated at the end of 200-frame prediction. Notice that since such inference network can be implemented in hardware, we thus disregarded its time complexity. Simulation results are displayed in Table II.

We observed during the experiment that the prediction error rate using either structure is irrelevant to the Hurst parameter (H), but highly sensitive to the variance. This can be perceived by the fact that, by and large, H manifests only long-term behavior, whereas variance greatly reflects short-term fluctuation. In essence, as shown in Table II under traffic $H = 0.8$, the error rate greatly increases with the variance. Furthermore, compared to CNF-based NFTP, CNF-correlation-based NFTP achieves greater precision (lower error rate) and lower space complexity (less number of fuzzy rules). We finally discovered in the table that NFTP (either structure) with 12 inputs invariably exhibits better performance under both variances. Namely, small or large numbers of inputs yield inferior performance for on-line prediction.

Moreover, we conducted another experiment via simulation to further demonstrate the viability of correlation-based NFTP for off-line training. In the simulation, we off-line trained a sample path of self-similar traffic with 100 CNF values, for three rounds. The traffic was generated with mean = 50, variance = 60, and H = 0.6 and 0.8. The NFTP network takes on 12 inputs. In addition, $\overline{F}_{in} = 0.04$, $\overline{F}_{out} = 1$, learning constant $\eta = 0.0005$, and $\beta = 0.7$. The LMS algorithm was replaced by the recursive least squares (RLS) algorithm [22]. Simulation results are sketched in Fig. 7. In the figure, we make comparisons between actual and predicted CNF's under both H = 0.6 and 0.8. We discovered that the off-line trained NFTP network achieves superior prediction precision at the expense of higher space complexity under both H values. Furthermore, the more fuzzy rules, the better the precision, irrelevant to H.

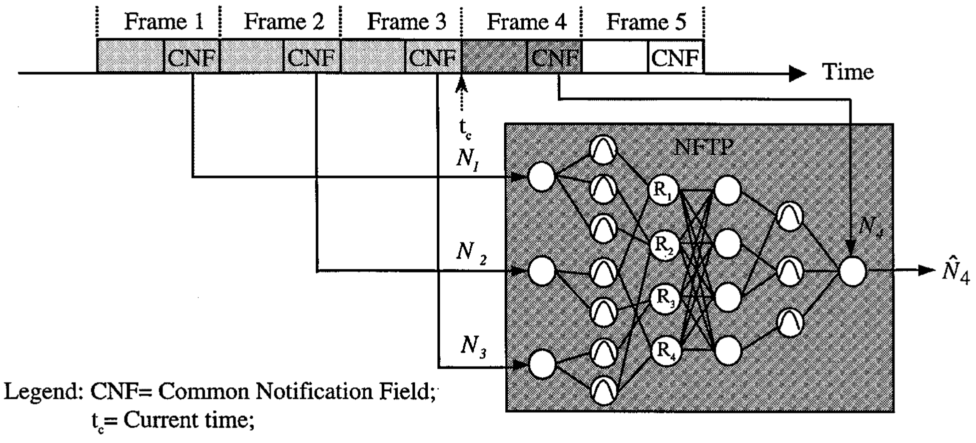


Fig. 6. NFTP architecture.

TABLE I
NFTP PARAMETERS USED IN SIMULATION

Variable	Definition	Value
$\overline{F}_{in}$	Input clustering threshold	0.1
$\overline{F}_{out}$	Output clustering threshold	0.99
$\rho(t)$	Membership function threshold	0.7
β	Initial width of Gaussion function	0.25
η	Back-propagation learning constant	0.08
λ	Smoothing constant	0.5

TABLE II
PERFORMANCE OF NFTP USING TWO DIFFERENT STRUCTURES

H = 0.8	CNF-based NFTP			CNF-correlation-based NFTP		
	Number of inputs	Number of fuzzy rules	Error rate	Number of inputs	Number of fuzzy rules	Error rate
Mean = 50 Variance = 20	4	27	6.1	4	6	5.5
	8	33	6.0	8	12	5.6
	12	42	5.9	12	23	5.4
	16	47	6.6	16	20	5.5
	20	54	6.7	20	28	5.3
	24	55	7.3	24	23	5.6
Mean = 50 Variance = 60	4	27	10.9	4	11	9.7
	8	33	10.7	8	20	9.5
	12	42	10.6	12	17	9.3
	16	47	11.8	16	20	9.8
	20	54	12.1	20	20	9.7
	24	55	13.1	24	23	9.9

V. INTELLIGENT BANDWIDTH ALLOCATOR (IBA)—DETERMINATION OF OPTIMAL SD

A. Design Principle

The bandwidth allocation problem can be elucidated by the following dilemma. We observed that greater SD values yield appealing SCR blocking probability, but at the expense of penalized ABR throughput. Nevertheless, smaller SD values still render unfavorable ABR throughput and aggregate throughput despite the price of increasing SCR blocking probability paid. Therefore, the objective of IBA has been the determination of the optimal SD per every frame, aiming at satisfying SCR blocking probability and ABR throughput requirements, while retaining maximal aggregate throughput. In short, IBA has been designed to provide optimal allocation between $\mathrm{C_S}$ and $\mathrm{C_A}$ types of bandwidth.

To this end, we performed both precise and simulation-based throughput analyses. In both analyses, SCR traffic is invariantly assumed Poisson distributed. ABR traffic is first simplified as Poisson distributed in the precise throughput analysis. ABR traffic is then practically modeled as self-similar in the simulation-based throughput analysis. The generated throughput results then postulate the optimal SD's under various traffic conditions. These results are then off-line trained and constructed using a back propagation neural network (BPNN) [25] which is used on-line by IBA. Without loss of generality, we assume that the number of slots in the aggregate bandwidth $(\mathrm{C_S} + \mathrm{C_A})$ remains a constant throughout this section.

B. Precise Throughput Analysis

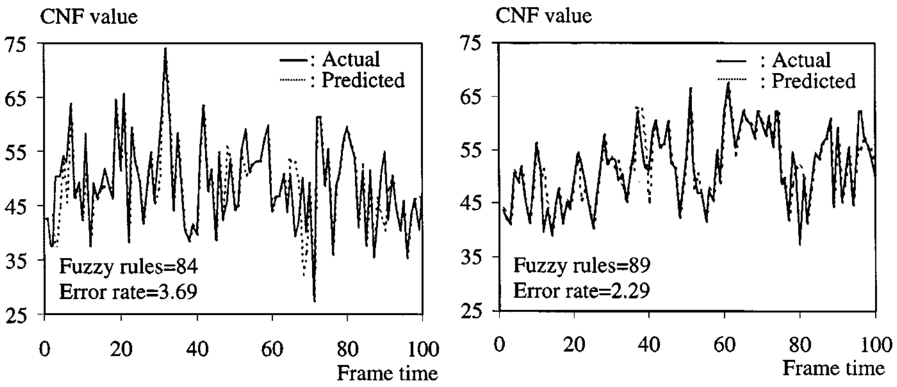
In this subsection, the *aggregate throughput* is derived under two cases: $\mathrm{SD} = 0$, and $\mathrm{SD} = 1$. Analyses for higher SD values can be similarly applied. Variables used throughout the analysis are summarized in Table III.

The aggregate throughput, denoted as T, is defined as the ratio of the mean number of successful slots for SCR and ABR cell transmissions to the total number of slots in a frame (L). Namely,

$$T = \frac{E[\tilde{s}_S] + E[\tilde{s}_A]}{L}. \tag{5}$$

Case 1—$\mathrm{SD} = 0$*:* $\mathrm{SD} = 0$ corresponds to no splitting. Therefore, the total numbers of slots allocated to SCR and ABR traffic are constants, namely L_{S0} and $L - L_{S0}$, respectively. Moreover, for each of a total of k SCR Poisson arrivals (cells), the probability of successful transmission is $(1 - (1/L_{S0})^{k-1}$. Thus,

$$\begin{aligned} E[\tilde{s}_S] &= \sum_{k=0}^{\infty}\{E[\tilde{s}_S|\tilde{n}_S = k] \cdot P[\tilde{n}_S = k]\} \\ &= \sum_{k=0}^{\infty}\left\{k \cdot \left(1 - \frac{1}{L_{S0}}\right)^{k-1} \cdot \frac{e^{-E[\tilde{n}_S]}E[\tilde{n}_S]^k}{k!}\right\} \\ &= E[\tilde{n}_S] \cdot e^{((-E[\tilde{n}_S])/L_{S0})}. \end{aligned} \tag{6}$$



(a) Traffic H=0.6

(b) Traffic H=0.8

Fig. 7. Comparisons of actual and predicated CNF values. (a) Traffic H = 0.6. (b) Traffic H = 0.8.

TABLE III
VARIABLES USED THROUGHOUT THE ANALYSIS

Variable	Definition
T	Aggregate throughput (C-type bandwidth)
L	Total number of (C-type) slots per frame
$\tilde{s}_S$	Number of successful SCR slots per frame
$\tilde{s}_A$	Number of successful ABR slots per frame
L_{S0}	Size of basic allocation (in slot) for SCR traffic
$\tilde{n}_S$	Number of newly-arriving SCR cells per frame
$\tilde{n}_A$	Number of newly-arriving ABR cells per frame
$\tilde{c}_S$	Number of collided SCR cells

Similarly, $E[\tilde{s}_A]$ for ABR traffic using a total number of $L-L_{S0}$ remaining slots, can be given as

$$E[\tilde{s}_A] = E[\tilde{n}_A] \cdot e^{(-E[\tilde{n}_A])/(L-L_{S0})}. \tag{7}$$

From (5)–(7), T can be directly obtained.

Case 2—SD $= 1$*:* Since there is one level of collision resolution in this case, the total numbers of $\mathrm{C_S}$ and $\mathrm{C_A}$ slots in each frame are no longer constants. With SCR and ABR throughput jointly considered, (5) becomes (8), as shown at the bottom of the page.

To compute the conditional mean number of successful slots in (8), we consider two contention results prior to the first splitting: no collision, and collisions of l SCR cells, $1 \leq l \leq k$. In addition, random variable $\tilde{c}_S$ is independent of $\tilde{n}_A$. We thus get

$$\begin{aligned} &E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j] \\ &\quad = E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j,\, \tilde{c}_S = 0] \\ &\qquad \cdot P[\tilde{c}_S = 0 | \tilde{n}_S = k] \\ &\qquad + \sum_{l=1}^{k} \{E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j,\, \tilde{c}_S = l] \\ &\qquad\qquad \cdot P[\tilde{c}_S = l | \tilde{n}_S = k]\}. \end{aligned} \tag{9}$$

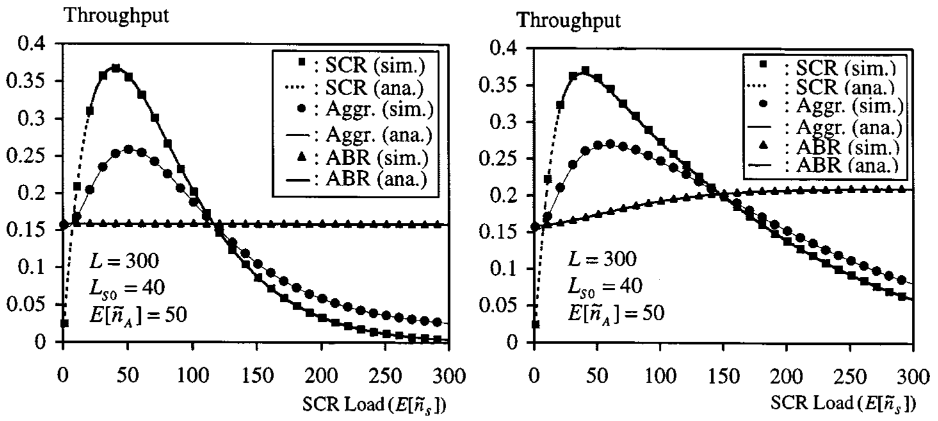
The conditional mean in the first term of (9) can simply be given by the sum of k successful SCR slots and mean successful ABR slots. Namely,

$$\begin{aligned} &E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j,\, \tilde{c}_S = 0] \\ &\quad = k + j\left(1 - \frac{1}{L - L_{S0}}\right)^{j-1}. \end{aligned} \tag{10}$$

To compute the conditional mean in the second term of (9), we first consider the total number of slots with collisions before the first splitting. For instance, if there are m slots with collisions, there will be $2m$ slots allocated for SCR traffic during the first splitting, and a number of $L - L_{S0} - 2m$ remaining slots allocated for ABR traffic. As a result,

$$\begin{aligned} &E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j,\, \tilde{c}_S = l] \\ &= \sum_{m=1}^{\min(L_{S0},\,(l/2))} \{E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j,\, \tilde{c}_S = l, \\ &\qquad \tilde{c} = m] \cdot P[\tilde{c} = m | \tilde{n}_S = k,\, \tilde{c}_S = l]\} \\ &= \sum_{m=1}^{\min(L_{S0},\,(l/2)} \left\{ \left(k - l + l\left(1 - \frac{1}{2m}\right)^{l-1}\right) \right. \\ &\qquad + j\left(1 - \frac{1}{L - L_{S0} - 2m}\right)^{j-1} \\ &\qquad \left. \cdot P[\tilde{c} = m | \tilde{n}_S = k,\, \tilde{c}_S = l] \right\}. \end{aligned} \tag{11}$$

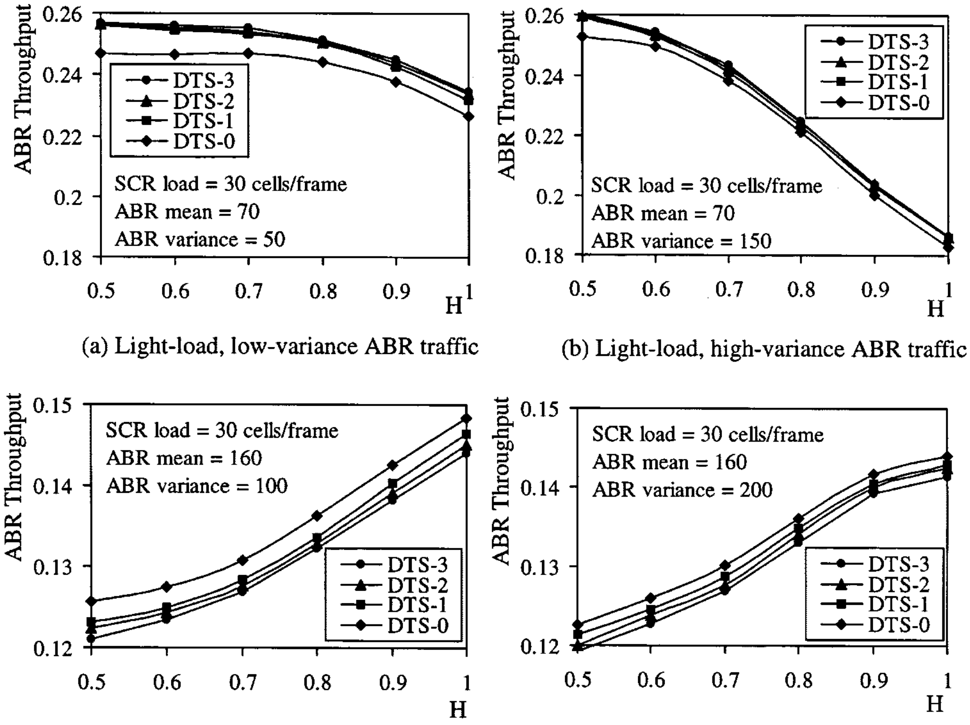
$$\begin{aligned} T &= \frac{\sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\{E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j] \cdot P[\tilde{n}_S = k,\, \tilde{n}_A = j]\}}{L} \\ &= \frac{\sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\left\{E[\tilde{s}_S + \tilde{s}_A | \tilde{n}_S = k,\, \tilde{n}_A = j] \cdot \frac{e^{-E[\tilde{n}_S]}E[\tilde{n}_S]^k}{k!} \cdot \frac{e^{-E[\tilde{n}_A]}E[\tilde{n}_A]^j}{j!}\right\}}{L} \end{aligned} \tag{8}$$



(a) SD=0

(b) SD=1

Fig. 8. Analytical and simulation results of throughput. (a) SD = 0. (b) SD = 1.



(a) Light-load, low-variance ABR traffic

(b) Light-load, high-variance ABR traffic

(c) High-load, low-variance ABR traffic

(d) High-load, high-variance ABR traffic

Fig. 9. ABR throughput versus H. (a) Light-load, low-variance ABR traffic. (b) Light-load, high-variance ABR traffic. (c) High-load, low-variance ABR traffic. (d) High-load, high-variance ABR traffic.

Next, we compute the conditional probability in (9), namely, $P[\tilde{c}_S = l|\tilde{n}_S = k]$. Define function $F(l, i)$ as the number of arrangements such that a number of l SCR cells undergo collisions within i slots. Since there exist at least two SCR cells in each collision, $F(l, i)$ can be formulated as

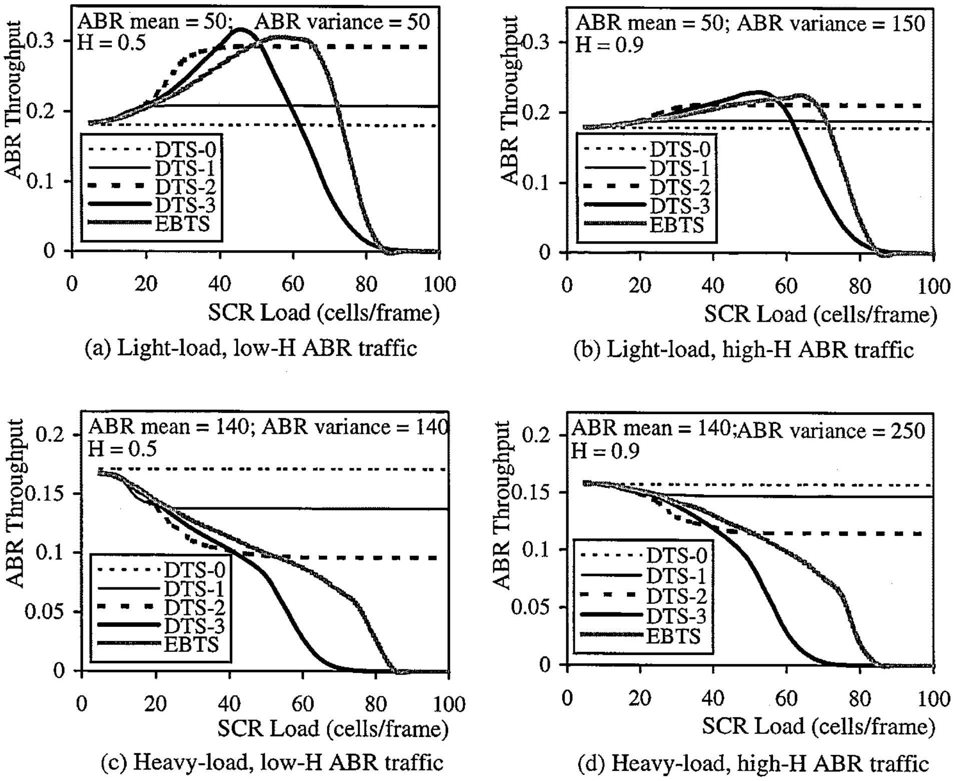
$$F(l, i) = \sum_{\substack{n_1, n_2 \cdots n_i \geq 2 \\ n_1 + n_2 + \cdots + n_i = l}} \frac{l!}{n_1! n_2! \cdots n_i!}. \tag{12}$$

Furthermore, $P[\tilde{c}_S = 1|\tilde{n}_S = k]$ is equal to zero. Also, $P[\tilde{c}_S = l|\tilde{n}_S = k]$ is equal to zero if the total number of slots with successful transmissions ($=k - l$) exceeds the size of basic allocation. Accordingly, we get (13) as shown at the bottom of the next page. With function $F(l, i)$ defined, $P[\tilde{c} = m|\tilde{n}_S = k, \tilde{c}_S = l]$ in (11) can simply be expressed as

$$P[\tilde{c} = m|\tilde{n}_S = k, \tilde{c}_S = l] = \frac{F(l, m)}{\sum\limits_{i=1}^{\min(L_{SO}-(k-l),(l/2))} F(l, i)} \tag{14}$$

Finally, aggregate throughput T can be directly derived from (8)–(14).

To demonstrate the validity of this analysis, we also carried out event-based simulation under both SD = 0 and 1 cases. The simulation was terminated after the execution of a total of 10^5 frames at which the system has reached its steady state. Fig. 8 depicts the analytical and simulation results of SCR, ABR, and



(a) Light-load, low-H ABR traffic

(b) Light-load, high-H ABR traffic

(c) Heavy-load, low-H ABR traffic

(d) Heavy-load, high-H ABR traffic

Fig. 10. ABR throughput versus SCR load. (a) Light-load, low-H ABR traffic. (b) Light-load, high-H ABR traffic. (c) Heavy-load, low-H ABR traffic. (d) Heavy-load, high-H ABR traffic.

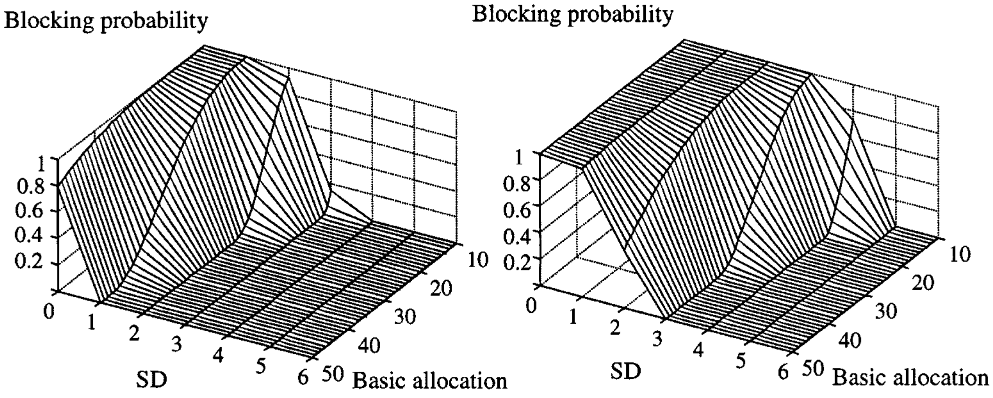
aggregate throughput, under a light ABR load (50 cells/frame). It is worth mentioning that we observed more trivial and unsurprising results under higher ABR loads. These results are thus omitted here. First, analytical results are shown to be in profound agreement with simulation results. Surprisingly, compared to $\mathrm{SD} = 1$, the $\mathrm{SD} = 0$ case invariantly yields poorer SCR and ABR throughput, resulting from the waste of unused remaining bandwidth for ABR traffic with relatively light load.

C. Simulation-Based Throughput Analysis

We adopted the fractional Gaussian noise (FGN) process [23], [24] and a fast-generation algorithm [26], for the modeling and generation of self-similar traffic, respectively. Particularly for traffic generation, we considered a set of ten slots each time for generating a nonnegative number of cell arrivals. For managing negative arrivals, alternative approaches can be found in [27] and [28]. Given a mean arrival, we first randomly generated a number, which represents the total number of arriving cells, in each group of ten slots. The exact arriving epochs of these cells were then uniformly distributed in ten slots.

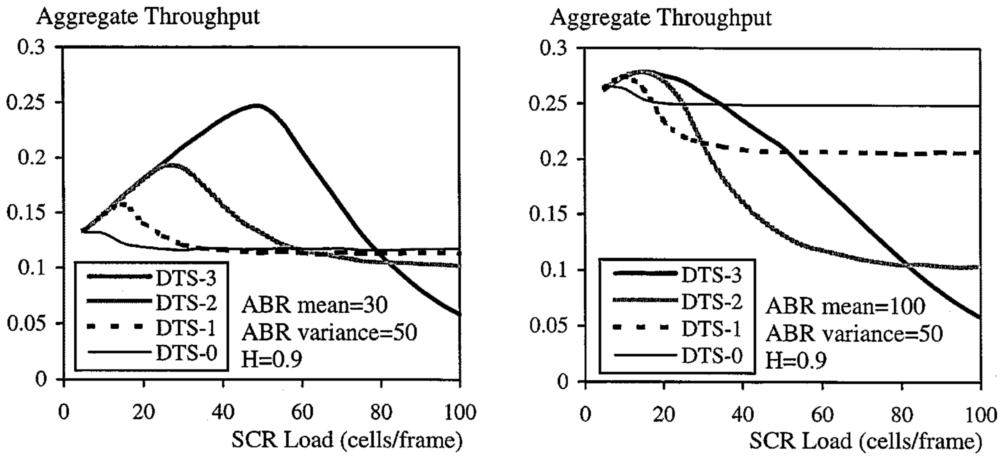
As was previously mentioned, SCR and ABR cells are handled differently with respect to the backoff policy. Collided ABR cells back off in the next frame. Collided SCR cells (calls) back off a maximum of d times within a frame provided that $\mathrm{SD} = d$. Failed calls keep retrying the next frame until reaching the predetermined number of frames, namely the RC. Failed calls are at that moment considered blocked. Notice that the RC is inferred from the maximum tolerable call-setup delay. In our simulation, the RC was set to five (frames) corresponding to a maximum call-setup delay of 50 ms. Moreover, it is required to impose limits on the basic-allocation size and maximum SD value so that the total amount of $\mathrm{C_S}$ bandwidth never exceeds the frame size. In the simulation, for a frame of 300 slots in

$$P[\tilde{c}_S = l | \tilde{n}_S = k] = \begin{cases} \dfrac{\dbinom{k}{l} \displaystyle\sum_{i=1}^{\min(L_{S0}-(k-l),(l/2))} \dbinom{L_{S0}}{i} F(l,\, i) \dfrac{(L_{S0}-i)!}{(L_{S0}-i-(k-l))!}}{L_{S0}^{k}} & \text{if } (l \neq 1) \text{ and } \\ & \{(l = 0) \text{ or } (k - l \le L_{S0} - 1)\} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$



(a) Light SCR load (30 cells/frame) (b) Heavy SCR load (90 cells/frame)

Fig. 11. Blocking probability for SCR traffic. (a) Light SCR load (30 cells/frame). (b) Heavy SCR load (90 cells/frame).



(a) Light ABR load (b) Heavy ABR load

Fig. 12. Aggregate network throughput. (a) Light ABR load. (b) Heavy ABR load.

length, the size of the basic allocation ranged from 10 to 50 slots, and the maximum eligible SD value was set as 6.

1) Satisfaction of ABR QoS-ABR Throughput: We experimented on ABR throughput under a variety of ABR traffic based on the DTS-$d (d = 0, 1, 2, 3)$ collision resolution algorithm. Simulation results are displayed in Fig. 9. As shown in Fig. 9(a) and (b), under light ABR loads, ABR throughput declines with increasing H. This phenomenon can be perceived by the fact that greater H corresponds to higher burstiness, resulting in more collisions. The situation deteriorates under higher-variance conditions, as depicted in Fig. 9(b). However, we surprisingly discovered from Fig. 9(c) and (d) that ABR throughput increased with H under heavy loads. This is due to the fact that heavy-load and lower-H traffic yields a large amount of cells to be evenly distributed among slots, causing collisions everywhere.

We further investigated the impact of different DTS-d algorithms on ABR throughput. We once more discovered incompatible performance under light and heavy ABR loads. That is, ABR throughput increases with d under light loads, but decreases with d under heavy loads. This is because under high bandwidth demand, there appears to be a clear tradeoff between SCR and ABR throughput. But, in contrast, under light loads or low bandwidth demand, ABR throughput no longer benefits from decreasing SD, as exhibited in Fig. 9(a) and (b).

Moreover, in Fig. 10, we draw comparisons of ABR throughput versus the SCR load. In the experiment, we employed four DTS-d variants and the traditional exhaustive binary-tree-splitting (EBTS) collision resolution algorithm. Notice that the EBTS algorithm corresponds to DTS-4 in this case, resolving collisions up to the entire bandwidth within a frame. As shown in Fig. 10(a) and (b) under light ABR loads, DTS-2 outperforms other DTS and EBTS algorithms. As the SCR load increases reaching a turning point, which is located distinctively under different algorithms, ABR throughput starts declining. Among all approaches, DTS-3 and EBTS undergo the most deteriorating performance. In addition, higher variance and H traffic results in inferior throughput [see Fig. 10(b)]. On the other hand, under heavy ABR loads as shown in Fig. 10(c) and (d), ABR throughput invariably declines with increasing SCR load in all algorithms. The greater the SD, the poorer the throughput. Specifically, DTS-0 achieves the best performance among all algorithms due to the provision of a fixed amount of bandwidth to ABR traffic despite the increase in the SCR load.

2) Satisfaction of SCR QoS-Blocking Probability: We now discuss (from Fig. 11) the sensitivity of SCR blocking probability with respect to the SD value and basic allocation, under light and heavy SCR traffic loads. As was expected, blocking probability declines with increasing basic allocation and SD value. Specifically, heavier loads [see Fig. 11(b)] demand greater SD values to achieve the same grade of blocking probability. For example, to achieve nonblocking, SD $= 4$ and SD $= 5$ are required under light and heavy SCR loads, respectively.

3) Maximization of Aggregate Throughput: We finally examine the aggregate throughput under various traffic and SD values. In Fig. 12 we depict the aggregate throughput as a function of SCR load under light and heavy ABR loads, using four variants of the DTS algorithm. Initially starting from a light SCR and ABR load in Fig. 12(a), greater SD values unsurprisingly achieve better throughput. However, as the SCR load increases, greater SD values can no longer benefit the aggregate throughput resulting from substantial unresolved collision. At this moment, greater SD values yield more bandwidth waste, leading to poorer throughput. The turning point again is located differently for different variants of the DTS algorithm. Under a heavy ABR traffic load shown in Fig. 12(b), we observed consistent plots which, however, exhibit earlier turning points owing to the contribution of the heavy ABR load.

Accordingly, the optimal SD is dependent on four traffic characteristics: ABR mean load, variance, the Hurst parameter, and the SCR mean load. As was previously stated, these results are then off-line trained and constructed via a BPNN, which can be effectively accessed on-line by IBA offering optimal bandwidth allocation.

VI. Conclusion

In this paper, we proposed an integrated system, IMACS, facilitating a hybrid-TDM-based MAC protocol and dynamic bandwidth allocation, via three components—MACER, TEP, and IBA. Unlike existing protocols, MACER particularly employs dynamic-tree-splitting collision resolution parameterized by the optimal SD. With estimation and prediction of ABR self-similar traffic through TEP, IBA provides efficient bandwidth allocation by determining the optimal SD, achieving satisfactory SCR blocking probability and ABR throughput requirements, while retaining maximal aggregate throughput. Analytical and simulation results demonstrated that the optimal SD is highly dependent on ABR mean, variance, the Hurst parameter, and SCR mean. The dependency, which is often contrary under different traffic settings, can be off-line constructed using a BPNN.

Acknowledgment

The authors would like to thank the editors and anonymous reviewers for the helpful comments and suggestions that greatly improved the presentation of the paper.

References

[1] D. Cox, "Wireless personal communications: What is it?," *IEEE Personal Commun.*, vol. 2, pp. 20–35, Apr. 1995.

[2] D. Raychaudhuri and N. Wilson, "ATM-based transport architecture for multiservices wireless personal communication," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1401–1414, Oct. 1994.

[3] D. Raychaudhuri, L. French, R. Siracusa, S. Biswas, R. Yuan, P. Narasimhan, and C. Johnston, "WATMnet: A prototype wireless ATM system for multimedia personal communication," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 83–95, Jan. 1997.

[4] N. Wilson, R. Ganesh, K. Joseph, and D. Raychaudhuri, "Packet CDMA versus dynamic TDMA for multiple access in an integrated voice/data PCN," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 870–883, Aug. 1993.

[5] R. Rom and M. Sidi, *Multiple Access Protocols—Performance and Analysis*. New York: Springer-Verlag, 1990.

[6] M. Arad and A. Leon-Garcia, "A generalized processor sharing approach to time scheduling in hybrid CDMA/TDMA," in *Proc. IEEE INFOCOM*, 1998, pp. 1164–1170.

[7] M. McTiffin, A. Hulbert, T. Ketseoglou, W. Heimsch, and G. Crisp, "Mobile access to an ATM network using a CDMA air interface," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 900–908, June 1994.

[8] D. Goodman, R. Valenzuela, K. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, pp. 885–890, Aug. 1989.

[9] J. Kim and I. Widjaja, "PRMA/DA: A new media access control protocol for wireless ATM," *Proc. IEEE ICC*, pp. 240–244, 1996.

[10] M. Listanti, F. Mascitelli, and A. Mobilia, "D^2MA: A distributed access protocol for wireless ATM networks," in *Proc. IEEE INFOCOM*, 1998, pp. 315–321.

[11] B. Paris and B. Aazhang, "Near-optimum control of multiple-access collision channels," *IEEE Trans. Commun.*, vol. 40, pp. 1298–1309, Aug. 1992.

[12] G. Polyzos and M. Molle, "A queueing theoretic approach to the delay analysis for the FCFS 0.487 conflict resolution algorithm," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1887–1906, Nov. 1993.

[13] A. Bar-David and M. Sidi, "Collision resolution algorithms in multistation packet-radio networks," *IEEE Trans. Commun.*, vol. 37, pp. 1387–1391, Dec. 1989.

[14] D. Levine, I. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 1–12, Feb. 1997.

[15] A. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE/ACM Trans. Networking*, vol. 6, pp. 635–644, Oct. 1998.

[16] P. Narasimhan and R. Yates, "A new protocol for the integration of voice and data over PRMA," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 623–631, May 1996.

[17] M. Karol, Z. Liu, and K. Eng, "Distributed-queueing request update multiple access (DQRUMA) for wireless packet (ATM) networks," in *Proc. IEEE ICC*, 1995, pp. 1224–1231.

[18] J. Lehnert and M. Pursley, "Error probabilities for binary direct-sequence spread-spectrum communications with random signature sequences," *IEEE Trans. Commun.*, vol. COM-35, pp. 87–98, Jan. 1987.

[19] C. Chang, K. Chen, M. You, and J. Chang, "Guaranteed quality-of-service wireless access to ATM networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 106–117, Jan. 1997.

[20] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2–15, Jan. 1998.

[21] S. Giordano, S. Miduri, M. Pagano, F. Russo, and S. Tartarelli, "A wavelet-based approach to the estimation of the Hurst parameter for self-similar data," in *Proc. DSP*, 1997, pp. 479–482.

[22] C. Jung and C. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12–32, Feb. 1998.

[23] J. Beran, "Statistical methods for data with long-range dependence," *Stat. Sci.*, vol. 7, no. 4, pp. 404–427, 1992.

[24] J. Beran, R. Sherman, M. Taqqa, and W. Willinger, "Long-range dependence in variable bit rate video traffic," *IEEE Trans. Commun.*, vol. 43, pp. 1566–1579, Feb./Mar./Apr. 1995.

[25] M. Yuang, P. Tien, and S. Liang, "Intelligent video smoother for multimedia communications," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 136–146, Feb. 1997.

[26] V. Paxson, "Fast, approxmate synthesis of fractional Gaussian noise for generating self-similar network traffic," in *Proc. ACM/SIGCOMM*, 1997, pp. 5–18.

[27] R. Addie, M. Zukerman, and T. Neame, "Broadband traffic modeling: Simple solutions to hard problems," *IEEE Commun. Mag.*, vol. 36, pp. 88–95, Aug. 1998.

[28] R. Addie, D. Platt, and M. Zukerman, "Performance of a Pi persistence protocol subject to correlated gaussian traffic," in *Proc. IEEE INFOCOM*, 1996, pp. 263–270.

Maria C. Yuang (M'91) received the B.S. degree in applied mathematics from the National Chiao Tung University, Taiwan, in 1978; the M.S. degree in computer science from the University of Maryland, College Park, in 1981; and the Ph.D. degree in electrical engineering and computer science from the Polytechnic University, Brooklyn, NY, in 1989.

From 1981 to 1990, she was with AT&T Bell Laboratories and Bell Communications Research (Bellcore), where she was a member of Technical Staff working on high speed networking and protocol engineering. She was also an Adjunct Professor at the Department of Electrical Engineering, Polytechnic University, during 1989–1990. In 1990, she joined National Chiao Tung University, Taiwan, where she is currently a Professor of the Department of Computer Science and Information Engineering. Her current research interests include high speed networking, multimedia communications, and performance analysis.

Po L. Tien was born in Taiwan in 1969. He received the B.S. degree in applied mathematics, and the M.S. degree in computer and information science, from the National Chiao Tung University, Taiwan, in 1992 and 1995, respectively.

He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering of the same university. His current research interests include high speed networking, multimedia communications, performance analysis, and applications of neural-fuzzy networks.