The Jitter Time-Stamp Approach for Clock Recovery of Real-Time Variable Bit-Rate Traffic

Weilian Su, *Student Member, IEEE,* and Ian F. Akyildiz, *Fellow, IEEE*

***Abstract*—When multimedia streams arrive at the receiver, their temporal relationships may be distorted due to jitter. Assuming the media stream is packetized, the jitter is then the packet's arrival time deviation from its expected arrival time. There are various ways to reduce jitter, which include synchronization at the application layer, or synchronization at the *asynchronous transfer mode* (ATM) *adaptation layer* (AAL). The new source rate recovery scheme called *jitter time-stamp* (JTS) provides synchronization at the *ATM adaptation layer 2* (AAL2) which is used to carry variable bit-rate traffic such as compressed voice and video. JTS is implemented, and experiments have shown that it is able to recover the source rate.**

***Index Terms*—ATM adaptation layer 2 (AAL2), ATM networks, constant bit-rate (CBR), synchronization, timing recovery, variable bit-rate (VBR).**

I. Introduction

CLOCK recovery at the ATM Adaptation Layer 1 (AAL1) utilizes the synchronous residual time-stamp (SRTS) method [14], which is designed for constant bit-rate (CBR) traffic carried by AAL1 packets. The SRTS method maps the service clock to the network clock to determine the packet's residual timing information. This information is used by the receiver with the network clock to reconstruct the source rate.

The pulse stuffing and the absence of a network clock in SRTS [14], [13] contribute additionally to the jitter, which is the packet's arrival time deviation from its expected arrival time.

Pulse stuffing introduces jitter even at undesirable low frequencies depending on the pulse stuffing ratio between the network clock and the source clock. Research has been conducted to determine pulse stuffing ratios which minimize the waiting time jitter [12] (or pulse stuffing jitter in conventional stuffing synchronizer). Since the receiver depends on the network clock to reconstruct the source clock, a slight deviation of the network clock will create noise in the reconstructed source clock.

A new scheme is proposed in [15] to reconstruct the source frequency in the AAL1 layer. The scheme uses a buffer control technique which changes the bandwidth of the low pass filter according to the filling level of the buffer and the statistical characteristic of the jitter.

To date, to our knowledge, there is no source rate recovery scheme for AAL2 [10]. AAL2 is mainly used to carry variable bit-rate (VBR) traffic such as compressed voice and video. As mentioned before, the SRTS method is used for CBR traffic and is not well-suited for VBR traffic, because VBR has variable ON and OFF burst lengths. Since compressed multimedia traffic is inherently bursty, the best way to transmit the compressed multimedia traffic is through VBR connections. By this way, more users can be supported at the same time and the bandwidth gain increases significantly due to statistical multiplexing. In particular, the bandwidth gain of multiplexing VBR sources into AAL2 packets has been demonstrated in [18]–[21].

In this paper, we introduce a new source rate recovery scheme, called the jitter time-stamp (JTS) approach which will be used in AAL2 layer. Recovering the source rate at the AAL2 layer has the following advantages.

- The computational load of the application layer can be reduced.
- There is no need for traffic smoothing since traffic is sent as VBR.
- VBR video coding is allowed and relatively stable picture quality can be achieved.
- VBR sources can be multiplexed to achieve high bandwidth gain while QoS requirements can still be satisfied.

This paper is organized as follows. In Section II, we present an overview of the new source rate recovery scheme called JTS. In Section II-A, we describe the initialization steps necessary to obtain the round-trip delays measured at the source and receiver, and the reference network delay. In Section II-B, we explain how packets obtain their time indication (*TI*) at the source, and in Section II-C, we describe how timing packets are dejittered at the receiver. The packet(s) with timing information will be referred as timing packet(s) throughout the paper. Also, the terms data and timing data will be used instead of packet's payload and timing packet's payload, respectively, throughout the paper. Due to the changes in the network condition and drifting reference clocks, packets' jitters are biased. This situation is also addressed. After incoming timing packets are dejittered, the source rate is recovered with the source rate recovery scheme. In Section III, we evaluate the performance of the JTS scheme. In Section IV, we conclude the paper.

II. Jitter Time-Stamp (JTS) Approach

First, we discuss the characteristics of VBR traffic and the difference between source rate and reference clock. A burst in a VBR traffic is defined as a duration of data being sent to the receiver. The VBR traffic could be classified as two types, *dependent* and *independent* burst traffic. The dependent burst traffic is when the size of the burst varies while the interburst time is

Manuscript received November 17, 2000; revised April 28, 2001; recommended by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Low. This work was supported by NASA-Ames under Contract NAG 2-1262.

The authors are with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: weilian@ee.gatech.edu; ian@ee.gatech.edu).

Publisher Item Identifier S 1063-6692(01)10550-9.

constant, e.g., a video stream. The interburst time is the time between two bursts of data. The independent burst traffic is when both the size of the burst and the interburst time vary. Telephony is an example of this. Independent burst traffic is more tolerable to the playout delay of a burst. For example, a voice talkspurt could be delayed while the listener does not find it annoying. On the other hand, a video stream burst is very sensitive to the playout delay. If a frame is delayed, it will cause jitter movements in the video. The CBR traffic is also another example of dependent burst traffic, where the size of burst and interburst time are constant.

The JTS recovers the source rate rather than the reference clock to maintain a temporal relationship between consecutive packets of a real-time VBR stream. Unlike the CBR traffic, the VBR traffic has a difference in the meaning between source rate and reference clock. The reference clock is a clock used by the application to send a packet to the AAL on every clock tick. The source rate is the number of packets sent to the AAL per second since the VBR traffic consists of ON and OFF periods.

Unlike the SRTS which encodes the source rate by a reference clock derived from the network clock and transmits the source rate to the receiver, JTS recovers the source rate by removing AAL packet's network jitter, removing clock drift between source and receiver, and detecting ON/OFF periods of the VBR traffic at the receiver. As a result, JTS does not depend on the network clock. The receiver is driven by the reference clock frequency that is set and sent by the source.

The JTS approach has the following procedures.

- *Initialization*.
- *Source Behavior:* How timing information and sequence numbers are inserted into data packets, which are sent to the receiver. This information is needed by the receiver to reconstruct the VBR traffic's source rate.
- *Receiver Behavior:* How VBR traffic's source rate is reconstructed at the receiver with the timing information, sequence number, estimated jitter, and the waiting time for the next packet from the source.

A. Initialization

Before VBR traffic is sent to the receiver, initialization between source and receiver is performed. Both the source and receiver send a packet to acquire the round-trip delay of the network. D_{S} and D_{R} are the round-trip delays measured at the source and receiver, respectively, in terms of reference clock cycles. The source and receiver then communicate their round-trip delays, D_{S} and D_{R}, and select the reference network delay D_{ref}.

$$D_{\mathrm{ref}} = 1/2 \cdot \max\{D_{\mathrm{S}}, D_{\mathrm{R}}\}. \tag{1}$$

The VBR traffic has the typical ON (active) and OFF (passive) periods. We assume packets are sent at source rate SRA during the active period. The source rate for active period SRA is also sent to the receiver by the source as

$$\mathrm{SRA} = \frac{\gamma}{\eta} \text{ (packets/s)} \tag{2}$$

where γ is the source rate in bytes/s and η is the number of information bytes per packet. For example, a voice talkspurt (an active period) is packetized with η information bytes per packet and sent at SRA while otherwise nothing is sent. Each packet is assumed to be fixed size.

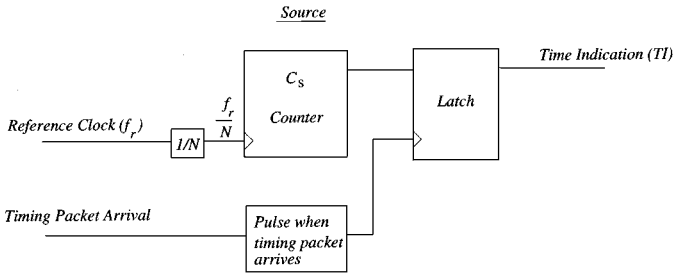

Fig. 1. Time indication.

The receiver can then use the SRA value to calculate the reference clock frequency f_{r} used at the AAL level. f_{r} is the reference clock frequency in hertz (Hz) with which a packet is processed at each f_{r} clock tick. For example, if SRA is 400 packets/s, then f_{r} is 400 Hz. f_{r} is used by the source to send a packet from the application layer to AAL with each clock tick. The period of the reference clock frequency f_r is then

$$T_{\mathrm{r}} = \frac{1}{f_{\mathrm{r}}} \text{ (s)}. \tag{3}$$

Also, the value N is sent to the receiver by the source. N is the number of reference clock cycles per C_{s} counter increment as shown in Fig. 1. It also indicates how often packets are inserted with timing information, which is explained in Section II-B.

B. Source Behavior

The VBR traffic has typical ON (active) and OFF (passive) periods. During an active period, the VBR traffic is packetized. Every packet generated by the source will carry a sequence number (SN). Before we proceed with the description of the source behavior, we present some definitions.

Definition 1—Timing Insertion: Every Nth packet contains the timing information. Such packet is referred as a timing packet, as previously mentioned. Insertion of this timing information at every Nth packet is called *timing insertion*.

As shown in Fig. 1, the value of C_{s} counter is latched when the timing packet (Nth packet) arrives. The latched value is the time indication (*TI*) for the Nth packet. The number of bits used for C_{s} counter is

$$b \geq \left\lceil \log_2 \left(\frac{J_{\max}}{T_{\mathrm{r}} \cdot N} \cdot 2 \right) \right\rceil \tag{4}$$

where N is the number of reference clock cycles per C_{s} counter increment, T_{r} is calculated by (3), and $J_{\max}$ is the network's maximum jitter in seconds. The value 2 on the right-hand side of (4) is needed, because the time indication (*TI*) must be able to represent twice the network's maximum jitter in order for the receiver to estimate the received packet's jitter with counters. Equation (4) must be satisfied in order for the dejittering process at the receiver to work properly.

Definition 2—Timing Insertion Accuracy: A timing packet could arrive before the C_{s} counter changes to the next value, since C_{s} counter changes value every N reference clock cycles. The timing insertion accuracy is the number of reference clock cycles away since when the C_{s} counter has last changed its value

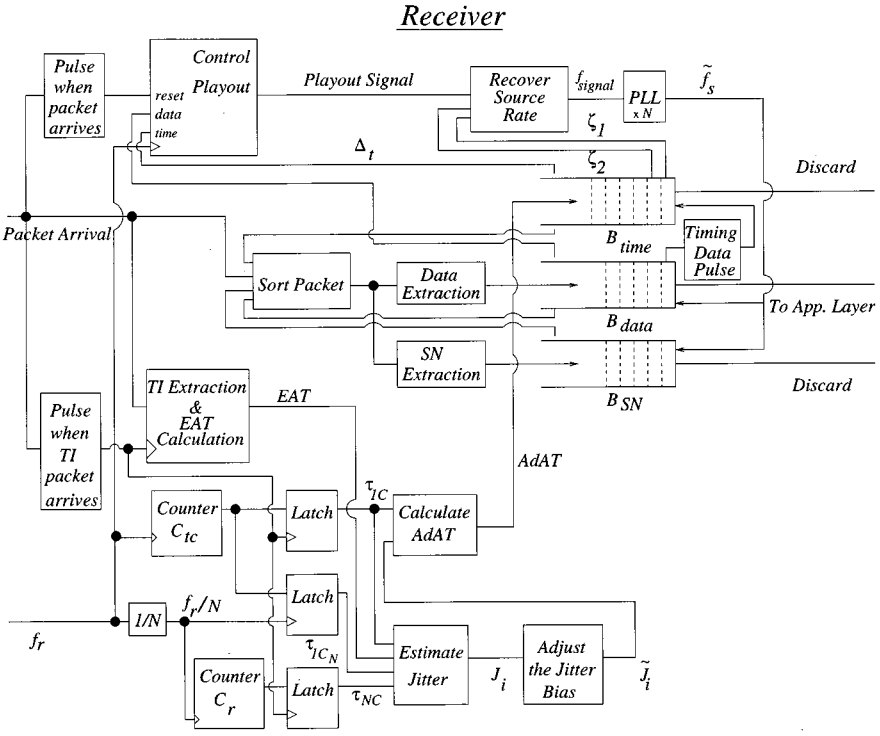

Fig. 2. Dejittering and source rate recovery.

when a timing packet arrives and obtains the time indication (*TI*).

Since the choice of N in (4) affects the timing insertion accuracy, an error similar to quantization error may occur. If the packet's arrival time is asynchronous, i.e., when packet arrival pulse is not aligned with rising edge of the reference clock, then this error $\varepsilon_{\text{insertionA}}$ is constrained by

$$\varepsilon_{\text{insertionA}} \leq N \cdot T_{\text{r}}. \tag{5}$$

Since the reference clock f_{r} is the same as the source clock, the packet's arrival time is synchronous to the reference clock. As a result, the error $\varepsilon_{\text{insertionS}}$ for the synchronous case is

$$\varepsilon_{\text{insertionS}} \leq (N-1) \cdot T_{\text{r}}. \tag{6}$$

C. Receiver Behavior

When VBR traffic in an active period is packetized and sent through an ATM network, the packets' arrival time at the receiver may deviate from their expected arrival time. The receiver estimates the timing packets' jitter by using the timing packets' *TI* and their measured arrival time, which are latched values of two counters. The *TI* is used to capture both the variable and constant part of the network delay, but only the variable part is significant. It is the varying network delay that causes jitter to packets. At the receiver, the source rate is recovered by using the timing packets' *TI*, sequence number, estimated jitter, and the waiting time for the next packet.

A detailed description of the receiver behavior will be discussed in the following sections.

1) Packet Arrival: As shown in Fig. 2, when a packet arrives at the receiver, the *sort packet* block uses the received packet's sequence number (SN) to find the correct position ω in the data buffer B_{data} for inserting the new data when the received packet is arrived out of order, otherwise the correct position ω is at the tail of B_{data}. After the correct position ω is found, the data and SN are extracted from the packet and stored at position ω in B_{data} and the sequence number buffer B_{SN}, as shown in Fig. 2. When the received packet is a timing packet, the adjusted arrival time (AdAT) will be stored at position ω in the time buffer B_{time} after the timing packet's TI is processed as described in the following section.

2) Timing Packet Arrival: A pulse will also be generated by the receiver when the received packet is a timing packet, as shown in Fig. 2. After which the counters C_{tc} and C_{r} are latched. The counter C_{tc} ticks every reference clock f_{r} cycle while the counter C_{r} ticks every N reference clock f_{r} cycles, as shown in Fig. 2. The latched values of the counters C_{tc} and C_{r} are $\tau_{1\text{C}}$ and τ_{NC}, respectively. Both τ_{NC} and $\tau_{1\text{C}}$ values capture the arrival times of a timing packet where $\tau_{1\text{C}}$ is more precise in measurements than τ_{NC} because $\tau_{1\text{C}}$ has the precision of one reference clock cycle while τ_{NC} has N cycles. The number of bits for counter C_{r} is b, i.e., the same size as time indication (*TI*), while the number of bits for counter C_{tc} is

$$b_{1\text{C}} \geq \left\lceil \log_2 \left(\left(\frac{J_{\max}}{T_{\text{r}}} \right) \cdot 2 \right) \right\rceil. \tag{7}$$

$J_{\max}$ and T_{r} are described in Section II-B.

Also the time indication (*TI*) is extracted from the timing packet in the *TI extraction and EAT calculation* block as shown in Fig. 2. The *TI* is used to calculate the expected arrival time (EAT) by

$$\text{EAT} = \left(\text{TI} + \left\lceil \frac{D_{\text{ref}}}{N} \right\rceil \right) \bmod 2^b \tag{8}$$

where D_{ref} is determined by (1), b is the number of bits reserved for time indication (*TI*), and N is described by timing insertion.

In addition, the value of counter $C_{\rm tc}$ is latched at every N reference clock $f_{\rm r}$ cycles (or whenever counter $C_{\rm r}$ increments its value) as shown in Fig. 2. The latched value is τ_{1C_N} which is used to determine $\tau_{\rm NC}$'s error in measuring the timing packet's arrival time.

Once the EAT, τ_{1C}, τ_{1C_N}, and $\tau_{\rm NC}$ are calculated by (8) and obtained by latching counters $C_{\rm tc}$ and $C_{\rm r}$, respectively, the timing packet's estimated jitter J_i in number of reference clock cycles is calculated in the *estimate jitter* block as shown in Fig. 2 by

$$J_i = J_{\rm main} + J_{\rm tail} \tag{9}$$

where i represents the jitter of timing packet i. $J_{\rm main}$ is calculated by

$$J_{\rm main} = N \cdot [({\rm EAT} - \tau_{\rm NC}) \bmod 2^b] \tag{10}$$

where $J_{\rm tail}$ is

$$J_{\rm tail} = (\tau_{1C} - \tau_{1C_N}) \bmod 2^{b_{1C}} \tag{11}$$

where τ_{1C} and τ_{1C_N} are obtained by latching counter $C_{\rm tc}$ as shown in Fig. 2 and b_{1C} is given in (7).

After the estimated jitter J_i is determined by (9), it will be adjusted due to bias such as network condition and drifting clocks, and $D_{\rm ref}$ may not be the mean network delay. The bias adjusted jitter $\tilde{J}_i$ in number of reference clock cycles is measured relative to the first mean network delay, which is illustrated in Fig. 2. Thus, it is necessary to find the deviation (or jitter bias) μ_1 at which the reference network delay ($D_{\rm ref}$) calculated by (1) is away from the first mean network delay as shown in Fig. 3. μ_1 is computed by

$$\mu_1 = \frac{\sum_{i=1}^{M_1} J_i}{M_1} \tag{12}$$

where M_1 is a constant, and J_i is the estimated jitter calculated by (9).

The network condition may change over time, and the reference clocks at the source and receiver may drift from each other over time. These are classified as drift bias δ_k from the first mean network delay at kth network condition as shown in Fig. 3. So, it is important to find the jitter bias μ_k at which the reference network delay ($D_{\rm ref}$) is away from the kth mean network delay at kth network condition

$$\mu_k = \frac{\sum_{i=M_{k-1}+1}^{M_k} J_i}{M_k - (M_{k-1}+1)} \tag{13}$$

where M_k are constants for $k = 2, \ldots, \infty$. M_k could have different values depending on how often the receiver wants to update μ_k. δ_k is then

$$\delta_k = \mu_1 - \mu_k. \tag{14}$$

The reference clocks at the receiver and source may also drift from each other, because they are not synchronous, and buffer slips may occur. This problem is addressed by μ_k. Buffer slips

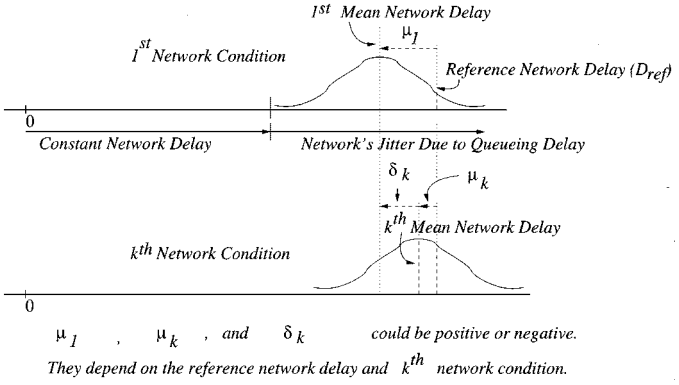

Fig. 3. Jitter bias and drift bias.

may occur, but it is not accumulative like SRTS due to low-frequency harmonics. JTS may speed or slow data to the application layer to compensate for the slip. This type of clock drift is factored into the estimated jitter J_i and removed from J_i by taking μ_k away to obtain the bias adjusted jitter $\tilde{J}_i$.

The bias adjusted jitter $\tilde{J}_i$ is calculated by

$$\tilde{J}_i = (J_i + \delta_k) - \mu_1 \tag{15}$$

where J_i is the estimated jitter computed by (9) and δ_k and μ_1 by (14) and (12), respectively. After substituting (14) into (15), the adjusted jitter $\tilde{J}_i$ is obtained as

$$\tilde{J}_i = \begin{cases} J_i - \mu_1, & \text{first network condition} \\ J_i - \mu_k, & k\text{th network condition.} \end{cases}$$

The number of estimated jitter J_i must be large when computing μ_k by (13). This is to avoid the additional error when determining the bias adjusted jitter $\tilde{J}_i$ by (15) while μ_k is approaching its mean value in (13).

After the estimated jitter J_i is obtained from (9), it is adjusted for the bias by the *adjust the jitter bias* block as shown in Fig. 2. The adjusted jitter $\tilde{J}_i$ given by (15) is then used to capture the adjusted arrival time (AdAT):

$${\rm AdAT} = (\tau_{1C} - \tilde{J}_i) \bmod 2^{b_{1C}} \tag{16}$$

where τ_{1C} is obtained by latching the counter $C_{\rm tc}$ when the timing packet arrives while b_{1C} is determined by (7). The computed AdAT, (16), is then stored at position ω of the time buffer $B_{\rm time}$ as described in Section II-C-1.

3) Control Playout: When a packet arrives at the receiver as shown in Fig. 2, a pulse is generated and sent to the *Control Playout* block. Then a waiting time counter is initialized and turned ON in the *Control Playout* block. The waiting time value $W_{\rm next}$ for the next packet arrival is measured in units of reference clock cycles. If each received packet arrives within $\beta_{\rm wait}$ where $W_{\rm next}$ is less than $\beta_{\rm wait}$, i.e., the threshold which indicates an active period, the waiting time counter resets itself and $W_{\rm next}$ is set to zero. Otherwise, a playout signal is sent to the *recover source rate* block as shown in Fig. 2.

When the data buffer $B_{\rm data}$ is empty, the waiting time counter resets and turns OFF. Also, the $W_{\rm next}$ and S_{Δ_t}, i.e., the sum of playout time difference Δ_t as obtained from (17), are set to zero. S_{Δ_t} is computed in the *Control Playout* block with parameter Δ_t, (18), when there are AdAT in the time buffer $B_{\rm time}$. S_{Δ_t} estimates the playout duration of the received packets. If S_{Δ_t}

is greater than a certain value, i.e., α_{jitter}, it indicates to the receiver that the source maybe sending a long burst, and it is fine to send packets to the application layer.

$$S_{\Delta_{\text{t}}} = \sum_{t=1}^{\lceil T/2 \rceil} \Delta_{\text{t}} \tag{17}$$

where T is the number of timing information AdAT, i.e., the adjusted arrival time, in the time buffer B_{time}, and Δ_{t} is computed from

$$\Delta_{\text{t}} = (\zeta_{\text{t}+1} - \zeta_{\text{t}}) \bmod 2^{b_{1\text{C}}} \tag{18}$$

where $\zeta_{\text{t}+1}$ and ζ_t are the AdAT in the time buffer B_{time} which correspond to $(t+1)$th and tth timing data in the data buffer B_{data}, respectively, and $b_{1\text{C}}$ is determined by (7). Throughout the remainder of the paper, the subscript t of Δ_{t} is dropped to represent the playout time difference between ζ_2 and ζ_1, i.e., AdAT corresponding to the second and first timing data in the data buffer B_{data}.

$$\Delta = (\zeta_2 - \zeta_1) \bmod 2^{b_{1\text{C}}}. \tag{19}$$

If $S_{\Delta_{\text{t}}}$, (17), is greater than α_{jitter}, i.e., the threshold which indicates that the source may be sending packets at SRA obtained from (2) during the active period, then the constant rate indicator (CRI) is set in the *Control Playout* block, and a playout signal is sent to the *recover source rate* block as shown in Fig. 2. Note that a choice of α_{jitter} and β_{wait} should have values close to $J_{\max}$, so a change in the network's condition will not affect the source rate recovery scheme.

4) To Recover the Source Rate: After the playout signal is received by the *recover source rate* block as shown in Fig. 2, the *recover source rate* block calculates the number of pulses λ which will be generated within the playout time difference Δ determined by (19).

$$\lambda = \frac{\text{SN}_2 - \text{SN}_1}{N} \tag{20}$$

where SN_1 and SN_2 are the sequence numbers corresponding to ζ_1 and ζ_2.

Since the source rate recovery scheme uses the received first and second timing data's sequence numbers and their AdAT, i.e., ζ_1 and ζ_2, to recover the source rate, packet loss does not affect the source rate recovery scheme.

For example, ζ_1 and ζ_2 point to the first and second timing data in data buffer B_{data}, and N as described by timing insertion is equal to 5. Also, a timing data is assumed missing. If the first and second timing data have sequence numbers 1 and 11, respectively, and their playout time difference Δ is ten reference clock cycles, two pulses will be generated with equal interval within ten reference clock cycles. If the timing data is not missing, λ is then equal to one. If Δ is five reference clock cycles, one pulse will be generated within five reference clock cycles. In ten reference clock cycles, two pulses are generated, which is the same as if timing data is missing. As a result, the source rate recovery scheme is robust to packet losses.

Let f_{signal} be the generated pulses, and it serves as the reference signal to the phase-locked loop (PLL) as shown in Fig. 2. The PLL's output signal $\tilde{f}_{\text{s}}$ which is the reconstructed source clock is then

$$\tilde{f}_s = N \cdot f_{\text{signal}} \text{ (Hz)} \tag{21}$$

where the PLL has a multiplying factor of N. Note that the reconstructed source clock has ON/OFF periods of pulses, because f_{signal} is generated based on (19), (20), and playout signal from the *Control Playout* block as shown in Fig. 2. As a result, it is different than the reference clock which runs on a constant frequency.

The reconstructed source clock $\tilde{f}_{\text{s}}$ is used to service the data buffer B_{data} as shown in Fig. 2. η octets are sent to the application layer with each $\tilde{f}_{\text{s}}$ clock tick since a data is η octets long. Sequence numbers in the sequence number buffer B_{SN} are also discarded by $\tilde{f}_{\text{s}}$ where each clock tick discards one sequence number.

Whenever a timing data is at the head of the data buffer B_{data} while data is being sent to the application layer by the reconstructed source clock $\tilde{f}_{\text{s}}$, the head timing information AdAT of the time buffer B_{time} is discarded after the time buffer B_{time} receives the *Timing Data Pulse* as shown in Fig. 2. A new playout time difference Δ and a new number of pulses λ within Δ are calculated by (19) and (20), respectively, for the new ζ_1 and ζ_2 pair.

We provide an example timing diagram of the receiver illustrated in Fig. 2. The timing diagram is shown in Fig. 4. For the example, N is equal to two. That is, for every two packets, a *timing indicator* (*TI*) is inserted into the packet. Also, the reference clock frequency f_{r} value is not specified, because we are trying to show the response of the receiver at each clock cycle. In addition, the example only shows how the receiver handles a burst of three incoming packets. For simplicity, both β_{wait} and α_{jitter} are equal to ten reference clock cycles. Usually, β_{wait} and α_{jitter} are given in units of seconds. They are converted into number of reference clock cycles after the reference clock frequency is known. In addition, all receiver's components are assumed ideal, i.e., no delay, so racing condition is assumed not to occur in our example.

At every reference clock f_{r} cycle, the *Counter* C_{tc} is incremented as shown in Fig. 4. The period of a f_{r}/N clock cycle is two f_{r} clock cycles since N is equal to two. At every rising edge of the f_{r}/N clock, the *Counter* C_{r} is incremented, and the *Counter* C_{tc} is latched. The latched value of *Counter* C_{tc} is $\tau_{1\text{C}_\text{N}}$. At marker A position in Fig. 4, a packet arrives, and a *Packet Arrival* signal pulse is generated. Since the packet contains a *TI* value, a *TI Packet Arrival* signal pulse is also generated. In addition, the *Counter* C_{tc} and *Counter* C_{r} are latched. Their latched values are $\tau_{1\text{C}}$ and τ_{NC}. At that time, the AdAT is assumed to have a calculated value of one as shown in Fig. 4. In addition, the *Number of Data in* B_{data}, *Number of Data in* B_{SN}, and *Number of Data in* B_{time} are incremented by one, and the *waiting time counter* is turned ON, which is indicated by the *Waiting Time Counter State* signal. As a result, the waiting time value W_{next} starts to increment by one on every reference clock f_{r} cycle, and at marker B position as illustrated in Fig. 4, the W_{next} value is incremented from zero to one.

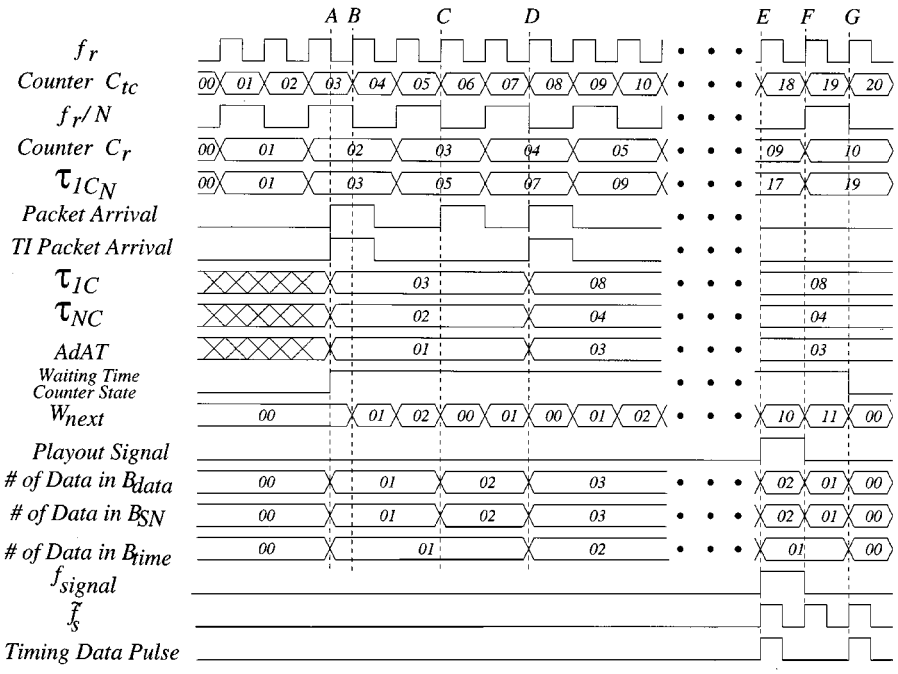

Fig. 4. Timing diagram.

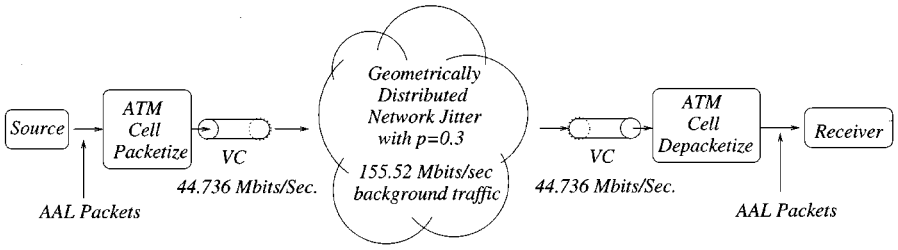

Fig. 5. Simulation model.

Another packet arrives at marker C position, but this packet does not carry any *TI* information. That is why the *Packet Arrival* signal has a pulse while the *TI Packet Arrival* signal does not have a pulse at marker C position. The *Number of Data in* $B_{\rm data}$ and *Number of Data in* $B_{\rm SN}$ are incremented by one. The *waiting time counter* is reset, and the waiting time value $W_{\rm next}$ is reset to zero. At marker D position, the third packet arrives with a *TI* value. Both *Packet Arrival* and *TI Packet Arrival* signals pulse. Afterward, the *Counter* $C_{\rm tc}$ and *Counter* $C_{\rm r}$ are latched, and their values are $\tau_{\rm 1C}$ and $\tau_{\rm NC}$, respectively. The assumed calculated value of AdAT is 3, as shown in Fig. 4.

After the third packet, there is no more packet coming into the receiver, and the waiting time value $W_{\rm next}$ increments by one on every reference clock $f_{\rm r}$ cycle. When $W_{\rm next}$ reaches 10, that is, the value of $\beta_{\rm wait}$, at marker E position, the $f_{\rm signal}$ signal pulses. From (19) and (20), Δ is equal to 2, and λ is equal to 1. That is why there is only one pulse within two reference clock $f_{\rm r}$ cycles. We ignore the performance of a PLL, and the reconstructed source clock $\tilde{f}_{\rm s}$ is given by (21). As a result, $\tilde{f}_{\rm s}$ is twice as fast as $f_{\rm signal}$. The *Number of Data in* $B_{\rm data}$ and *Number of Data in* $B_{\rm SN}$ each decreases by numeric value of 1 on every $\tilde{f}_{\rm s}$ clock cycle at marker E, F, and G positions. If the data being sent to the application layer is a timing data, e.g., first and third data in the $B_{\rm data}$ buffer, which is shown in Fig. 4 at marker E and G positions, the *Timing Data Pulse* signal pulses to remove AdAT from $B_{\rm time}$ buffer at markers E and G positions. When the *Number of Data in* $B_{\rm data}$ is empty, the *Waiting Time Counter State* signal is reset, and the *waiting time counter* is turned OFF at marker G position.

III. Performance Evaluation

We performed simulations to test JTS. The simulation model is illustrated in Fig. 5. The source traffic is packetized into AAL packets which are encapsulated into ATM cells. The AAL packets are AAL1 type for CBR traffic and AAL2 type for VBR traffic. Each timing AAL packet contains one byte of time indication (*TI*), so b is 8, (4). Note that (4) only gives the lower bound for b. All AAL packets also contain 7 bits of SN. The application traffic from the source is modeled as ON and OFF. The traffic is sent at $\rm SRA$ which is calculated by (2) when ON, and nothing is sent when otherwise. Also, the packets' arrival time at the ATM adaptation layer is assumed to be synchronous to the reference clock, and the timing packet arrives when the $C_{\rm s}$ counter changes value. So, the timing insertion error $\varepsilon_{\rm insertionS}$ is zero.

Also, one AAL packet generation is equivalent to one ATM cell generation from the source's point of view. The ATM cells

are sent through a 44.736-Mb/s ATM virtual circuit (VC) as shown in Fig. 5. A network jitter distribution of ATM cells through an ATM network is used to test the JTS since only the packets' jitter have an effect on JTS. The network jitter distribution can be characterized as geometrically distributed when the number of nodes between the source and receiver is large and the background traffic is heavy [17]. Background traffic is the traffic which competes for the same resources as the application stream when the application stream is sent from a source to a receiver. The background traffic is 155.52 Mb/s. The network jitter is modeled as geometrically distributed with probability of success equals to 0.3 as shown in Fig. 5.

After the ATM cells obtain their jitters, they are depacketized into AAL packets at the receiver. We set the D_{ref} value that is calculated by (1), i.e., the reference network delay, to zero to allow for the maximum time which JTS takes to determine the mean network jitter at the receiver. We also set α_{jitter}, β_{wait}, and $J_{\max}$ to 100 ms. For simplicity, we did not use a PLL in simulations. The reconstructed source clock is $N \cdot f_{\mathrm{signal}}$ without going through the PLL. We avoided the PLL here, so only the performance of JTS is analyzed without taken into account of the loop gain and filtering effects.

To evaluate the performance of JTS, we compare the performance of SRTS and JTS with CBR traffic. We also analyze the performance of JTS with VBR voice and video traffic. For all simulations, only the first network condition is used where M_1 is set equal to the number of AAL packets generated for the source traffic. Also, the source and receiver are implemented in software. All simulations are performed on 30-s traffic. Note that the JTS is mainly developed for AAL2 to handle real-time VBR traffic, but it is not limited to AAL2 since JTS recovers source rate by maintaining temporal relationships between consecutive packets of a time-sensitive data stream transported across an asynchronous network.

A. JTS Versus SRTS

A CBR simulation with nominal source DS3 frequency (44 736 000 Hz) and network frequency (155 520 000 Hz) is performed. The DS3 traffic is transmitted by AAL packets. Each AAL packet contains 47 octet of source information. The AAL type used is AAL1. The jitter performance is only measured in ideal situation; the receiver's buffer is able to tolerate the maximum jitter, and the clock drifts do not exist. Under ideal situation, the only noise contribution to JTS's reconstructed source clock is the network's jitter while SRTS consists of the destuffing jitter. The JTS recovers timing information by measuring and removing each AAL packet's jitter. It uses a geometrically distributed jitter distribution with probability of success equal to 0.3. The jitters are in milliseconds. The jitter distribution is calculated based on [17] where the source stream is competing with the background traffic. The source stream is running at 44 736 000 b/s while the background traffic is at 155 520 000 b/s. The performance of SRTS under ideal situations can be found in [14], [13], [12].

A jitter spectrum comparison between SRTS and the new timing scheme (JTS), Fig. 6, shows that JTS is about 1 dB higher than SRTS for all frequencies when N is equal to 8. N is the number of AAL packets when there is timing information. Also,

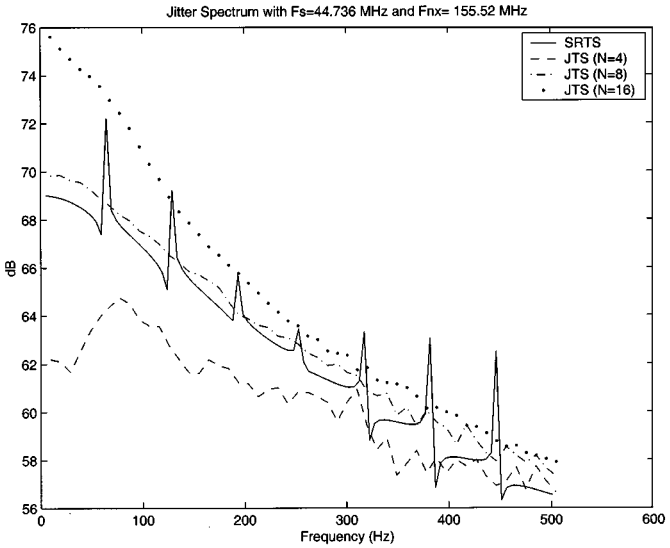

Fig. 6. Jitter spectrum of JTS and SRTS.

JTS lacks SRTS's harmonic spikes with first harmonic at 64 Hz. As N gets larger, JTS's jitter spectrum becomes much worse than SRTS. On the other hand, if N gets smaller, JTS's jitter spectrum becomes much better than SRTS. All the JTS's processes are essential to its jitter characteristics, especially JTS's ability to allow different values of N, i.e., the frequency at which the time indication (*TI*) is inserted into a packet, for different types of traffic.

The jitter spectrums of JTS and SRTS do not have a high-decibel rolloff, because a simple low-pass filter with a single pole at 0.95 is used instead of a PLL in [14]. A first-order approximation of the simple low-pass filter's cutoff frequency is 118 Hz. An off-the-shelf PLL has a cutoff frequency of around 100 Hz. In order to minimize the effects of SRTS's harmonic spikes, a specialized PLL with a low cutoff frequency of around 20 Hz is needed. This would result in higher cost to build the receiver. Since SRTS requires 4 bits of timing information for every eight AAL packets, the best fit in performance and cell formatting to JTS is when N equals to 8. JTS will use 8 bits of timing information for every eight AAL packets.

B. JTS With VBR Voice Traffic

A 8-kB/s voice signal is packetized into AAL2 packets where each AAL2 packet contains 20 octet of voice information. The SRA is 400 AAL2 packet/s. The voice signal is modeled as ON and OFF. The ON and OFF durations are exponentially distributed with average ON period of 400 ms and average OFF period of 600 ms. The voice signal is transported through a 44.736-Mb/s VC. The jitter is geometrically distributed with probability of success equal to 0.3 since the background network traffic is assumed to be 155.52 Mb/s.

From Fig. 7, the playout time of the reconstructed voice bursts at the receiver is trailing the playout time of the actual voice bursts, because talkspurts at the receiver are allowed to be delayed. By delaying the playout time of the talkspurts, data associated with the talkspurts could arrive in time for playout to minimize jitter within voice bursts. Users will not notice the small initial playout delay of bursts, but they are aware of the

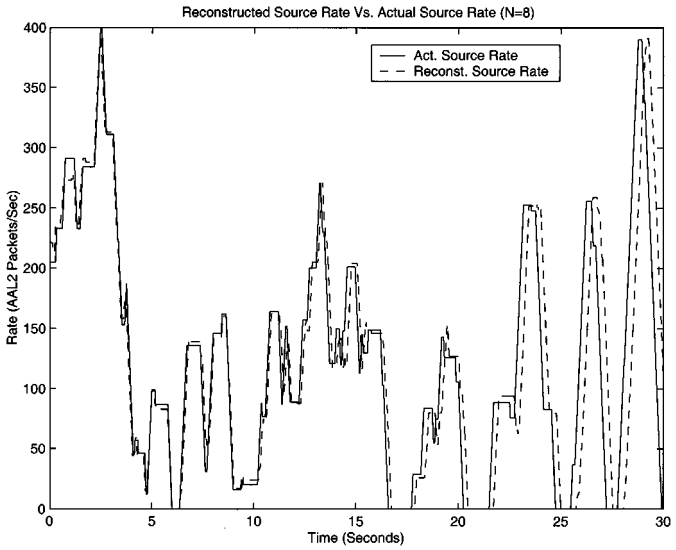

Fig. 7. Source rate versus reconstructed source rate ($N = 8$).

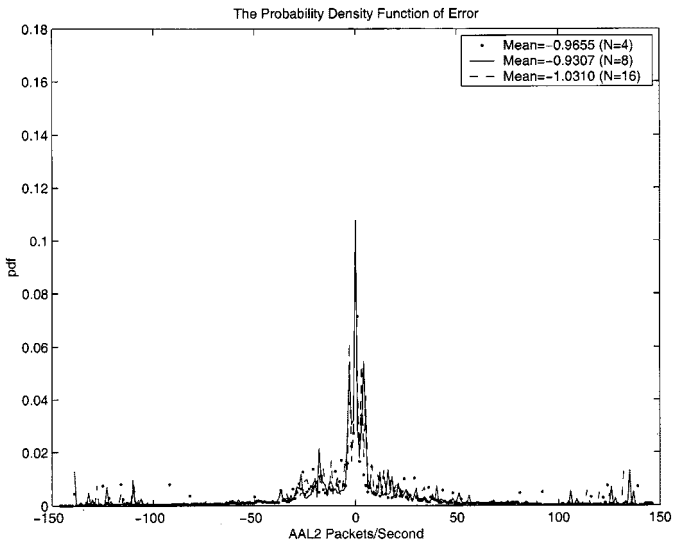

Fig. 8. Probability density function ($N = 4, 8, 12$).

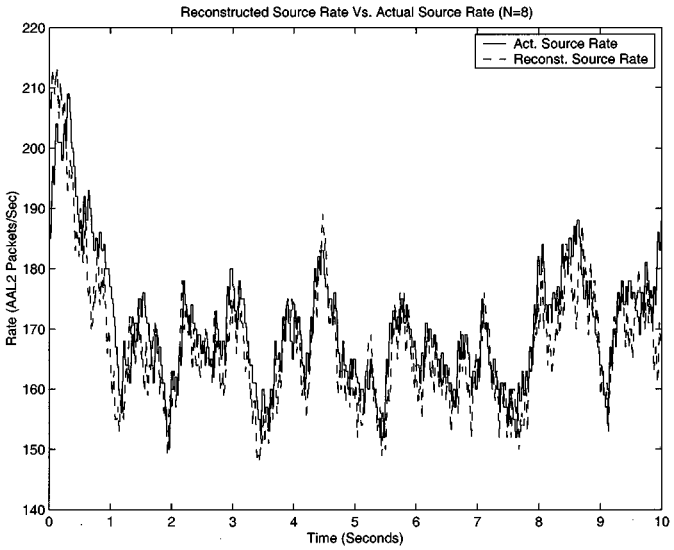

Fig. 9. Source rate versus reconstructed source rate ($N = 8$).

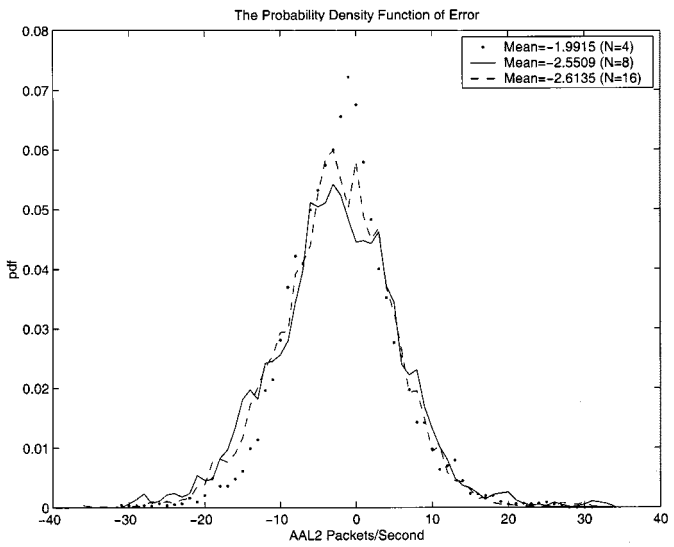

Fig. 10. Probability density function ($N = 4$, 8, 12).

jitter within the bursts. The probability density function of reconstructed source rate's error, reconstructed source rate minus actual source rate, is plotted in Fig. 8. Since α_{jitter} is set to 100 ms, error above 40 AAL2 packet/s are due to playout delay of each talkspurt. The average error from the actual source rate is -0.9307 AAL2 packet/s for N equals to 8. The average value is negative, because the reconstructed source rate is trailing the actual source rate on average by 0.9307 AAL2 packet/s.

C. JTS With VBR Video Traffic

A generalized ON/OFF video traffic which bursts every 63 ms has a random active duration uniformly distributed from 10 to 40 ms. When ON, the video traffic is sent at 8 kB/s. A 8-kB/s video rate is used, so a comparison between VBR voice and VBR video could be done. The purpose is to see the effect of the VBR video type traffic on the new source rate recovery scheme. The video traffic is packetized into 20-octet AAL2 packets. It is sent at 400 AAL2 packet/s when ON. Like VBR voice, the video traffic is transported through a 44.736-Mb/s VC with 155.52-Mb/s background network traffic. The jitter distribution is also geometrically distributed with probability of success equal to 0.3.

A plot of the reconstructed source rate versus the actual source rate in Fig. 9 shows that the reconstructed source rate is following the actual source rate. Only 10 s of the 30-s traffic is plotted in order to show more detail information of the simulation results. The probability density function (pdf) of the error from the actual source rate is in Fig. 10. The average error is -2.5509 AAL2 packet/s for N equals to 8. As N increases, the average error also increases. When N decreases, the average error also decreases. The average error for VBR video traffic is higher than VBR voice traffic, because VBR video traffic bursts periodically and expects to be played out periodically. On the other hand, the playout time of each voice talkspurt could be relaxed without distorting the perceptual quality of the voice signal. Comparing the pdf of VBR voice, Fig. 8, with VBR video, Fig. 10, VBR video traffic does not have error greater than 40 AAL2 packet/s. This is consistent

with the fact that VBR video is a dependent VBR traffic where each burst must be played out periodically and could not be delayed. Hence, error greater than 40 AAL2 packet/s does not exist for VBR video traffic.

IV. Conclusion

The JTS is able to reconstruct CBR and VBR traffic. It has a higher noise level than SRTS when N is equal to 8, but it does not have SRTS's harmonic spikes. When N decreases, the noise performance is better than the SRTS. The JTS is able to reconstruct VBR voice and video traffic with the reconstructed source rate trailing the actual source rate. On average, the reconstructed source rate trails the actual source rate by 0.9307 and 2.5509 AAL2 packet/s for VBR voice and VBR video traffic, respectively, when N is set to 8.

Note that the maximum jitter which a packet experienced during simulations is below α_{jitter} and β_{wait}, so an unstable situation never occurred. When a significant amount of packets experience jitter which are greater than α_{jitter} and β_{wait}, the JTS will cause an unstable situation in source synchronization such as buffer underflow and significant packet drop. Hence, α_{jitter} and β_{wait} must be chosen carefully. The size of AAL packets and the choice of N also affect the reconstructed source rate's performance. They should be selected to satisfy the traffic's quality of service (QoS) requirements. In summary, QoS requirements can be adjusted by modifying β_{wait}, α_{jitter}, the size of the AAL2 packets, and N. For our simulation, we set β_{wait} and α_{jitter} equal to $J_{\max}$, which is 100 ms.

References

[1] I. Akyildiz and W. Yen, "Multimedia group synchronization protocols for integrated services networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 162–173, Jan. 1996.

[2] T. V. Johnson and A. Zhang, "Dynamic playout scheduling algorithms for continuous multimedia streams," *ACM/Springer Multimedia Syst. J.*, vol. 7, pp. 312–325, July 1999.

[3] P. N. Zarros, M. J. Lee, and T. N. Saadawi, "A synchronization algorithm for distributed multimedia environments," *ACM/Springer Multimedia Syst. J.*, vol. 4, pp. 1–11, Feb. 1996.

[4] P. Tien and M. Yuang, "Intelligent voice smoother for silence-suppressed voice over Internet," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 29–41, Jan. 1999.

[5] M. C. Yuang, B. C. Lo, Y. G. Chen, and P. L. Tien, "A synchronization paradigm with QoS guarantees for multimedia communications," in *Proc. IEEE Globecom*, 1999, pp. 214–220.

[6] A. Osman, A. M. Darwish, and S. I. Shaheen, "Adaptive synchronization for real-time multimedia applications," *Proc. SPIE*, vol. 3528, pp. 202–213, Nov. 1998.

[7] C. Tryfonas and A. Varma, "A restamping approach to clock recovery in MPEG-2 systems layer," Univ. of California, Santa Cruz, CA, May 1998.

[8] H. Liu and M. El Zarki, "Delay and synchronization control middleware to support real-time multimedia services over wireless PCS networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1660–1672, Sept. 1999.

[9] R. Chang, M. Chen, J. Ho, and M. Ko, "An effective and efficient traffic smoothing scheme for delivery of online VBR media streams," in *IEEE Infocom*, 1999, pp. 447–454.

[10] ATM Forum Technical Committee, "Principles for AAL2 SSCS Applications," ATM Forum, Contributions 97-0375, 1997.

[11] K. Sriram, T. Lyons, and Y. T. Wang, "Anomalies due to delay and loss in AAL2 packet voice systems: Performance models and methods of mitigation," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 4–17, Jan. 1999.

[12] J. Walker and A. Cantoni, "Determining parameters to minimize jitter generation in the SRTS method," *IEEE Trans. Commun.*, vol. 46, pp. 82–90, Jan. 1998.

[13] ——, "Jitter analysis for two methods of synchronization for external timing injection," *IEEE Trans. Commun.*, vol. 44, pp. 269–276, Feb. 1996.

[14] H. Uematsu and H. Ueda, "Implementation and experimental results of CLAD using SRTS method in ATM networks," in *Proc. IEEE Globecom*, 1994, pp. 1815–1821.

[15] G. Panza, S. Cucchi, and D. Meli, "Buffer control technique for transmission frequency recovery of CBR connections over ATM networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, Mar. 1999, pp. 2187–2190.

[16] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: Single node case," in *Proc. IEEE Infocom*, 1994, pp. 232–241.

[17] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: Multiple nodes," *IEEE/ACM Trans. Networking*, vol. 5, pp. 122–133, Feb. 1997.

[18] D. W. Petr, R. R. Vatte, P. Sampath, and Y. Lu, "Efficiency of AAL2 for voice transport: Simulation comparison with AAL1 and AAL5," in *Proc. IEEE ICC*, vol. 2, June 1999, pp. 896–901.

[19] B. Subbiah and S. Dixit, "ATM adaptation layer 2 (AAL2) for low bit rate speech and data: Issues and challenges," in *Proc. IEEE ATM Workshop*, May 1998, pp. 225–233.

[20] T. Okutani, H. Watanabe, and T. Nisase, "Performance evaluation of multiplexing AAL2 voice traffic and TCP/IP data at the ATM cell level," in *Proc. IEEE ATM Workshop*, May 1999, pp. 391–396.

[21] H. Saito, "Performance evaluation and dimensioning for AAL2 CLAD," in *Proc. IEEE Infocom*, vol. 1, Mar. 1999, pp. 153–160.

Weilian Su (S'00) received the B.S. degree in electrical and computer engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1997, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2001. He is currently working toward the Ph.D. degree in the School of Electrical and Computer Engineering, Georgia Institute of Technology.

His research interests include timing recovery and sensor networks.

Ian F. Akyildiz (M'86–SM'89–F'96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively.

He is a Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of the Broadband and Wireless Networking Laboratory. His current research interests include wireless networks, satellite networks, and the Internet.

Dr. Akyildiz is the Editor-in-Chief of *Computer Networks* (Elsevier) and an Editor for the ACM–Springer *Journal for Multimedia Systems*, the ACM–Kluwer *Journal of Wireless Networks*, and the *Journal of Cluster Computing*. He is a past Editor for IEEE TRANSACTIONS ON COMPUTERS (1992–1996) and for IEEE/ACM TRANSACTIONS ON NETWORKING (1996–2001). He was the Program Chair of the Ninth IEEE Computer Communications Workshop, the ACM/IEEE MOBICOM'96, and the IEEE INFOCOM'98 conference. He will be the General Chair for the ACM/IEEE MOBICOM 2002 conference and the Technical Program Chair of the IEEE ICC 2003. He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria in Chile. He served as a National Lecturer for the ACM from 1989 to 1998, and received the ACM Outstanding Distinguished Lecturer Award for 1994. He also received the 1997 IEEE Leonard G. Abraham Prize award for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architecture," published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in January 1996. He is a Fellow of the ACM.