A Multimedia Synchronization Protocol for ATM Networks*

Chang-Jia Wang, Liang-Seng Koh, Chao-Hui Wu
Ming T. Liu

Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277, USA

Abstract

The media streams of a multimedia communication application may traverse through different communication channels. Because of the network delay and jitter, data in these streams sent at the same time may not arrive at a destination simultaneously. In this paper, a multimedia synchronization protocol that resynchronizes the media streams is proposed. The protocol inserts extra control cells into the media streams, and aligns the cells at the receiver site. Three synchronization policies, namely, the Drop-Old, the Transmit-Old and the Delayed-Transmit, are introduced to align the control cells. To satisfy the Quality of Service (QOS) specified by an application, the protocol must negotiate an appropriate QOS with the underlying ATM network. For each of the three synchronization policies, the relationship between the specified QOS and the negotiated QOS is derived.

1 Introduction

Multimedia communication applications require communication of multimedia objects across the underlying network within very short delays [1]. Each multimedia object is comprised of mixed media types including text, voice, graphics, image, audio and video. Because of different characteristics of the media, each media may be carried by a different channel. In the case that there is no network delay or jitter, the *media streams* that are generated at the same time have no problem to reach a destination simultaneously. However, since these media streams may traverse through different communication channels, the data may incur different delays or jitters before reaching the destination. If these streams are not resynchronized before delivery to the destination, the quality of multimedia objects can be seriously degraded. A simple example is that there can be no coordination between a speaker's lip and voice. Therefore, *multimedia synchronization* is considered one of the most important requirements for a multimedia communication application.

*Research reported herein was supported by U.S. Army Research Office, under contracts No. DAAL03-91-G-0093 and No. DAAL03-92-G-0184. The views, opinions, and/or findings contained in this paper are those of the authors and should not be construed as an official Department of the Army position, policy or decision.

Some works on multimedia synchronization have been published lately [2, 3, 4, 5, 6, 7]. Their main emphasis has been on the architecture to support multimedia applications in the Open Systems Interconnection (OSI) environments. However, the basic problem of governing the ways data are buffered and resynchronized is only briefly mentioned or totally ignored.

In this paper, a simple protocol is proposed for synchronizing media streams; it is assumed that the underlying network operates in the Asynchronous Transfer Mode (ATM) [8]. The proposed protocol inserts extra control cells into media streams and tries to align those cells at the receiver site. Three different policies that can be used to align the control cells are introduced, namely, the Drop-Old, the Transmit-Old and the Delayed-Transmit. Based on the policies, the *Quality of Service* (*QOS*) is derived to ensure that the network delay, cell loss and out-of-sync are within the tolerable limits specified by applications.

The rest of this paper is organized as follows. In the next section, the synchronization protocol and the three synchronization policies are described in detail. In Section 3, the translations of the QOS for the three policies are discussed. At the end Section 3, an example is given for further elaborating the policies. Finally, in Section 4, a concluding remark is made.

2 The Synchronization Protocol

2.1 Network Model

The proposed protocol serves as an interface between the underlying ATM network and the multimedia applications. Figure 1 shows the structure where the synchronization protocol is placed. A set of media streams of a multime-

1063-6927/94 $03.00 © 1994 IEEE

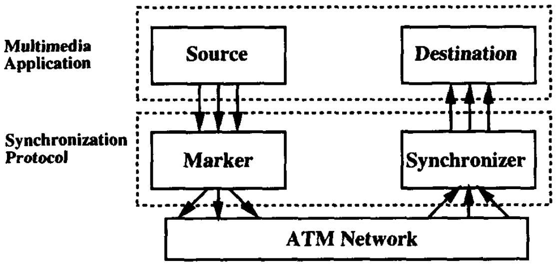

Figure 1: The structure of the synchronization protocol

dia application is sent from the source to the destination. The streams may traverse through different routes in the network; therefore, they may not be able to arrive at the destination simultaneously. Hence, the synchronization protocol is in charge of ensuring that the data in these media streams sent from the source at the same time will also arrive at the destination at about the same time within an acceptable tolerance.

The synchronization protocol consists of two entities. The *marker* inserts some control cells, called *synchronizing cells* (or simply *sync cells*), into the media streams to mark the places where data should be synchronized. The *synchronizer*, on the other hand, recognizes the sync cells and transmits the streams so that those places in the streams marked by the sync cells will be received by the destination almost simultaneously within out-of-sync tolerance. The proposed synchronization protocol is built on top of an ATM network. In the CCITT standard, there are two bits in an ATM cell that indicate the type of payload of a cell. Therefore, it is not difficult to distinguish sync cells from any other data cells.

An important feature of the ATM network is that it must be able to negotiate the QOS with the network users, and guarantee the QOS throughout the session once it has been initiated. Because of the overhead introduced by the synchronization protocol, service requirements for the underlying network must be tougher than the QOS specified by a multimedia application. In the network model shown in Figure 1, there are two levels of QOS. The application specifies the required QOS to the synchronization protocol. Then, the protocol calculates its overhead and negotiates its own version of QOS with the ATM network. The QOS agreements between the protocol and the network must be adequate to support the QOS agreements between the application and the protocol.

Based on its requirements, a multimedia application may negotiate the following parameters of the QOS with the synchronization protocol. These QOS include:

Cell loss rate: How many cells can be lost in the transmission?

Delay tolerance: How many cells in the transmission must reach the destination within the allowed end-to-end delay?

Out-of-sync tolerance: How long can the time difference be between the first and the last should-be-synchronized cells in different media streams to arrive at the destination?

Data transmission rate: How fast should the source of the multimedia application send its data?

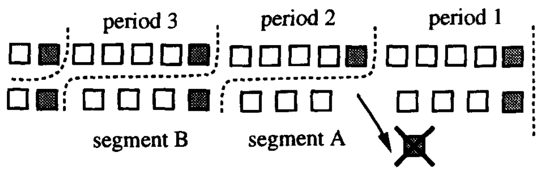

Figure 2: A loss sync cell causes misalignment

With the QOS specified by the application, the synchronization protocol calculates its overhead and negotiates another set of QOS with the ATM network. These QOS include:

Cell loss rate: How many cells can be lost in the network?

Delay characteristic: How many cells in the transmission must reach the synchronizer within the allowed end-to-end delay ?

Network Bandwidth: How fast should the network transmit cells?

A good synchronization protocol must incur the least amount of overhead; i.e., it must make the QOS between the application and the protocol, and the QOS between the protocol and the network, as close as possible.

2.2 Protocol Description

As mentioned, a sync cell is inserted into every media stream either periodically, or by request of an application. A sync cell and its consecutive data cells form a *segment*. A *period* describes a set of segments in different streams that need to be synchronized. A sync cell can be sent with a higher priority, but this does not prevent it from being lost in the network. When a sync cell in a segment is lost, the rest of the data cells in the segment will be misinterpreted as belonging to the previous segment, and the next segment will be aligned to the segments in the current period. For example, in Figure 2, because of the loss of a sync cell, the synchronizer will misinterpret the data cells in segment A as part of the data in period 1, and will treat segment B as the data of period 2. To prevent such a problem, a sequence number that distinguishes one period from another is added to each sync cell. For example, if the sync cell in period 3 were marked by sequence number 3, the synchronizer would have been able to notice that segment B does not belong to period 2. In general,

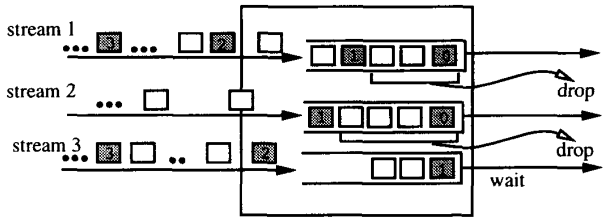

Figure 3: The Drop-Old policy

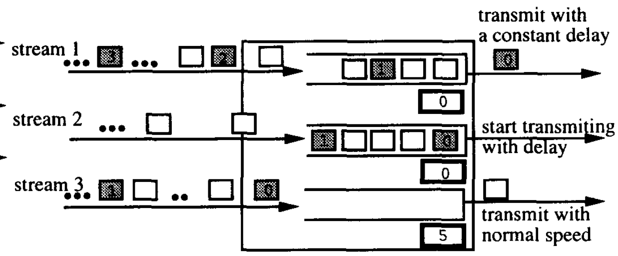

Figure 4: The Delayed-Transmit policy

if there are n distinct sequence numbers, the synchronization protocol will be able to tolerate $n-1$ consecutive loss of sync cells. In this paper, a segment with a sync cell that has sequence number i is called segment i, and the period contains the segments with sequence number i is called period i.

Certainly, the performance of the synchronization protocol is affected by some system parameters associated with buffers and sync cells. The following are some issues regarding system parameters and these issues will be addressed in Section 3.

- ◇ How often should the sync cells be inserted?
- ◇ How many distinct sequence numbers are needed?
- ◇ How large should the buffers be that the synchronizer reserves?

2.3 Synchronization Policies

A *synchronization policy* is a procedure describing how the synchronizer aligns the sync cells. There are several synchronization policies that can be used to resynchronize the media streams. In this paper, three policies are proposed: the *Drop-Old* , the *Transmit-Old* and the *Delayed-Transmit.* Both the Drop-Old and the Transmit-Old are *blocking policies.* These two policies block early arrival segments and transmit the segments in the same period simultaneously when all the segments in the period have been received. However, the Drop-Old policy discards the segments in a period when one of the sync cells in the period is lost, while the Transmit-Old policy always transfers the segments that would have been discarded in the Drop-Old policy. The Delayed-Transmit policy, on the other hand, does not block the early arrival segments. Instead, it inserts extra delays before transmitting the data cells of the early arrival segments to the destination.

2.3.1 The Drop-Old Policy

As mentioned, there is a chance that the sync cells may be lost in the network. However, since every sync cell is attached with a sequence number, it is not difficult to recognize the loss of a sync cell when the next sync cell arrives at the synchronizer with a sequence number different from those waiting in buffers. The Drop-Old policy is that whenever a different sequence number arrives at the synchronizer indicating loss of sync cells, the old segments waiting in the buffer will be discarded. For example, in Figure 3, the synchronizer realizes that stream 3 loses sync cell 0 when sync cell 1 arrives at the head of the corresponding buffer. Therefore, segments in period 0 of streams 1 and 2 will be dropped from the respective buffers.

2.3.2 The Transmit-Old Policy

Another approach dealing with the loss of sync cells is to transmit old segments. In the case that the Transmit-Old is used in the situation described in Figure 3, segments in streams 1 and 2 waiting for stream 3 will be transmitted if segment 1 of stream 3 arrives at the head of the buffer in stream 3. The newly arrived segment 1 of stream 3 will wait at the buffer for the segments of period 1 in all the other streams to reach the head of their buffers.

2.3.3 The Delayed-Transmit Policy

The Delayed-Transmit policy is a non-blocking policy. Cells that arrive early will be delayed a period of time before they are transmitted to the destination. Therefore, the streams that are transmitted too fast will be slowed down and, eventually, the streams will be transmitted almost synchronously. The delay can be added in various ways. However, in order to reduce overhead, a simple approach is adopted.

The synchronizer maintains a register for each media stream to indicate the period of the stream that is being transmitted. For example, in Figure 4, data of period 0 in streams 1 and 2 are being transmitted, while data of period 5 (which precedes period 0) in stream 3 is still being transmitted. When registers show that there are data still in earlier periods, the synchronizer will transmit each early arrival cell with a constant delay. That is, streams 1 and 2 in Figure 4 will be transmitted at a slower rate than stream 3.

If all the streams are in the same period, the synchronizer will resume its normal transmission speed. For example, when stream 3 starts transmitting period 0, streams 1 and 2 will resume their normal transmission speeds.

3 QOS Provided

The synchronization protocol is responsible for translating the QOS specified by the users into the QOS requirements for the network. In this section, three sets of QOS parameters are used. One is the QOS specified by a multimedia application, and another is the QOS that can be provided by the ATM network. In between, a set of parameters indicating the actual QOS provided by the synchronization protocol is used to translate the application-specified QOS into the network-provided QOS. In practice, the synchronization protocol considers both the application specified QOS and the overhead introduced by the protocol to come out with a new QOS; then, it negotiates this derived QOS with the ATM network. However, for convenience, the QOS provided by the synchronization protocol is derived from the network-provided QOS, and the application-specified QOS must fall within the range of service that the protocol can support.

In addition, there are some system parameters, such as the segment length, buffer size and data retransmission rate, which determine the behavior of the synchronization protocol. These parameters must be tuned so that the protocol can satisfy the QOS specified by the application.

In this section, the relationship among QOS and system parameters are derived for the three policies described in the previous section. The notations for QOS and system parameters are defined as follows.

QOS specified by the multimedia applications: These are the parameters given by the applications. They are denoted by capital letters with a "hat" ($\hat{}$).

- ◇ Cell loss rate $\hat{P}$: Ratio of the number of lost cells to the total number of cells transmitted.
- ◇ Delay tolerance $\hat{D}$ and α: At least α portion of cells must arrive at the destination within the maximum end-to-end delay allowed ($\hat{D}$); cells arrived latter than $\hat{D}$ are subject to be dropped.
- ◇ Out-of-sync tolerance $\hat{\Phi}$ and γ: $\hat{\Phi}$ is the maximum time difference allowed between any two segments of the streams that should be synchronized. For the Delayed-Transmit policy, an application needs to specify the minimum percentage of cells (γ) not violating the out-of-sync tolerance.
- ◇ Data transmission rate $\hat{\Omega}$: Number of data cells transmitted in a unit of time.

QOS provided by the synchronization protocol: These variables denote the service quality of the protocol. They are denoted by capital letters.

- ◇ Cell loss rate P.
- ◇ Delay characteristic $F(t)$: Probability density function describing the characteristic of the delay introduced by the protocol.
- ◇ Maximum out-of-sync Φ: Maximum time difference allowed between any two segments of the streams that should be synchronized.
- ◇ Data transmission rate Ω.

QOS provided by the network: These parameters are the amount of QOS needed for the network to support the QOS specified by the applications. They are denoted by lower case letters.

- ◇ Cell loss rate p.
- ◇ Network delay tolerance d and β: At least β portion of cells must pass through the network within a time limit d. (A probability density function $f(t)$ is assumed to describe the network delay characteristic, i.e., $\int_0^d f(t)dt \geq \beta$. However, the function is only used for calculation; no actual knowledge of $f(t)$ is required.)
- ◇ Cell transmission rate ω: Number of cells transmitted in a unit of time; i.e., the network bandwidth reserved by the protocol.

System parameters:

- ◇ m: Number of media streams.
- ◇ s: Number of cells per segment.
- ◇ n: Number of distinct sequence numbers for sync cells.
- ◇ b: System buffer size.
- ◇ μ: Cell transmission rate of the synchronizer; i.e., number of cells sent by the synchronizer in a unit of time. (It is required that $\mu > \omega$. Otherwise, the synchronizer will not be able to digest cells in the buffers and buffer overflow will occur eventually.)
- ◇ ν: Cell transmission rate for the early streams in the Delayed-Transmit policy. (Usually, $\nu < \omega$.)

In the rest of this paper, a subscript of a variable denotes the corresponding parameter associated with a media stream.

3.1 The Drop-Old Policy

Delay Tolerance

When the buffer of stream i is empty, a segment in the stream will be transmitted at time t if (1) it arrives the latest at the synchronizer at time t, or (2) it arrives at the synchronizer earlier than time t and some sync cells of other streams arrive the latest at time t. Assuming that there is virtually no chance that two sync cells in two streams can arrive at the same time (i.e., $f_i(t)f_j(t) \to 0$, for all $i \neq j$), the resulting delay characteristic becomes:

$$
\begin{aligned}
F_i(t) = \;& f_i(t)\prod_{j\neq i}\int_0^t f_j(x)dx \\
& + \int_0^t f_i(x)dx \sum_{j\neq i}[f_j(t)\prod_{k\neq j,k\neq i}\int_0^t f_k(x)dx] \\
= \;& \sum_j f_j(t)\prod_{k\neq j}\int_0^t f_k(x)dx
\end{aligned}
\tag{1}
$$

Note that the synchronizer's outgoing delay characteristic does not depend on the streams; i.e., $F_i(t) = F_j(t)$ for any i, j, and t. In addition, the following inequality holds since α_i is the portion of the cells needed to be sent before time $\hat{D}_i$.

$$\int_0^{\hat{D}_i} F_i(t)dt \geq \alpha_i$$

In the case that the network delay patterns for all streams are the same; i.e., $f_i(t) = f_j(t) = f(t)$ for any i, j, and t, Equation (1) can be modified as follows.

$$F_i(t) = mf(t)[\int_0^t f(x)dx]^{m-1}$$

Because $f(t) = 0$ when $t \leq 0$,

$$\int_0^{\hat{D}_i} F_i(t)dt = [\int_0^{\hat{D}_i} f(x)dx]^m \geq \alpha_i$$

Thus,

$$\int_0^{\hat{D}_i} f(x)dx \geq \alpha_i^{\frac{1}{m}} \tag{2}$$

ensures that an adequate number of cells are received in time when the buffer is originally empty. In the worst case, the time that a cell in stream i spends in the buffer is b_i/μ_i. Therefore,

$$\int_0^{\hat{D}_i-\frac{b_i}{\mu_i}} f(x)dx \geq \alpha_i^{\frac{1}{m}} \tag{3}$$

ensures that an adequate number of cells are received in time in any cases. When the delay patterns for the streams are different, the worst case can be used. In other words,

$$\min_j[\int_0^{\hat{D}_i-\frac{b_i}{\mu_i}} f_j(x)dx] \geq \alpha_i^{\frac{1}{m}} \tag{4}$$

That is, at least $\alpha_i^{\frac{1}{m}}$ portion of cells in stream i must arrive at the synchronizer before $\hat{D}_i - \frac{b_i}{\mu_i}$ ($d_i \leq \hat{D}_i - \frac{b_i}{\mu_i}$ and $\beta_i \geq \alpha_i^{\frac{1}{m}}$). When an application specifies $\hat{D}_i$ and α_i, the protocol in turn tunes the respective parameters $\hat{D}_i - \frac{b_i}{\mu_i}$ and $\alpha_i^{\frac{1}{m}}$ before negotiating them with the ATM network.

Cell Loss Rate

In the Drop-Old policy, a cell can be transmitted successfully only when its corresponding segment is not dropped and itself is not lost. One of the possibilities that a segment can get through is that all the corresponding sync cells in the other media streams arrive successfully. Thus, if the protocol cell lost rate for stream i is P_i, then

$$1 - P_i \geq (1-p_i)\prod_{j\neq i}(1-p_j)$$

To ensure that the application obtains satisfactory QOS, p_j must be small enough such that

$$\hat{P}_i \geq 1 - \prod_j(1-p_j) \geq P_i \tag{5}$$

In other words, if the synchronization protocol negotiates a set of p_j small enough such that the above inequality holds, the protocol will be able to provide the necessary QOS for the application since P_i is even smaller than $1 - \prod_j(1-p_j)$. Note that the inequality holds for any i. Therefore, p_j should be chosen so that it satisfies the strongest $\hat{P}_i$. In addition, since those cells received after $\hat{D}$ are subject to be discarded, inequality (5) should be modified as follows.

$$\hat{P}_i \geq (1 - \prod_j(1-p_j)) + (1-\alpha_i) \tag{6}$$

Out-of-Sync Tolerance and Segment Size

The worst scenario of a stream being out-of-sync is that the stream consequently loses sync cells. The synchronizer will continuously transmit the cells in the stream while holding cells from other streams in the buffer. Let Θ_i be the number of cells in stream i that is ahead of other streams. Then,

$$\Theta_i = (1 - \prod_{j\neq i} p_j)(s_ip_i + 2s_ip_i^2 + 3s_ip_i^3 + \cdots)$$

where $(1 - \prod_{j\neq i} p_j)$ is the probability that some other streams do not lose the sync cells, and $(s_ip_i + 2s_ip_i^2 + 3s_ip_i^3 + \cdots)$ is the expected number of consecutive data cells that lose their sync cells. Let $q_i = 1 - \prod_{j\neq i} p_j$. Then,

$$\Theta_i = \frac{s_iq_ip_i}{(1-p_i)^2}$$

Therefore,

$$\hat{\Phi} \geq \max_i \frac{\Theta_i}{\omega_i} = \frac{s_iq_ip_i}{\omega_i(1-p_i)^2} \tag{7}$$

Thus, to ensure satisfactory stream synchronization, the protocol must choose an appropriate network bandwidth (ω_i) and a cell lost rate (p_i). Another option to guarantee the QOS of an application is to adjust the segment size s_i. Inequality (7) can be rewritten as

$$s_i \leq \hat{\Phi}\omega_i \frac{(1-p_i)^2}{q_i p_i} \tag{8}$$

Therefore, increasing the frequency of adding sync cells (decreasing the segment size) improves the synchronization quality.

Data Transmission Rate and Segment Size

The effective data cell transmit rate for the protocol is determined by the following inequality.

$$\hat{\Omega}_i \leq \Omega_i = \omega_i \frac{s_i}{s_i+1} \tag{9}$$

In other words,

$$s_i \geq \frac{\hat{\Omega}_i}{\omega_i - \hat{\Omega}_i} \tag{10}$$

Note that out-of-sync tolerance ($\hat{\Phi}$) prefers smaller segment sizes (inequality (8)), while the data transmit rate prefers larger segments. A trade-off must be determined between these two factors.

Range of Sync Cells' Sequence Numbers

Suppose that there are n distinct sequence numbers for the sync cells. When n sync cells are lost consequently in a stream, the protocol will treat the next round of sequence numbers as if it were to be synchronized with cells blocked in the buffer. As the result, the streams will be out of phase by ns_i cells. Therefore, n should be large enough so that

$$p_i^n \leq \epsilon \tag{11}$$

where ϵ is a small number approximating zero.

Buffer Size

Since out-of-sync segments will be discarded in the Drop-Old policy, the buffer only needs to be large enough to contain cells within out-of-sync parameters. Therefore,

$$b_i \geq \hat{\Phi}\omega_i \tag{12}$$

3.2 The Transmit-Old Policy

Delay Tolerance

The worst delay for a cell happens when the cell is held at the end of the buffer, and some other streams consecutively lose sync cells. The number of cells passes through the synchronizer in the stream j that loses sync cells is

$$s_j(p_j + 2p_j^2 + 3p_j^3 + \cdots) = \frac{s_j p_j}{(1-p_j)^2}$$

Therefore, the maximum delay for stream i is the total of the time it spends on waiting for a sync cell of some other streams to arrive, and the time it spends to reach the head of the buffer. Let D_i be the maximum delay of stream i. Then,

$$D_i = \max_{j \neq i} \frac{s_j p_j}{\omega_j (1-p_j)^2} + \frac{b_i}{\mu_i} \tag{13}$$

Thus, similar to the argument for inequality (3), the following inequality can be developed.

$$\int_0^{\hat{D}_i - D_i} f(t)dt \geq \alpha_i^{\frac{1}{m}} \tag{14}$$

That is, at least $\alpha_i^{\frac{1}{m}}$ portion of cells must arrive at the synchronizer before $\hat{D}_i - D_i$ ($d \leq \hat{D}_i - D_i$ and $\beta_i \geq \alpha_i^{\frac{1}{m}}$).

Cell Loss Rate

Since the Transmit-Old policy does not intent to drop any cells, the following inequality can be established.

$$\hat{P}_i \geq p_i + (1 - \alpha_i) \tag{15}$$

Other Parameters

The way of old segments being handled does not have effect on buffer size, segment size, out-of-sync allowance, data transmission rate, or the range of sync cells' sequence number. Therefore, these parameters are the same as those of the Drop-Old policy.

3.3 The Delayed-Transmit Policy

Delay Tolerance

Since a media stream in the Delayed-Transmit policy will not wait for other late streams, its delay depends only on the network delay and the buffer delay. Hence, to ensure that the delay of the protocol satisfies a satisfactory factor α, the network delay characteristic should be as follows.

$$\int_0^{\hat{D}_i - \frac{b_i}{\nu_i}} f(x)dx \geq \alpha_i \tag{16}$$

where ν_i is the data retransmit rate for the early streams. That is, at least α_i portion of cells must arrive at the synchronizer before $\hat{D}_i - \frac{b_i}{\nu_i}$ (i.e., $d_i \leq \hat{D}_i - \frac{b_i}{\nu_i}$ and $\beta_i \geq \alpha_i$).

Out-of-Sync Tolerance and Data Retransmission Rate for Early Streams

Let T be an arbitrary delay time. It can be shown that

$$\begin{aligned} \mathrm{Prob}(\phi \leq \Phi) \geq & \ \mathrm{Prob}(\text{All sync cells arrives in } (T, T+\Phi)) \\ = & \ \textstyle\prod_i \int_T^{T+\Phi} f_i(t)dt \end{aligned}$$

where ϕ is the amount of out-of-sync among the streams. Therefore,

$$\prod_i \int_T^{T+\Phi} f_i(t)dt \geq \gamma$$

ensures $\text{Prob}(\phi \leq \Phi) \geq \gamma$.

Assuming that each stream has the same delay characteristic, the above inequality can be written as

$$\int_0^{T+\Phi} f(t)dt - \int_0^T f(t)dt \geq \gamma^{\frac{1}{m}} \tag{17}$$

If the delay characteristic is different, an $f_i(t)$ whose integration $\int_T^{T+\Phi} f_i(t)dt$ is the minimum among all i, can be selected as $f(t)$ to make inequality (17) hold.

Let γ_1 and γ_2 be the portions of cells that can be transferred within delays d_i and $d_i + \Phi$, respectively. Inequality (17) means that if there exists a delay time d_i, such that $\gamma_2 - \gamma_1$ is greater than $\gamma^{\frac{1}{m}}$, then there will be more than γ portion of cells being retransmitted within the out-of-sync tolerance Φ. In order to find out whether the network is able to support inequality (17), the protocol must ask the underlying network two questions:

1. Is transmitting γ_1 portion of cells in the network with the maximum delay of d_i possible?
2. Is transmitting γ_2 portion of cells in the network with the maximum delay of $d_i + \Phi$ possible?

If the answer to the first question is no (no more than γ_1 of cells can arrive before d_i), and to the second is yes (more than γ_2 of cells can arrive before $d_i + \Phi$), then the network will be able to support out-of-sync tolerance Φ when $\gamma^{\frac{1}{m}} \leq \gamma_2 - \gamma_1$.

Let ν_i be the data retransmission rate for the i-th stream. The value of Φ in inequality (17) should be chosen such that

$$\Phi + \frac{b_i}{\nu_i} \leq \hat{\Phi} \tag{18}$$

for all i.

Cell Loss Rate

Similar to the Transmit-Old policy, the Delayed-Transmit policy does not intent to drop any cells. Therefore,

$$\hat{P}_i \geq p_i + (1 - \alpha_i) \tag{19}$$

Buffer Size

The buffer in the Delayed-Transmit policy constantly digests cells at the rate of ν_i. As mentioned before, the buffer only needs to be large enough to hold cells within the tolerance of out-of-sync. The number of cells within such tolerance is $\omega_i \hat{\Phi}$, among which $\nu_i \hat{\Phi}$ has been retransmitted. Therefore, the buffer size should be at least as follows.

$$b_i \geq (\omega_i - \nu_i)\hat{\Phi} \tag{20}$$

3.4 An Example

In this section, an example is given to illustrate and compare the three policies. For ease of referencing, a summary of the above discussion is given below.

Delay Tolerance

- ◇ Drop-Old: $d_i = \hat{D}_i - \frac{b_i}{\mu_i}$ and $\beta_i = \alpha_i^{\frac{1}{m}}$.
- ◇ Transmit-Old: $d_i = \hat{D}_i - D_i$ and $\beta_i = \alpha_i^{\frac{1}{m}}$, where $D_i = \max_{j \neq i} \frac{s_j p_j}{\mu_j (1-p_j)^2} + \frac{b_i}{\mu_i}$.
- ◇ Delayed-Transmit: $d_i = \hat{D}_i - \frac{b_i}{\nu_i}$ and $\beta_i = \alpha_i$.

Cell Loss Rate

- ◇ Drop-Old: $\hat{P}_i \geq 1 - \prod_j (1 - p_j) + (1 - \alpha_i)$
- ◇ Transmit-Old and Delayed-Transmit: $\hat{P}_i \geq p_i + (1 - \alpha_i)$

Buffer Size

- ◇ Drop-Old and Transmit-Old: $b_i \geq \omega_i \hat{\Phi}$
- ◇ Delayed-Transmit: $b_i \geq (\omega_i - \nu_i)\hat{\Phi}$

Out-of-Sync Tolerance

- ◇ Drop-Old and Transmit-Old: $\hat{\Phi} \geq \max_i \frac{s_i q_i p_i}{\omega_i (1-p_i)^2}$, where $q_i = 1 - \prod_{j \neq i} p_j$
- ◇ Delayed-Transmit: $\int_{d_i}^{d_i+\Phi} f(t)dt \geq \gamma^{\frac{1}{m}}$, where $\Phi + \frac{b_i}{\nu_i} \leq \hat{\Phi}$.

Consider a real-time application needs to synchronize an audio and a video streams with the QOS as follows [9]:

- ◇ Video Stream: $\hat{\Omega} = 25Mbps$, $\hat{P} = 10^{-3}$ and $\alpha = 99.9\%$.
- ◇ Audio Stream: $\hat{\Omega} = 64Kbps$, $\hat{P} = 10^{-1}$ and $\alpha = 90.0\%$.

In addition, the application requires that $\hat{D}$ and $\hat{\Phi}$ to be 250ms and 133ms, respectively.

Table 1 shows the derived network QOS and the system parameters of the three policies according to the application QOS. The delayed transmission rate,ν_i, is assumed to be chosen as $\frac{3}{4}\omega_i$ by the synchronization protocol. From Table 1, it can be observed that the Drop-Old policy requires the network to support a significantly less cell loss rate than the other two policies do. Thus, the Drop-Old policy is more demanding on network cell loss rate. However, under the Drop-Old policy, the delay requirement for the network is less restricted. The Delayed-Transmit policy imposes stronger demand on network delay (i.e. shorter delay period); nevertheless, it allows a smaller

Table 1: Network QOS and System Parameters to Support the QOS of the Example

		Drop-Old		Transmit-Old		Delayed-Transmit	
		video	audio	video	audio	video	audio
Delay Tolerance	d_i(ms)	117.02	116.98	100.26	116.55	72.70	72.64
	β_i(%)	≥ 99.95	≥ 94.87	≥ 99.95	≥ 94.87	≥ 99.90	≥ 90.00
Transmission Rate	ω_i(bps)	27.6M	70.67K	27.6M	70.67K	27.6M	70.67K
Cell lose	p_i	$\leq 10^{-3}$	$\leq 10^{-3}$	$\leq 10^{-3}$	$\leq 10^{-1}$	$\leq 10^{-3}$	$\leq 10^{-1}$
Buffer Size	b_i(b)	$\geq$ 3.67M	$\geq$ 9.4K	$\geq$ 3.67M	$\geq$ 9.4K	$\geq$ 0.92M	$\geq$ 2.35K
Segment Size	s_i(b)	12M	9.6K	12M	9.6K	12M	9.6K

portion of cells to be received correctly in that delay period. In addition, this policy requires a smaller buffer size since cells in a stream are sent, rather than buffered, with a slower speed if the stream is ahead of some other streams.

4 Conclusion

In this paper, a multimedia synchronization protocol is proposed. The protocol inserts synchronizing cells into the media streams that need to be synchronized. Then, the sync cells are aligned at the receiver site. Consequently, the actual data cells that follow the sync cells are synchronized.

Three different policies are introduced to align the sync cells. The Drop-Old and the Transmit-Old policies block early arrival sync cells in buffers until all the sync cells sent at the same time arrived. If a sync cell is lost in the network, not all the sync cells that are sent at the same time can be received. Then, the Drop-Old policy discards all the blocked sync cells and their consecutive data cells, while the Transmit-Old policy transmits them before sending the data of the next period. The Delayed-Transmit policy does not block the early arrival cells. Instead, it transmits each data cell in early arrival segments with a constant delay.

The synchronization protocol must guarantee the QOS specified by multimedia applications. Since the protocol introduces some overhead, the specified QOS is recalculated into a stronger one that the protocol can negotiate with the ATM network. A QOS translation is derived for the three synchronization policies. The Drop-Old policy introduces less delay, while the Transmit-Old policy loses fewer cells. Both the blocking policies are good at resynchronizing the media streams, but they also introduce additional jitter to the streams. As the result, the quality of playing back at the destination will be degraded. On the other hand, the Delayed-Transmit policy transmits the media streams more smoothly at the expense of worse synchronization quality.

References

[1] H. T. Mouftah, "Multimedia communications: An overview," in *IEEE Communications Magazine*, pp. 18–35, May 1992.

[2] W. A. Montomery, "Techniques for packet voice synchronization," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 6, pp. 1022–1028, December 1983.

[3] S. Ades, R. Want, and R. Calnan, "Protocols for real time voice communication on a packet local network," in *Proc. IEEE INFOCOM '87*, pp. 525–530, 1987.

[4] G. Karlsson and M. Vetterli, "Packet video and its integration into network architecture," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, pp. 739–751, June 1989.

[5] C. Nicolaou, "An architecture for real-time multimedia communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 391–400, April 1990.

[6] P. V. Rangan and D. C. Swinehart, "Software architecture for integration of video services in the Etherphone," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 9, pp. 1395–1404, December 1991.

[7] T. D. Little and A. Ghafoor, "Multimedia synchronization protocols for broadband integrated services," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 9, pp. 1368–1382, December 1991.

[8] M. Prycker, *Asynchronous Transfer Mode*. Ellis Horwood, 1991.

[9] A. Campbell, G. Conlson, F. Garlia, D. Hutchison, and H. Leopold, "Integrated quality of service for multimedia communications," in *Proc. IEEE INFOCOM '93*, pp. 732–739, 1993.