Distributed Synchronization Protocols for Multimedia Services on Internet

Zafar Ali, Miae Woo and Arif Ghafoor
Distributed Multimedia Systems Lab
School of Electrical Engineering, Purdue University
West Lafayette, Indiana 47907

Abstract

In this paper we present a distributed architecture for the existing and the emerging Internet for providing *Synchronized Virtual Channels (SVCs)* to support presentation of multimedia information. The *SVC Architecture (SVCA)* provides both intra-stream and inter-stream synchronization using resources available over the Internet and implementing distributed transmission scheduling mechanisms. We also propose a set of *Quality Of Presentation (QOP)* parameters that quantify the quality of multimedia presentation from the user's point of view. Based on these parameters, we evaluate the proposed architecture and provides trade-offs between the QOP parameters and the required network resources to maintain this quality. Subsequently, these trade-offs are used to generate an optimal schedule for transmission of multimedia information over the SVCA. We also present protocol mechanisms to realize the SVCA. [1]

1 Introduction

As we enter into the era of multimedia information, tremendous interest has emerged in supporting multimedia applications over the Internet [3]. Most of these applications/services will use some form of *pre-orchestrated* information stored at various sites [5]. Distribution and access of pre-orchestrated multimedia information poses a whole new set of networking challenges that are different from the ones faced in the present day Internet.

One of the major issues is *temporal synchronization* of multimedia information units as they are transmitted over the network. This synchronization is generally of two types: *Intra-stream* and *Inter-stream* [6]. Intra-stream synchronization is used to smoothen the delivery of multimedia information and to minimize the impact of jitter delays in the network. On the other hand, inter-stream synchronization is needed to ensure concurrent presentation of multiple data streams as they are communicated over different channels. Maintaining *lip-sync* between audio and video data in video-on-demand applications is an example of inter-stream synchronization requirement.

Recently several schemes have been proposed in the literature to provide end-to-end synchronization over a network. One such approach uses a single Virtual Circuit (VC) for a point-to-point connection [4]. The VC delivers these streams in order, thus facilitating temporal synchronization. However, multiplexing of diverse data streams cannot meet media-specific requirements, i.e., *Quality of Service*, resulting in an under-utilization of the network resources. Alternatively, independent virtual channels can be used for the transmission of all the related objects [6]. In this technique, synchronization is maintained via buffering at the destination.

In the Internet environment, end-to-end delay variations (jitter) over individual channels make it difficult to maintain intra-stream synchronization. Furthermore, the end-to-end delays over independent virtual channels may vary widely, requiring extensive buffering at the destination to maintain inter-stream synchronization. Such buffering can be prohibitively large [3]. It is well known that in such an environment, the related streams transmitted over different channels eventually drift out of synchronization due to failure in maintaining network throughput and due to discrepancies among distributed clocks [3]. In such a situation, rate feedback technique to regain synchronization is proposed in [6] and is shown to be effective in a LAN environment. However, this technique is not be feasible when the source and the destination are geographically distributed over a wide area, as is

[1] This research was supported in part by the National Science Foundation under grant number CDA-9121771, awarded to School of Elect. Engr., Purdue University.

the case of the global Internet.

The objective of this paper is to design mechanisms and network protocols for providing both intra-stream and inter-stream synchronization over the Internet. For this purpose, we introduce the concept of end-to-end *Synchronized Virtual Channels*, that incorporates the strengths of both single VC and multiple VC schemes and makes it possible to implement rate feedback techniques more effectively over the Internet.

2 Synchronization Requirements of Multimedia Information

In this section, we first describe the concept involved in communication and presentation of pre-orchestrated multimedia information. Subsequently, we define the QOP parameters characterizing the presentation requirements and media-specific synchronization constraints.

2.1 A Model for Communication of Multimedia Information

In order to satisfy some pre-specified temporal constraints at the time of presentation of pre-orchestrated multimedia information, models both at the presentation level and the network level are needed. Several models have been proposed for specifying temporal constraints among multimedia objects, e.g., Hytime, Petri-net based models, MODE, G-Net (See [2] for a survey and comparison of these models). Unfortunately, most of these models cannot be used directly to specify network synchronization requirements. In these models temporal specification is provided at the object level where objects can be of variable duration, such as a video clip, an audio segment, or an image-sequence of arbitrary length. Synchronization at the object level is rather coarse and hence is difficult to perform synchronized transmission over the networks using these specifications. In a networked environment, synchronization needs to be performed at a finer level. For this purpose, we packetize each object in terms of *Atomic Units of Presentation (AUPs)*, of equal duration that constitute the minimum data unit which is perceivable to the user. For example, a frame can be considered as an *AUP* for video object, while a packet containing equal number of PCM code samples can be taken as an *AUP* for audio objects. For discrete media objects such as image and text, the entire object can be viewed as an *AUP*.

To facilitate inter-stream synchronization, each object is divided into *AUPs* of equal duration. Associated with each *AUP* is a unique temporal interval that

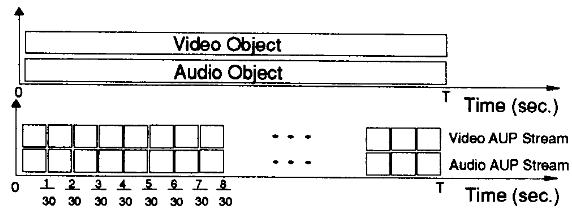

Figure 1: Temporal segmentation of multimedia objects into AUPs.

is a subinterval of the whole object. Each such interval has a relative deadline for presentation with respect to presentation start time of the related objects. This is illustrated in Fig. 1 where duration of each such temporal interval is chosen to be $\frac{1}{30}th$ of a second, which is the size of a video frame. *AUPs* can be used efficiently for controlled transmission of concurrent multimedia objects, thus facilitating inter-stream synchronization.

A strict presentation requirement is that all the *AUPs* belonging to the same temporal interval must be played out during their common temporal interval. To meet this requirement, it is important that the source deliver *AUPs* for all objects at a constant rate (in *AUP* per second). This requires controlling the transmission of individual *AUPs* that in turn needs the knowledge of the transmission deadline of each *AUP*. Therefore, if $d_i^{s,1}$ is the transmission deadline for the first *AUP*, then $d_i^{s,j} = d_i^{s,1} + (j-1)\tau_i^{\text{AUP}}$ is the transmission deadline for the jth *AUP*, where τ_i^{AUP} denotes the temporal interval of an AUP of the object O_i. Since *AUPs* of different objects are transmitted over independent channels, we would like to ensure that these objects have no *skew* at the time they arrive at the destination. Scheduling considerations to meet this requirement are discussed in Section 4.

2.2 QOP parameters and their Impact on Synchronization

In this section we formally define the proposed QOP parameters that are specified by the user and are used in determining resource requirements for the proposed SVCA.

2.2.1 Probability of Deadline Misses

It is desirable that multimedia objects be played out with zero skew. This requires that all AUPs meet their deadlines which cannot be guaranteed in an internetworking environment. Specifically, in these networks many users statistically share network resources. Therefore, jitter delay cannot be bounded from above. Hence, a fraction of the AUPs will miss their deadline. Remedial strategies such as blocking or restricted blocking, can be used in case of deadline misses.

Synchronization failures due to deadline misses can be quantified in terms of the *maximum tolerable probability of deadline misses*, $\mathcal{P}_i^d$. $\mathcal{P}_i^d$ represents the probability with which an AUP of object O_i can miss its deadline.

2.2.2 Probability of Buffer Overflow

Buffers can be used to smoothen out the effect of delay jitter. Early arrival of data can result in synchronization failure due to buffer overflow. This is a serious problem since data lost in this manner cannot be recovered. Retransmission strategies to overcome such failures are not feasible in multimedia applications. Data loss due to buffer overflow can be quantified in terms of the *maximum tolerable probability of jitter compensation buffer overflow*, $\mathcal{P}_i^b$.

2.2.3 System Response Time

Pre-orchestrated multimedia applications are more adaptive to the networking environment in the sense that instead of requiring the network to make strict performance guarantees (which is unrealistic for the Internet), the applications can accommodate themselves to the network characteristics. This is because in these applications, data is not generated in real-time, rather it is pre-stored. This gives the source more latitude in scheduling the times at which data is transmitted to the destination. Furthermore, it is important to note that, unlike live transmissions (such as video conferencing) where the response time is the critical factor, pre-orchestrated stored multimedia applications are more tolerant of network delays. Although quick response is desirable, smoothing of data stream and compensation for jitter delays is more important for such applications.

3 Synchronized Virtual Circuits Architecture (SVCA)

In order to support multimedia applications over the Internet, in this section we propose an architecture called *Synchronized Virtual Channels Architecture (SVCA)* which is a value-added feature for the Internet that can be used to regulate the flow of multimedia streams over subnets. The architecture consists of nodes, which we call *Intermediate Synchronization Sites (ISSs)*. The case where two ISSs are used to provide an end-to-end SVC connection is shown in Fig. 2. For the Internet, the gateways at the subnet level are the most suitable choices to act as ISSs. These ISSs provide necessary capabilities including buffering and scheduling the transmission of multimedia streams to smoothen jitter delay (for intra-stream synchronization) and to reduce skew (for inter-stream synchronization) that may be caused by unpredictable behavior of the subnets. For this purpose, the ISSs coordinate with each other to maintain synchronization among related streams. Each ISS stores part of the multimedia objects and forwards it to the next ISS according to some pre-determined schedule. The reason for storing and forwarding objects by the ISSs is to provide compensation for jitter delays so that objects are delivered in a synchronized form to the destination with the desired QOP. It is, therefore, important that the selected ISSs not only have proper resources (buffering/ channels) but also have the capability to transmit multimedia objects according to some pre-determined schedule. These functionalities can be implemented by some protocol entities (discussed in Section 5) that are created at each ISS at the time of connection establishment. Typically, a protocol entity is a process that interacts with the local node to secure resources and to generate transmission schedules for multimedia objects. The overall procedure for the channel establishment including generation of these protocol entities and reservation of proper resources is discussed in Section 5.

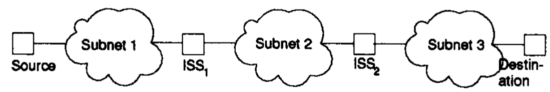

Figure 2: The SVC Architecture

The resources allocated at each ISS must be sufficient to meet the desired end-to-end QOP. This requires decomposing of the overall QOP parameters ($\mathcal{P}_{i,T}^d$, $\mathcal{P}_{i,T}^b$) into equivalent parameters ($\mathcal{P}_{i,j}^d$, $\mathcal{P}_{i,j}^b$) at the subnet level. It is obvious that the QOP parameters at the subnet level are related to the overall QOP parameters by the following relations:

$$\begin{aligned} 1-\mathcal{P}_{i,T}^d &= \prod_{j=1}^m (1-\mathcal{P}_{i,j}^d) \\ 1-\mathcal{P}_{i,T}^b &= \prod_{j=1}^m (1-\mathcal{P}_{i,j}^b). \end{aligned} \tag{1}$$

Where m represents the total number of ISSs, including the destination. The values of $\mathcal{P}_{i,j}^d$ and $\mathcal{P}_{i,j}^b$ can be calculated on the basis of the buffering resources available at each site. This non-uniform assignment

Symbol	Explaination
s_i	Size of object O_i
μ_i	Transmission/ consumption rate
$\mathcal{P}_i^d$	probability of deadline misses at the RxISS
$\mathcal{P}_i^b$	probability of buffer overflow at the RxISS
$k_i = \frac{K_i}{s_i}$	Fraction of the object that can be buffered
AUP_i^j	jth AUP of object O_i
$\tau_i(\tau_i^j)$	Duration of object O_i (AUP_i^j)
$d_i(d_i^j)$	Deadline of object O_i (AUP_i^j)
$d_i^s(d_i^{s,j})$	Control time for object O_i (AUP_i^j)
$\mathbf{D}_i$	ISS-to-ISS delay
$F_{\mathbf{D}_i}(\cdot)$	ISS-to-ISS delay distribution

Table 1: Notations

of QOP parameters is desirable as subnets may have heterogeneous resource capabilities. Subsequently, the QOP parameters of each subnet can be used to generate the transmission schedule and to calculate the buffering required at each ISS. In the next section we present a detail analysis for this purpose.

Another important feature of the SVCA is that it can be viewed as a jitter control channel, where instead of concurrent objects, a single object is transmitted. This is because each ISS through its buffering capability reduces jittering between consecutive AUPs. As a result of this smoothing, the SVCA gives an illusion of a constant delay channel that satisfies the desired QOP. The SVCA provides a jitter control channel in an application specific manner over the Internet where subnets can only provide best effort delay channels.

As we show in Section 5, the proposed SVCA can help in implementing distributed rate feedback techniques to detect and rectify any asynchrony since it exercises a finer control at the ISS level rather than a coarse control at the Internet level. Furthermore, it evenly spreads out synchronization buffering requirement over the Internet thereby reducing the buffering requirement at the destination.

4 Resource Requirement for the SVCA

The SVCA requires proper reduction of the jittering effect of each subnet with a guaranteed QOP at the subnet level. This in turn requires a proper selection of control time and explicit resource reservation at each ISS – the subject matter of this section. Due to the space limitation, proofs of the theorems have been skipped. They can be found in [1].

We need few notations for our analysis that are summarized in Table 1. Bold faced letters denote the random variables.

4.1 Object Scheduling: AUP Deadline Consideration

For synchronized delivery of an AUP, say AUP_i^j over the SVCA, it is required that sufficient time (control time, T_i^j) must be allowed to overcome the random delay ($\mathbf{D}_i^j$) experienced by the AUP_i^j during its transmission from the transmitter ISS (TxISS) to the receiver ISS (RxISS). As the ISS-to-ISS delays are random, determination of the T_i^j should be governed by the synchronization requirements of the multimedia object O_i. The following theorem states the minimum possible control time that satisfies the synchronization constraints defined by the maximum tolerable probability of deadline misses.

Theorem 4.1 *Let $\mathcal{P}_{i,j}^d$ denote the maximum tolerable probability of deadline miss for the AUP_i^j. Then the control time T_i^j that guarantees this bound is given by:*

$$T_i^j \geq F_{\mathbf{D}_i}{}^{-1}(1 - \mathcal{P}_i^{j,d})$$

We assume that the tolerable probability of deadline miss is same for all AUPs. Under this assumption, we can find the following bound on the control time T_i, using Theorem 4.1.

$$T_i \geq F_{\mathbf{D}_i}{}^{-1}(1 - \mathcal{P}_i^d). \quad (2)$$

Increasing control time reduces probability of deadline misses but it also increases both the buffering requirements and the system response time. In the next section we evaluate the object scheduling process with respect to the tolerable buffer overflow and find the size of the RxISS buffers to keep overflow below this value.

4.2 Synchronization Buffering Requirements at the RxISS

In the following theorem, we find the expected size of the buffer needed by an ISS in order to keep the deadline misses above some tolerable limit $\mathcal{P}_i^d$. Later in Section 4.2.4, we find the actual buffer size needed to maintain the buffer overflows below this limit.

Theorem 4.2 *The expected fraction of the object that needs to be buffered at an ISS for jitter compensation* is given by

$$E[\boldsymbol{\xi}_i] = min(\bar{\gamma}_i(1 - \mathcal{P}_i^d), 1), \tag{3}$$

where γ_i denotes the random fraction of the time for which the AUP has to wait at the ISS buffer before it becomes schedulable and $\bar{\gamma}_i$ is its expected value. $\bar{\gamma}_i$ can be interpreted as the jitter compensation factor to reduce the jittering effect over the presentation and is given by

$$\bar{\gamma}_i \triangleq E[\gamma_i] = \frac{T_i - \bar{D}_i}{\tau_i}$$

In the following subsections, we use the above result to evaluate the actual buffer size required to limit both $\mathcal{P}_i^d$ and $\mathcal{P}_i^b$.

4.2.1 Object Scheduling: RxISS Buffering Consideration

For a fixed buffer size at an ISS, we need to avoid excessive pre-fetching of the object prior to its deadline, otherwise buffer overflow can occur. This in turn bounds T_i as stated in the following theorem.

Theorem 4.3 *In order to bound the probability of buffer overflow at an ISS within a specified limit $\mathcal{P}_i^b$, T_i should satisfy the following inequality*

$$T_i \leq F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b) + k_i\tau_i, \tag{4}$$

where k_i denotes the fraction of the object that can be buffered at the ISS.

In Equation (4), the term $k_i\tau_i$ represents the portion of the total forwarding duration of the object for which the object can be stored in the buffer. $F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b)$ denotes the relaxation on control time for tolerating buffer overflow.

4.2.2 Performance Trade-offs

In this section we elaborate the effect of T_i on the QOP parameters ($\mathcal{P}_i^d$, $\mathcal{P}_i^b$), and the required buffering at an ISS. We consider an example of transmission of a video clip consisting of 18,000 frames from the TxISS to the RxISS over a subnet. For the purpose of illustration of the numerical results, we choose Gaussian distribution (with mean 350 millisecond and variance 100 millisecond) as the delay distribution.

For this example, the effect of T_i on the deadline misses at the RxISS is shown in Fig. 3(a). It is easy to see from the figure that an increase in the control time reduces the risk of missing deadline. This increase in

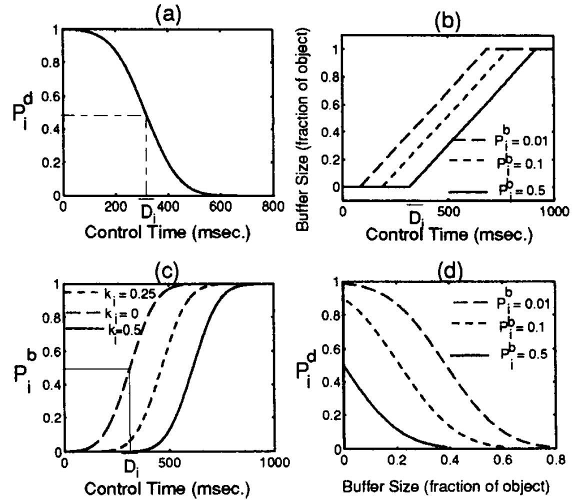

Figure 3: Trade-offs between ISS buffer requirements and the QOP.

the control time, however, requires more buffering at the RxISS for any given value of $\mathcal{P}_i^b$ as can be seen from Fig. 3(b). Fig. 3 (d) indicates that in order to reduce $\mathcal{P}_i^d$, we must pre-fetch data earlier enough which in turn increases the RxISS buffering needs. However, it is important to note that even for arbitrary low values of $\mathcal{P}_i^d$, we are not required to buffer the whole object. Hence, with a proper selection of T_i it is possible to reduce the RxISS buffering requirement by a significant amount. Determining this optimal value of the control time and the crossponding minimum buffering requirement at an ISS is the subject matter of the Section 4.2.4.

4.2.3 Schedulability Condition and Schedulable Region

To this end, we have showed that a given $\mathcal{P}_i^d$ forces a lower bound on T_i, while a constraint on $\mathcal{P}_i^b$ does not allow T_i to exceed some value. Therefore, there exists a range of T_i for which the object O_i is 'schedulable'. The following theorem establishes a condition for the schedulability of the object O_i.

Theorem 4.4 *Given a finite buffer space $k_i \leq 1$ at an ISS, and the desired values of $\mathcal{P}_i^d$ and $\mathcal{P}_i^b$, the object O_i is schedulable iff*

$$F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b) + k_i\tau_i \quad \geq \quad F_{\mathbf{D}_i}{}^{-1}(1 - \mathcal{P}_i^d) \tag{5}$$

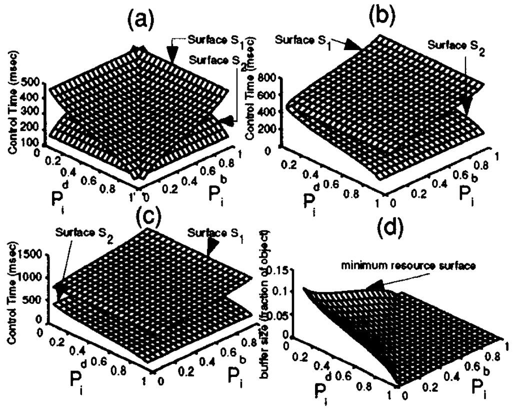

Figure 4: Schedulable regions (a) $k_i = 0$, (b) $k_i = 0.1$, (c) $k_i = 0.2$, (d) Minimum resource surface (k_i^{min}).

The inequality in this theorem can be used to identify the schedulability region, in a three dimensional space spanned by the tuple $(\mathcal{P}_i^b, \mathcal{P}_i^d, T_i)$ for which the transmission of O_i is feasible. These regions are shown in Fig. 4, for the example used in Section 4.4.1. Each plot shows two surfaces, an upper surface $\mathcal{S}_1$, defined by the equation $T_i = F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b) + k_i\tau_i$ and a lower surface $\mathcal{S}_2$, defined by $T_i = F_{\mathbf{D}_i}{}^{-1}(1 - \mathcal{P}_i^d)$. The values for tuple $(\mathcal{P}_i^b, \mathcal{P}_i^d, T_i)$ that lie in region defined by the intersection of the $\mathcal{S}_1$ and the $\mathcal{S}_2$ (above $\mathcal{S}_2$ and below $\mathcal{S}_1$) correspond to a feasible transmission schedule. The values of the QOP parameters in an unschedulable region cannot be guaranteed. The size of the schedulable region represents the flexibility in choosing arbitrary values for QOP parameters. This size increases with an increase in the buffer size. For example, with an increase in the buffer size, an object with a given QOP that was unschedulable in Fig. 4(a) becomes schedulable in Fig. 4(b) and Fig. 4(c).

4.2.4 Optimal Control Time And Minimum Buffer Requirements

In order to maintain the desired quality of presentation at each subnet level, it is important that some minimum amount of buffering at each ISS should be allocated. The following theorem states a bound on this minimum buffer requirement to maintain a given QOP level defined by $\mathcal{P}_i^d$ and $\mathcal{P}_i^b$.

Theorem 4.5 *For a desired level of QOP defined by $\mathcal{P}_i^d$ and $\mathcal{P}_i^b$, the minimum fraction of the object O_i that needs to be buffered at each ISS, including the*

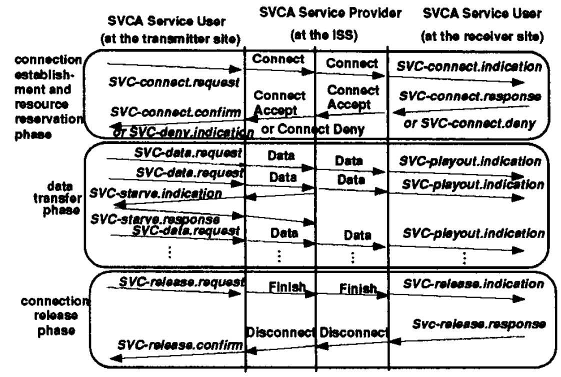

Figure 5: Protocol mechanisms for an SVC connection.

destination is given by

$$k_i^{min} = \frac{F_{\mathbf{D}_i}{}^{-1}(1-\mathcal{P}_i^d) - F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b)}{\tau_i} \quad (6)$$

The control time that achieves this value and the desired QOP is

$$T_i^{c,opt} = F_{\mathbf{D}_i}{}^{-1}(\mathcal{P}_i^b) + k_i^{min}\tau_i \quad (7)$$

According to this theorem, if an ISS can buffer k_i^{min} fraction of the object then scheduling the object with T_i^{opt} guarantees the desired QOP. Fig. 4(d) shows this minimum buffer against QOP parameters. The surface shown in this figure represents the minimum size of the RxISS buffer needed for a feasible schedule.

5 Protocol Mechanisms for an SVC Connection

In this section, we describe a typical sequence of events for the establishment and maintenance of an SVC involving m sites, including the destination site[2]. The channel is used for the presentation of n concurrent objects. Temporal relationships among these objects need to be observed during transmission. The case when $m = 3$ is depicted in Fig. 2.

5.1 Call Establishment and Resource Reservation Phase

We now present a protocol for the establishment of an SVC call. The protocol is independent of the

[2]When $m = 1$, there is no ISS between the source and the destination. Synchronization is performed only at the destination.

Primitives	Parameters
SVC-connect. request	source-address, destination-address, svc-id, QOS-requirements, $\tau_i, \tau_i^{\text{AUP}}, s_i,\ \forall i, 1 \leq i \leq n$, QOP-tolerance-level $(\mathcal{P}^d_{i,T}, \mathcal{P}^b_{i,T},\ \forall i, 1 \leq i \leq n)$.
SVC-connect. indication	All the above parameters plus $\hat{k}_j,\ \forall j, 1 \leq j \leq m$.
SVC-connect. response	svc-id, $\mathcal{P}^d_{i,j}, \mathcal{P}^b_{i,j}$, $k_j,\ \forall j, 1 \leq j \leq m$.

Table 2: Connection establishment primitives and their associated parameters.

underlying network routing mechanism. A set of service primitives that are used to establish an SVC is shown in Fig. 5. The parameters associated with these primitives are given in Table 2. The steps for an SVC establishment and resource reservation phase are as follows:

1. A channel establishment message (*SVC-connection.request*), carrying synchronization information associated with each multimedia object of the OCPN, is transmitted from the source to the destination. This message propagates from ISS to ISS and delivers its parameters mentioned in Table 2. This message is used to reserve n independent virtual channels, each satisfying the QOS requirements of one object, within each subnet.

2. Upon the reception of *SVC-connect.request*, each ISS (say ISS_j) tentatively reserves some buffers for each object. For this purpose, ISS_j estimates its QOP parameters ($\hat{\mathcal{P}}^d_{i,j}$, $\hat{\mathcal{P}}^b_{i,j}$) by equally dividing the overall QOP requirement ($\mathcal{P}^d_{i,T}$, $\mathcal{P}^b_{i,T}$) among $\hat{m}$ sites, where $\hat{m}$ is the worst case estimate of the number of ISS in an SVCA[3]. This estimate is given by

$$\begin{aligned} \hat{\mathcal{P}}^d_{i,j} &= 1 - \sqrt[\hat{m}]{1 - \mathcal{P}^d_{i,T}} \\ \hat{\mathcal{P}}^b_{i,j} &= 1 - \sqrt[\hat{m}]{1 - \mathcal{P}^b_{i,T}} \end{aligned}$$

Each ISS then appends information about the maximum buffering ($\hat{k}_j$) it can provide to the call.

3. The destination, upon receiving the message, calculates the QOP parameters for each ISS, (including itself) based on the overall QOP parameters ($\mathcal{P}^d_{i,T}$, $\mathcal{P}^b_{i,T}$), the buffer size ($\hat{k}$) that is tentatively reserved by each ISS , the number of ISS (m) in the transmission path and the constraint defined in Equation (1) of Section 3.

4. Next, the destination sends a message, *SVC-connect.response*, containing $\mathcal{P}^d_{i,j}$, $\mathcal{P}^b_{i,j}$, $\forall j, 1 \leq j \leq m$, back to the source through the selected ISSs.

5. When an ISS receives the *SVC-connect.response* message, it calculates the control time and actual buffering requirement based on Equation (7) and Equation (6) , respectively. It then passes the message on to the previous ISS (towards the source). Once the control time for each object is known, each ISS (including the source and the destination) can compute the complete schedule for each AUP based on the procedure outlined in Section 2.1.

6. In case the call is denied due to some reason such as lack of resources, the message *SVC-connect.deny* is used to release resources reserved for this call.

[3]An apriori knowledge at the ISS_j regarding the number of ISSs from its location to the destination may not be available during this phase.

5.2 Multimedia Data Synchronization Protocol

Once the SVC connection is accepted, each ISS including the source and the destination knows its transmission schedule. When the source receives the *SVC-connect.confirm* message, it starts transmission of objects at the time taken as zero with respect to its own clock. As discussed in the previous section, the transmission start time for each object depends on its control time. The source maintains transmission of *AUPs* according to its own schedule unless it receives some feedback signal from its neighboring ISS to change the transmission rate (see Section (5.2.1) for rate feedback). To facilitate both intra-stream and inter-stream synchronization the source stamps *AUPs* with a sequence number.

As timing relationships among objects are relative rather than absolute, the schedule at each ISS, including the destination, needs the control time information only for each media object. Two approaches can be used for this purpose. One approach is to use a delay stamp field that indicates the random delay that an *AUP* encounters during its transmission over a subnet. Its value is set to zero by the TxISS. Each node in the subnet increments the stamp by the amount of time that each packet has spent in that node (including propagation delays over the links). When an *AUP* arrives at an ISS, its delay stamp contains the delay value that the *AUP* has actually encountered over the subnet connection. Based on this delay value, the *AUP* sequence number and its transmission time in the transmission schedule, the ISS computes the wait

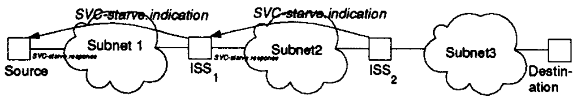

Figure 6: Feedback signal propagation in the SVCA

time for this *AUP* that is needs to be buffered. Upon expiration of this time, the ISS forwards the *AUP* to the next ISS with the delay stamp field reset to zero. If the *AUP* is already late, ISS immediately forwards it with delay stamp field containing the delay value by which it is late. This can help improving the deadline misses over the stream. Equivalently, if the network service provider runs some clock synchronization protocol, instead of the delay stamp field, the time stamp field can be used.

Each ISS not only provides buffering but also performs some monitoring and control actions. These actions are discussed in the following section.

5.2.1 Adaptive Feedback Technique

It is quite possible that related connections drift out of synchronization due to changes in delays over subnet with time, violation of bandwidth guarantee, discrepancies between remote clock rates, etc. Hence, the control time calculated to determine the start of transmission may become inaccurate later during the session. This can result in synchronization buffer to underflow (*starvation*) or overflow (*flooding*). This is especially true for the applications that involve presentation of multimedia objects with long duration, e.g., video-on-demand.

In order to remedy this problem, a feedback mechanism can be employed. In [6], a feedback scheme to compensate for the end-to-end delay and object playback rate variations is presented and shown to be effective in a LAN environment. In this scheme, a high priority feedback signal to the source is used to correct the asynchrony. As mentioned earlier, such technique may not be suitable when the source and the destination are geographically distributed over a wide area, as is the case with the global Internet. This is because, in such an environment a large amount of data may have already been in transit and any synchronization action taken by the source may be too late. All of the in-transit *AUPs* would have to be played back at the destination before synchronization action performed by the source becomes effective.

For the Internet, the proposed SVCA can provide an additional advantage. In this environment, the feedback signalling can be implemented more efficiently by distributing the mechanism over the ISSs as depicted in Fig. 6. Implementation of distributed feedback signalling is practicable in the SVCA since each ISS buffers a part of the object and has a *apriori* knowledge of its transmission schedule. Hence, each ISS can detect asynchrony and can take appropriate control actions (like the one proposed in [6]) rather than letting asynchrony to build till the destination site runs into serious synchronization problems.

6 Conclusion

In this paper, we have presented a distributed architecture for supporting presentation of concurrent multimedia objects over the Internet, in an efficient manner. The proposed architecture uses gateways in the Internet as synchronization sites that control jitter delays and skew so that from end-to-end point of view, both inter-stream and intra-stream synchronization are guaranteed.

Acknowledgments: The authors would like to thank S. R. Bavishi, M. F. Hameed, M. Tariq and the anonymous reviewers for their valuable comments and advice in preparing this document.

References

[1] Z. Ali, M. Woo and A. Ghafoor, "A Distributed Architecture for Synchronized Multimedia Services over the Internet," *Tech. Report*, Purdue Univ., April 1994.

[2] R. Erfle, "HyTime as the Multimedia Document Model of Choice", *Proc. IEEE Int. Conf. Multimedia Comp. and Sys.*, May 1994, pp. 445-454.

[3] D. Ferrari, "Distributed Delay Jitter Control in Packet-switching Internetworks," *Internetworking*, v. 4, n. 2, March 1993, pp. 1-20.

[4] W. F. Leung, T. J. Baumgartner, Y. H. Hwang, M.J. Morgan and S. Tu", "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks," *IEEE j. select. Areas Commun.*, v. 8, n. 3, April 1990, pp. 380-390.

[5] T. D. C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," *IEEE j. select. Areas Commun.*, v. 9, n. 9, Dec. 1991, p. 1368-1382.

[6] S. Ramanathan and P. V. Rangan, "Adaptive Feedback Techniques for Synchronized Multimedia Retrieval over Integrated Networks", *IEEE/ACM Trans. Networking*, v. 1, n. 2, April 1993, pp. 246-259.