Synchronization Techniques in Distributed Multimedia Presentation

Shahab Ud Din
Department of Computer Science
Vrije University
Amsterdam, the Netherlands
s.uddin@student.vu.nl

Dick Bulterman
(SEN5) Distributed and interactive systems
Centrum Wiskunde & Informatica (CWI)
Amsterdam, the Netherlands
dick.bulterman@cwi.nl

***Abstract*— In the last two decades, the transmission of multimedia streams using best effort network has been an attractive research area in multimedia communication. Multimedia streams have well defined temporal relations within themselves, generated when captured at the sender. At receiver these temporal relations have to be reconstructed to ensure smooth and synchronized multimedia presentation. The characteristics of best effort network –delay and jitter- degrade the temporal relations present in multimedia streams. Many methods have been proposed in order to mitigate the effect of network delay and jitter on the media streams. This paper classifies the work in the field of distributed multimedia synchronization. We have illustrated the techniques used in the three different multimedia synchronization types, namely, intra-media synchronization, inter-media synchronization and inter-destination synchronization.**

Keywords-distributed; multimedia; jitter; temporal relations; synchronization.

I. INTRODUCTION

Due to the last decade's breakthrough in the communication technologies, new applications in the area of distributed multimedia communication emerged. Distributed multimedia applications like video conferencing, video on demand, distance learning and others, are made feasible due to developments in the communication network. In such applications, at sender's side, different media streams are captured and sent to the receiver via packet switching network. On the receiver side, streams are received for presentation. These media streams can be classified into continuous and non-continuous streams. The continuous media streams have well defined temporal relations between the subsequent *Media Units (MUs*), for example, audio and video streams. The non-continuous media streams like images, text and graphics do not have temporal relations among MUs.

Multimedia presentation requires the integration of multiple media streams of both continuous and non-continuous streams. These streams have different temporal dependencies among the MUs of one or multiple streams. To ensure these relationships between the MUs of single and/or multiple media streams, a coordination process is required, which is called the *multimedia synchronization*. Typical synchronization solutions can be classified in to two basic types: (1) Intra-media synchronization deals with the reconstruction of the temporal relations between the MUs of the same media stream, at the presentation time. For example, maintaining the frame sequence and frame rate of the video stream to ensure a smooth presentation. (2) During presentation, reconstruction of the temporal relations between the MUs of the different but related media streams is referred as Inter-media synchronization. A typical example of the inter-media synchronization is lip synchronization [1] between the corresponding audio and video stream.

Developments in computer and communication technology led to the popularity of distributed multimedia applications. In these applications, a geographically separated sender and receiver are linked via a communication network. The sender is capturing the media stream with temporal relations and sending to receiver(s), which have to ensure these relationships during the presentation. The unreliability and unpredictability of best effort packet switching network make it hard for receiver to keep intact relations between the one or multiple streams. An accurate and explicit process of restructuring of the MUs at the receiver is required, which is called *distributed multimedia synchronization*. In distributed multimedia environment, apart from the two basic synchronization problems described earlier, another type of synchronization is required in case of multicast communication and is called inter-destination or group synchronization. This is required when geographically scattered group of receivers have to present the same stream(s) approximately at the same. With the emergence of Interactive Distributed Multimedia Applications (IDMA) a new type of interactive synchronization emerges and examples are [2-6]. In these types of applications, users can modify the presentation state of stream and this modification has to be communicated to all receivers to maintain the synchronized view of the presentation among them.

This survey is intended to study and classify research in the three types of synchronization solutions. The main objective is not to compare their techniques, but to classify them in such a way that is easy to comprehend for the multimedia research community. Although the classifications of the techniques presented in this paper are neither exhaustive, nor orthogonal, still they can act as a very good starting point for the researchers in field of distributed multimedia. The rest of the paper is structured as follows. In Section 2, we identify the causes of delay and present related work. Sections 3, 4 and 5 are dedicated to intra-media synchronization, inter-media synchronization and inter-destination synchronization techniques, respectively. The paper concludes in Section 6.

II. BACKGROUND

The current packet switching networks do not provide any guarantee on delay bounds of packet delivery. Rather they only promise best effort to deliver the data to the intended recipient. This characteristic of packet switching networks make the success of the distributed applications challenging. It causes asynchrony (de-synchronization) in Distributed Multimedia Applications. In the following section, we will briefly discuss the factor of asynchrony.

A. Causes of Asynchrony

MUs of the media stream suffer different type of delays from the generation at source to presentation at receiver. These delays can be different for different MUs depending upon the load at sender, network and receiver. These delay variations for different MUs cause asynchrony in the media presentation at the receiver. We can divide delays into three types: the delay caused by sender, network and by the receiver. Figure1 gives a pictorial representation of all three components of end-to-end delay.

Delay at sender: Capturing, coding, packetizing, protocol layer processing and transmission-buffering delays depend on the sender load and clock speed. At the different time instances, the sender may have different loads variations, which can cause the variation in these delays for different MUs. Moreover, if the related sub-streams are captured or/and sent by different sources, then, these delays experienced by different sub-stream can be more variable.

Network delay: Network delay is the delay experienced by the MUs in the network to reach its receiver, which varies according to network load. This delay can include the propagation delay and queuing delay at the intermediate routers. Network jitters is delay variations of inter-arrival of MUs at the receiver due to varying network load. This is due to the fact that the queues of the intermediate routers between sender and receiver may have different loads at the different time instances. This delay can cause intra-media asynchrony. Network skew is the time difference in arrival of temporally related MUs of different but related streams, i.e. differential delay among the streams, which can cause inter-media asynchrony. Clock drift is the rate of change of clock skew because of temperature differences or imperfections in crystal clocks. Clock skew is the clock time difference between the sender and the receiver. This is possible if the sender and the receiver are using local clock information instead of global clock information. The sender and receiver are considering time synchronized with respect to clock only if they are using the Network Time Protocol (NTP) or Global Positioning System devices.

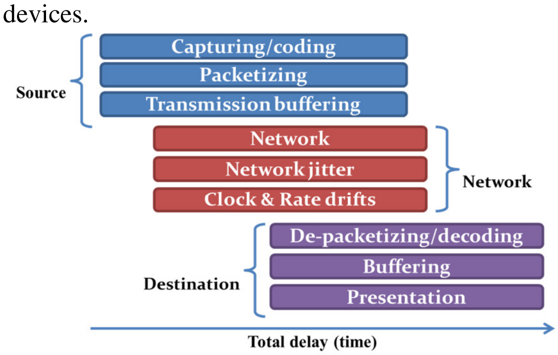

Figure 1. End-to-end causes of delay.

Delay at receiver: The presentation, decoding, depacketizing, protocol layer processing, and buffering delay at the receivers can be different for different MUs. These delay variations are present at the receiver due to the fact that different receivers may have different processing capabilities and different loads at the different time instance.

Depending on the nature of the application some or all of these problems may be relevant to different applications. Different synchronization mechanisms are needed to cope with these problems to ensure the temporal ordering of streams and to maintain the presentation quality.

B. Related Work

Most synchronization mechanisms in the literature are either very abstract, independent of the application at hand or very application specific. There are some surveys of multimedia synchronization mechanisms [24, 28, 30, 31], which either are specific to type of synchronization, or partly cover synchronization mechanisms.

Perez-Luque et al. presented a survey of multimedia synchronization in term of temporal specifications [31]. They presented a theoretical reference framework to compare temporal specification schemes and their relationship with multimedia synchronizations. Ehley et al. classify synchronization schemes as distributed schemes and local schemes depending upon the location of the sender and receiver [30]. They further classify the distributed schemes as “distributed protocol based”, “distributed among nodes” and “distributed on servers or co-processors”. Similarly, they classify the local schemes in to two categories namely “local at different level at workstation” and “local on servers or co-processors”. Ishibashi et al. present very comprehensive survey of only intra-media and inter-media synchronization schemes [28]. They classify the techniques into common control, basic control, preemptive control and reactive control schemes. They also compare the different algorithms in terms of location, clock information, and type of media. They did not include inter-destination synchronization, as it was not very matured at that time. Similar to their pattern, Boronat et al. [24] present a recent survey, which includes the inter-destination synchronization, but exclude intra-media synchronization. To the best of our knowledge, there is no single survey, which covers all the three types of multimedia synchronization. Our effort is the first attempt in this regard.

III. INTRA- MEDIA SYNCHRONIZATION

The reconstruction of temporal relations between media units of the same continuous media stream is referred to as intra-media synchronization. For audio streams, the basic media unit is audio sample. The spacing between samples is determined by the sampling process. The goal of inter-media synchronization is to ensure the same spacing at the presentation time. For video streams, the basic media unit is the video frame and the temporal relation is the frequency of the video frames. The frame rate determines the spacing between the

 ISBN: 978-1-61208-195-3

frames. At presentation time, similar frame-rate has to be ensured by reconstructing the temporal relationship.

Many schemes have been proposed in literature to ensure the temporal relationship at presentation time. All the schemes use a receiver buffer for the temporary storage of incoming MUs. The audio/video samples/frames are then presented at appropriate time from buffer. The use of a MU buffer introduces delay in the application, which is directly proportional to the size of this buffer. The objective of the process is to provide a presentation that resembles as good as possible to the temporal relations that were created during the encoding process.

All Distributed Multimedia Applications (DMAs) have their own end-to-end delay tolerance requirement [33] that depends upon the nature of the application. Interactive bidirectional applications such as online quizzes have very strict end-to-end delay requirements and the applications like video conferencing have slightly less strict latency requirements. On the other hand applications like video on demand (VOD) can allow larger latency. All the proposed schemes provide for a compromise between the intra-media synchronization quality and the increase of end-to-end delay due to the buffering of MUs. On one extreme, there can be a buffer less scheme with minimum delay by presenting the frame as soon as they arrive and other can be assured synchronization that completely eliminate the effect of jitter on the cost of high end-to-end delay.

The perfect intra-media synchronization quality can be achieved by completely eliminating any kind of distortion in the temporal relationships of MUs and to completely restore the stream to its initial form. If the delay variability is unbounded, meaning that an infinitely long inter arrival period may appear, then no technique with a finite buffer can eliminate the distortion from the MUs. But, some assured services (QoS) guarantee the bounded network delay. In this case, one can achieve assured/perfect synchronization.

We divided the intra-media synchronization in to two basic categories: Time-oriented techniques and buffer-oriented techniques. In time-oriented techniques sender puts a time stamp on the MUs. The sender and the receiver use clock in order to measure the delay and jitter. Receiver on the basis of these measurements devises a technique to ensure synchronous presentation of streams. Buffer-oriented techniques do not use the clock rather they implicitly measure network delay and jitter by the occupancy of the receiver buffer. The summary is presented in Table 1.

A. *Time-oriented Techniques*

We divide time-oriented techniques into three subcategories, depending upon the timing information: techniques using global clock information, techniques using local clock information, and techniques using approximated clock information.

Techniques in which sender and receiver use some mechanism for the synchronization of their clock are said to use *global clock information*. The existence of having the globally synchronized clock allows the receiver to measure the exact network delay of MUs. Due to exact measurements of network delay, it can guarantee that MUs will be delivered and presented before or at the required time.

The techniques "using the global clock information" [7, 8] measure network delays of the first MU. They then add buffering delay in already measured network delay to compose it to total delay. They set the Maximum Delay equal to this total delay. The receiver keeps the first MU in the buffer for minimum of buffering delay time plus the extra interval before extracting from the head of the buffer for presentation. This extra buffering delay for the first MU protects the synchronization of the stream for the succeeding MUs. This way, it is guaranteed that no MU will experience a larger delay then the first MU, thus no loss of synchronization will occur. The amount of this extra buffering delay will decide the quality of synchronization. The larger extra buffering delay means assured synchronization and smaller means small end-to-end delay but no assurance of synchronization. The amount of this extra buffering delay can be adjusted according to the nature of the application. For more interactive application this amount can be set low.

TABLE 1: SUMMARY OF INTRA-MEDIA SYNCHRONIZATION

Type	*Sub type*	*Description*
Time-oriented	global clock information	Due to exact measurement of network transfer delay it can guarantee that MU will be deliver before a particular time.
	local clock information	Instead of delay duration it works on differential delay information. Due to absence of global clock it can guarantee bounded delivery.
	approximated clock information	Approximate clock information by RTT value between source and destination. Can give soft bound on MU delivery.
Buffer-oriented	Pause/drop MU	Measure the delay by buffer occupancy. Drop MU if the occupancy is high and pause when occupancy is low.
	dynamic regulation of MU duration	Instead of dropping/pausing the MU, it dynamically regulates the MU duration in accordance with buffer occupancy.

The global clock can provide the highest degree of precision in terms of clock synchronization. It is the technique which supports the strictest synchronization which requires all the MUs to be presented at a constant small delay. As a global clock is not always available, many of the techniques are based on the delay differences instead of absolute delays of MUs. In these techniques, the receivers decide presentation time for the frames using the timestamps, in varying network delay environment, in absence of global clock information on the sender and receiver end. The receiver estimates the one way network delay and its variability using local clock information. These techniques are suitable for applications that do not require a constant end-to-end delay. These techniques can be categorized as techniques *with local clock information* or without global clock information.

 ISBN: 978-1-61208-195-3

As these techniques [9-12] are based on the delay differences, the two clocks need not be synchronized because their offset will be canceled while calculating the timestamp differences. But the two clocks should not drift. In these techniques, the total delivery delay of MUs cannot be kept constant rather it will fluctuate due to changing network delay. In this way the requirement of the tradeoff between the synchronicity and delay will be relaxed. The network delay differences act as indication of the current level of the jitter between the source and destination and are used as the main parameter of these schemes.

Apart from the two above mentioned techniques there is another category of techniques using *approximated clock information*. These techniques do not require a global clock, so cannot guarantee constant end-to-end delay like the techniques based on global clock information. But, they are better than the techniques with local clock information, which only promise fluctuating end-to-end delay due to the variable network delay. In these techniques, the receiver establishes a total delivery delay by measuring round trip time (RTT) between the sender and receiver. The receiver ensures that no MU will be presented after maximum delay value calculated by some expression of the RTT between sender and receiver. As a result of this assurance, these techniques promise a soft delivery guarantee.

In [13, 14], by exchanging probe packets, a three way handshake protocol is established to measure the RTT value between the sender and the receiver. The receiver then synchronizes its clock with the sender' clock by adopting its local time as of the timestamps of the probe packets. The receiver uses RTT/2 as the estimate of the network delay and adds some delay component to achieve a fixed soft end-to-end delay. To update the RTT value according to the current network load, receiver sends the periodic probe packets. During all the communication period, the clock of the receiver is adjusted virtually with the sender's clock. Thus, the clock of the receiver is; RTT + additional delay time units behind the sender's clock. Due to this virtual clock synchronization of the sender and the receiver, these techniques are also considered as based on *virtual clocks*. Later the receiver decides about the action to take against the packet on the basis of the local clock. Packets arriving at the receiver with the timestamp larger than the local clock are buffered and the packets that arrive with timestamps smaller than the local clock are considered late. The packets are extracted from the buffer and played when the local clock is equal to their timestamps.

A part from time-oriented and buffer-oriented techniques, another classification of these techniques is possible on the basis of how the receivers deal with the late MUs. A technique is characterized as being *delay preserving,* if it does not present late MUs (MUs that have missed their scheduled time). In *none delay preserving* techniques, the receiver may accept and present a late MU, instead of discarding it, to protect the continuity of the stream from further degradation. These techniques are mostly applied with the time-oriented techniques, where the timing information is explicitly available.

B. Buffer-oriented Techniques

The class of buffer-oriented techniques deals with the fundamental synchronization/latency tradeoff but do not require timestamps of MUs or the use of any kind of clock information. Buffer-oriented techniques implicitly assess jitter by observing the occupancy of the receiver buffer instead of using timestamps. As these techniques do not rely on timing information, they cannot provide the absolute/constant end-to-end delay guarantee. The total end-to-end delay comprises of fluctuating network and buffering delay. The better stream synchronization quality can be obtained by increasing buffering delay, which will result in increased end-to-end delay. Using these techniques, delay performance can approach requirements of interactive applications but this cannot be guaranteed. Due to this lack of guarantee regarding the end-to-end delay, buffer-oriented techniques are usually employed in video applications, where the interactivity requirements are more relaxed than in audio applications. These techniques indirectly measure impact of the delay jitter on a receiver by observing the occupancy of the presentation buffer over time. The fundamental idea is to adjust the receiver's consumption rate of the frame according to the buffer occupancy. As a result of more frames in buffer, the receiver increases its consumption rate to avoid buffer overflow which will make the presentation smoother, while in case of less frames in the buffer the receiver will decrease its consumption rate to avoid buffer underflow, which will cause the increase in the presentation time of a frame and ultimately decrease the smoothness of the stream. Buffer-oriented techniques can be divided in to two broad categories: Pause/drop MUs techniques and Dynamic regulation of MUs duration.

In *Pause/drop MUs* techniques [15-18], the receiver pauses or drops the frame from the presentation buffer according to the occupancy of the buffer. If the buffer has the higher occupancy of the frame due to decrease in the network delay, it will discard the newest frames from the buffer considering them as late frames without using the timing information, which is the dropping of late MUs. Similarly, if the buffer is suffering from the underflow the receiver decrease its consumption rate by pausing the MUs in the buffer, which will increase the presentation duration of the MUs. In both cases, the objective is to bring the buffer occupancy between the underflow and overflow stage to present a continuous and synchronized presentation.

In [15, 16, 17], authors used a series of thresholds for every occupancy level and then associated these thresholds with counters for the derivation of the frame discard decisions. Biersack et al. [18] proposed a more complex technique for adopting the presentation schedule by associating it with the threshold for underflow: Low Water Mark (LWM), overflow: High Water mark (HWM) and also for Low Target Boundary (LTB) and Upper Target Boundary (UTB). For regulation of the buffering delay most buffer-oriented techniques used the approach in which, they increase or decrease delay in constant amounts, that equal the duration of a MU, for example, discarding the late frame from an overpopulated buffer results in sharp delay reduction of constant duration of the video frame. Similarly, in case of under flow buffer, the presentation resumes after one or multiple MU periods. The benefit of these techniques is the simplicity of implementation. The drawback is that human visual system can detect this abrupt degradation of the perceptual quality. This will be more evident in case of low frame rate, where single video frame carries enough information. To solve this abrupt degradation problem, techniques [19, 20], which use *dynamic regulation of MUs* duration, were proposed.

These techniques demonstrated an improvement in the perceptual quality of the video, which was achieved through a fine grained regulation of presentation durations, based on the current occupancy of the presentation buffer. In [19], the receiver employs progressively reduced presentation rates, to avoid large underflow discontinuities as the buffer occupancy drops below a threshold value. The threshold value is selected prior to the stream initialization and it remains constant irrespective of the network delay jitter. It regulates the tradeoff between stream continuity and reduction of presentation rate. This work is enhanced in [20] by introducing a window based approach in which the sender optimizes the stream quality by changing network delay jitter values. This window acts as a dynamic threshold. By using a neural network approach, the sender estimates the network delay characteristics and then regulates the window accordingly. The regulation of the window will implicitly change the threshold values for the buffer occupancy. It results in dynamic selection of presentation durations for the buffered frames.

IV. INTER-MEDIA SYNCHRONIZATION

The inter-media synchronization is concerned with maintaining the temporal and/or logical dependencies among several streams in order to present the data in the same view as they were generated. At the receiver, MUs will not arrive in synchronized manner due to jitter in the network. The temporal relationship within sub-streams is destroyed and the time gaps between arriving MUs vary according to the occurred jitter. Thus, a synchronized presentation cannot be achieved at the receiver, if arriving MUs of sub-streams would be presented immediately. Hence, intra-media and inter-media synchronization is disturbed. To mitigate the effect of the jitter, MUs have to be delayed at the receiver so that, a continuous synchronized presentation can be guaranteed. Consequently, MUs have to be stored in buffer and the size of the buffer will correspond to the amount of jitter in the network.

For example, in video conferencing applications speech and video MUs must have the temporal relationships at the time the streams were captured at source. These speech and video MUs captured at the same time have to be presented together at receiver. Like any two different streams, the audio and video stream can be affected by the network delay differently. If these streams would be presented without any synchronization mechanism at the receiver, the audio and the corresponding lip movement in the video will not be synched. This temporal relation between the audio and the video stream is called inter-media synchronization or Lip synchronization. A pictorial representation of lip synchronization is presented in Figure 2.

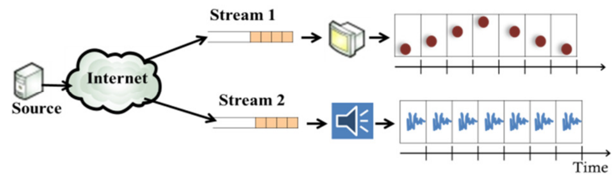

Figure 2. Inter-media synchronization.

The perfect inter-media synchronization quality is achieved by completely eliminating any kind of distortion in the temporal relationships of MUs among multiple streams and to completely restoring the stream to its initial form. This objective must be achieved on the fly as MUs arrive at the receiver, having crossed a network that alters the spacing between MUs, by imposing a variable network transfer delay.

There are many algorithms in literature that were applied in different applications to achieve the inter-media synchronization. Due to the different nature of the application, it is challenging to compare the performance of these algorithms. These algorithms used many synchronization techniques both at sender and receiver side. There is no benchmark found in literature to compare these techniques. Most of the algorithms evaluate their performance with the satisfaction of the users of the target application. So, instead of algorithm, we decided to survey these techniques that are the building blocks of algorithms. The study of inter-media synchronization technique is summarized in comprehensive manner in [24, 28, 29].

Several ways of classifying the technique are possible, we chose to categorize by location, purpose, type of content, and synchronization information. Before describing the categories of the technique, it is important to note that any algorithm can use multiple of these techniques to achieve the synchronization mechanism even from different categories. More over, these classifications are neither exhaustive, nor orthogonal, to each other, as one specific technique can be categorize according to the location, purpose, content and information used.

A. *Classification of Techniques*

Location of synchronization technique: The synchronization control can be performed either by source or receiver. The synchronization control on receiver is used more often as compare to source. If control is performed by the source, most of the time it will require some feedback information from the receiver. The receiver will tell the source about the degree of asynchrony at the current instance.

Purpose of synchronization technique: We divided the techniques into four sub categories with respect to its purpose: The *basic Control techniques* are required in almost all the algorithms. These must be present in all algorithms to provide synchronization. Examples are adding synchronization information in MUs at source and buffering of MUs at receiver. *Preventive control techniques* are used to prevent the asynchrony in the streams. These are applied to synchronized streams to keep them in the same state. *Reactive control techniques* are used to recover from the asynchrony, once it occurred. The *common control techniques* are techniques which can be applied in both ways.

Type of media: Some of the techniques are used only for stored media and some for live media, while some can be used for both types of media. Both types of media may have different implications for a particular technique. Some techniques suit better to stored media and others to live media.

Information used for synchronization technique: The information included in the MU for the synchronization purpose can be different like timestamp, sequence number. Some techniques used either sequence number or timestamp, while the other may use both.

B. *Introduction of Techniques*

Here, we define the techniques shortly and then we categorize them according to the criteria said above. Most of the time these techniques are naïve and self-explanatory, so we decided to include only the short description of technique. The summary of all these techniques can be found in the previous surveys [24, 28, and 29].

Attachment of synchronization information to MU: In this technique the synchronization information is attached with MUs. Timestamps, sequence numbers are the example of the timing information.

Buffering MU: On reception, the receiver stores MUs, to compensate for network jitter. It then presents MUs according to synchronization information attached to MU.

Transmission of MUs according to synchronization information: The MUs are transmitted according to the synchronization information attached with them. This information can be a timestamp.

Decrease the number of media stream transmission: When it is difficult for receiver to achieve synchronization, the source can temporarily stop the transmission of one of the stream. It will restart the transmission of the paused stream when the receiver is synchronized.

Deadline based transmission: The source schedules the transmission of MUs to meet the associated deadline requirements. The output deadline and the delay bounds associated to each MU must be known for this technique.

Interleaving of MUs: Source interleaves the MUs of multiple streams to make a single stream. This can degrade the intra-media quality of the stream(s)

Preventive skipping/pausing: The destination skips/discards or pause/repeat the play out of MUs depending upon the state of the buffer. It can be discarding of one from every two MUs (when the buffer occupancy exceeds the threshold) or play out every MU twice (when the buffer occupancy decreases the threshold).

Change buffering time with network delay estimation: By estimating the network delay the destination alters the buffering time of the MU. If the delay is increased to avoid buffer underflow, the buffering time of the MU can be decreased and vice versa.

Adjustment of transmission time: Upon reception of MUs, the receiver sends feedback information to the source for changing the transmission timing. The source then change the transmission period.

Reactive skipping/pausing of MU: If the play out time of the current MU is late, the receiver can skip (drop) the already received succeeding MUs. Similarly receiver can pause (play out again) the play out of the previous MU until next MU is available for play out.

Shortening/extending of play out duration: To gradually recover from asynchrony, instead of abrupt change in play out, destination can shorten/extend the play out time of MU.

Virtual time contraction/expansion: If the receiver is using the virtual time for the play out of MU instead of actual time, the MU should be played out when virtual time equals the target play out time of MU. This technique of contraction/expansion of virtual time is similar to "shortening/extending of play out duration", but it gains same effect through indirect way.

Master Slave switching: The role of master and slave stream can be interchanged dynamically, when the slave stream asynchrony is increased to a certain threshold.

Source skipping/pausing: The source can skip or pause the MUs according to the received feedback information from receiver. The receiver can also insert the dummy data (or resend the previous MU) instead of pausing the MU.

Advancement of transmission timing with network delay estimation: The source advances the transmission timing of MUs according to network delay estimates. In this way the source can skip the MUs. It will dynamically schedule the transmission of MU. It is similar with the "deadline-based transmission", which also schedules the transmission time statically.

Adjustment of capturing rate: Source adjusts the clock speed of the capturing device according to synchronization quality.

Adjustment of play out rate: The receiver adjusts the presentation device frequency according to the synchronization quality.

V. INTER-DESTINATION MULTIMEDIA SYNCHRONIZATION (IDMS)

In multicast media communication, apart from intra-media and inter-media synchronization, we can find another type of synchronization called group or inter-destination media synchronization (IDMS). The objective is to present the same stream at all the receivers in a group, approximately at the same time. To add to the complexity of the problem, these different receivers may be located at different geographical locations and may have different processing capabilities. These receivers may not only be of different type like smart phone and laptop computer but also may have the network connection of the different speeds. Network quizzes can be a good example of this scenario, where the objective will be to achieve the fairness among all the participants of the quiz. Solution will be required to display all the questions of the quiz to the entire participant at the same time.

The other example can be of the real time distance learning (tele-teaching), where the teacher multicasts a multimedia lesson to a number of students, who are located at different geographical areas. In this scenario, the teacher can also make comments about the live streaming of the lesson. Another similar example is of the interactive internet TV (Internet Social TV), where different groups of friends are watching a live online football match at different geographic locations. Consider the case, when these groups can chat (audio/video) to each other to comment on the game to experience of watching the match together from distinct location. It will be very important to synchronize the streams, so that they can watch the different events of the match at the same time to have the real experience of watching together. Figure 3 pictorially illustrates the scenario of inter-destination synchronization.

The level of required synchrony among the receivers depends on the application on hand. Considering the above three cited examples, to ensure the fairness among the participants of the online quiz, a hard synchronization will be required. In case of the other two examples, the required level of synchrony is softer as compared to the online quiz case.

The IDMS techniques cited in the literature fall under one of the three categories: *master/slave receiver scheme* (MSRC), *synchronization maestro scheme* (SMS) and *distributed control scheme* (DCS). The techniques presented in literature vary but the basic concept of the technique lies in one of the above. Here, we present the basic control scheme of each category. For better understanding of these three schemes, consider that M sources and N destinations are connected through a network. MUs of M different stream have been stored with timestamps in M source, and they are broadcasted to all receivers. The timestamp contained in an MU indicates its generation time. The streams fall into a master stream and slave streams. At each destination for inter-media synchronization, the slave streams are synchronized with the master stream by using inter-media synchronization mechanism.

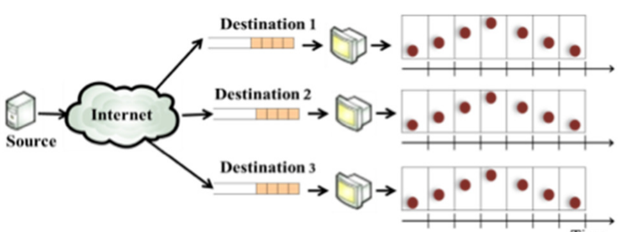

Figure 3. Inter-destination synchronization.

A. *Master/Slave Receiver Scheme (MSRS)*

In MSRS, the destinations are divided into a master destination and slave destinations. The master destination will be in control and will calculate the presentation timing of the MUs independently according to its own state of the received stream data. The slave destinations should present MUs at the same timing as the master destination. In practice multiple streams will be received at each destination and one of these streams will act as master stream for the purpose of inter-media synchronization at each destination. MSRS achieves group synchronization by adjusting the presentation time of the MUs of master stream at the slave destinations to that of the master destination.

In order to synchronize the slave destinations with the master destination, the master destination sends control packets to the slave destinations. In the beginning, the master destination multicasts a control packet including presentation time of its first MU of master stream to all slave destinations. This is called initial presentation adjustment. For the continuous synchronization among receivers the master periodically multicast control-packets whenever the target presentation time of the master destination is modified. The master notifies all the slaves about the modification by multicasting a control packet which contains the amount of time which is modified and the sequence number of the MU for which the target presentation time has been changed. Figure 4 presents the different type of message exchanges in the basic MSRS.

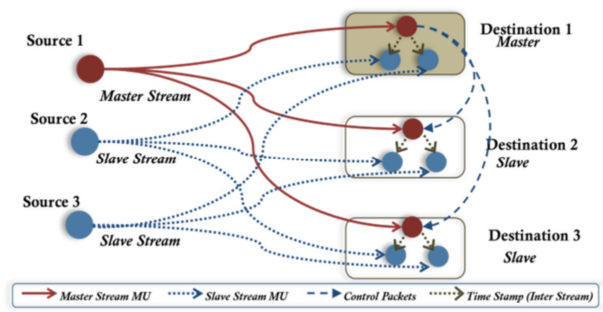

Figure 4. Master/Slave Receiver Scheme (MSRS).

This technique was initially presented in [21], and then presented in [22] by extending the RTP/RTCP messages for containing the synchronization information. The benefit of this technique is its simplicity and decreased amount of information exchange as control packet to support IDMS. Only the master destination will multicast the control packets occasionally when its target presentation time is modified or it will periodically multicast the control packets to accommodate the newly joined slave destination. Another factor which can influence the performance of the scheme is the selection of the master destination. If the slowest destination is selected as master, it can cause buffer overflow on fast slave destination, which will result as high packet drops at faster slave destination. On the other hand, if the faster destination is selected as the master destination, it can

 ISBN: 978-1-61208-195-3

cause the buffer underflow in the slower slave destination, which can result as the poor presentation quality at slow destinations. In [32], all the possible options with pros and cons are discussed for the master selection in this scheme. One issue with this technique is the associated degree of unfairness with the slave destinations. The other problem is that the master can act as bottleneck in the system.

B. Synchronization Manager Scheme (SMS)

SMS does not classify destinations into a master and slaves, therefore, all the destinations can be handled fairly. It involves a synchronization manager in order to synchronize the master stream among all destinations. The role of synchronization manager can be performed by one of the source or receiver. Each destination estimates the network delay and uses the estimates to determine the local presentation time of the MU. Each destination then sends this estimated presentation time of MU to the synchronization manager. The manager gathers the estimates from the destinations, and it adjusts the presentation timing among the destinations by multicasting control packets to destinations. SMS assumes that clock speed at the sources and destinations is the same, and that the current local times are also the same (i.e., globally synchronized clocks). The basic scheme is illustrated pictorially in Figure 5.

The SMS was initially presented in [23]. RTCP based schemes which follow the same basic principle were presented in [24]. The advantage of this scheme over MSRC is its fairness to the destinations, as the feedback information of all the destinations is accounted for determining the presentation time of the MU. But this fairness will cost more communication overhead among the destination and the synchronization manager. Like the MSRC, this scheme is also a centralized solution, so it can face the bottleneck problem.

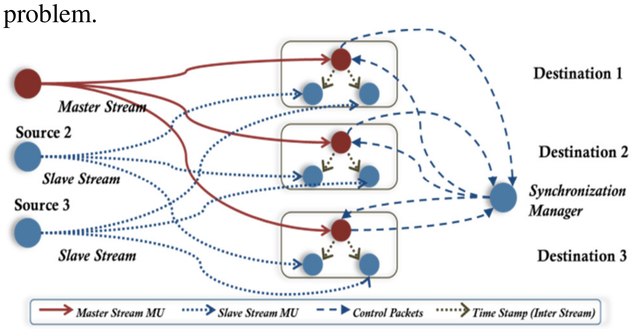

Figure 5. Synchronization Manager Scheme (SMS).

C. Distributed Control Scheme (DCS)

Unlike MSRC and SMS technique, DCS neither classifies the destination into master and slave, nor has a specific synchronization manager. In this technique, every destination estimates the network delay and then determines the presentation timing of the MU. It then sends (multicast) this presentation time to all destinations. Every destination will then have the entire view of the estimated time of MU. Each destination has the flexibility to decide the reference play out time among the timing of all the destinations. Figure 6 illustrates the pictorial representation of the scheme. This scheme was presented in [25-27].

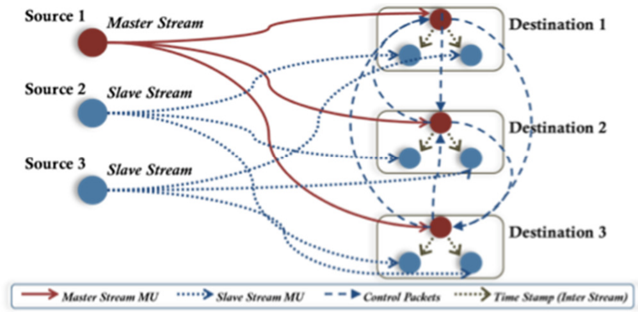

Figure 6. Distributed Control Scheme (DCS).

This scheme gives higher flexibility to each destination to decide the presentation time of MU. For example, it is possible that by selecting the presentation time of other destination, it can achieve higher IDMS quality but it may cause the inter-media or intra-media synchronization degradation. In this case, the destination has the flexibility to choose between the types of synchronization depending upon the nature of application on hand. If the application on hand demands the higher inter-media or intra-media synchronization and can sacrifice on the IDMS synchronization to certain limit, then destination can selects its own determined presentation time and vice versa. DCS is distributed scheme by nature and will not suffer from bottleneck problem. If one or more destinations leave the system, it will not disturb the overall scheme. This greater flexibility and the distributed nature of DCS make it complex in terms of processing, as before deciding presentation time of MU the destination have to do more calculations and comparisons. It has higher message complexity, as every destination will multicast the estimated presentation time.

VI. Conclusion and Discussion

The volume of research in distributed multimedia synchronization has increased significantly over the last decade. In this paper, we presented the three main types of synchronization, which are further categorized according to characteristics specific to each type. The issue of the intra-media synchronization is considered solved and no further research has been carried out for the last decade. The solutions of inter-media synchronization are challenging to compare qualitatively, since they are application specific and were evaluated subjectively. We included only initial research of group synchronization techniques, despite more solutions on these techniques have been developed lately. Although, there have been some research in inter-destination synchronization, more work is still needed to address its problems.

To the best of our knowledge, this survey is the first attempt that classifies the three main solutions at once. We hope that it will serve as a valuable asset for the research

 ISBN: 978-1-61208-195-3

community to comprehend the vast literature in the distributed multimedia synchronization.

VII. REFERENCES

[1] I. Kouvelas, V. Hardman, and A. Watson, Lip synchronisation for use over the Internet: analysis and implementation, Global Telecommunications Conference (GLOBECOM 96), Communications: The Key to Global Prosperity, 1996, vol. 2, pp. 893-898.

[2] E. Cronin, B. Filstrup, S. Jamin, and A.R. Kurc, An efficient synchronization mechanism for mirrored game architectures, Multimedia Tools Applications. vol. 23 (l) , 2004, pp. 7-30.

[3] C. Diot and L. Gautier, A distributed architecture for multiplayer interactive applications on the Internet, IEEE Network vol. 13 (4), 1999, pp. 6-15.

[4] C.M. Huang, C. Wang, and J.M. Hsu, Formal modeling and design of multimedia synchronization for interactive multimedia presentations in distributed environments, International Conference on Consumer Electronics, (ICCE 1998), Digest of Technical Papers, June 1998, pp. 458-459.

[5] Y. Ishibashi, S. Tasaka, and H. Miyamoto, Joint synchronization between stored media with interactive control and live media in multicast communications, IEICE Trans. On Commun. vol. E85-B (4), 2002, pp. 812–822.

[6] C.M. Huang, C. Wang, and C.H. Lin, Interactive multimedia synchronization in the distributed environment using the formal approach, IEEE Proc. Soft. 147 (4), 2000, pp. 131-146.

[7] N. Shivakumar, C.J. Sreenan, B. Narendran, and P. Agrawal, The concord algorithm for synchronization of networked multimedia streams, in international Conference on Multimedia Computing and Systems, May 1995, pp. 31-40.

[8] C.J. Sreenan, J.C. Chen, P. Agrawal, and B. Naendran, Delay reduction techniques for playout buffering, IEEE Transactions on Multimedia, vol. 2, no. 2, June 2000, pp. 88-100.

[9] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, Adaptive playout mechanisms for packetized audio applications in wide-area networks, in Proceedings of the Conference on Computer Communications (IEEE Infocom), Toronto, Canada, June 1994, pp. 680–688.

[10] V. Jacobson, Congestion avoidance and control, in SIGCOMM Symposium on Communications Architectures and Protocols. ACM, Aug. 1988, pp. 314-329.

[11] J.C. Bolot, End-to-end packet delay and loss behavior in the Internet, in ACM Computer Communication Review, vol. 23 (4), 1993, pp. 289–298.

[12] S.B. Moon, J. Kurose, and D. Towsley, Packet audio playout delay adjustment: performance bounds and algorithms, ACM/ Springer Multimedia Systems, vol. 5 (1), pp. 17–28, 1998.

[13] M. Roccetti, V. Ghini, G. Pau, P. Salomoni, and M.E. Bonfigli, Design and experimental evaluation of an adaptive playout delay control mechanism for packetized audio for use over the internet, Multimedia Tools and Applications, vol. 14, (1), 2001, pp. 23-53.

[14] F.A. Cuevas, M. Bertran, F. Oller, and J.M. Selga, Voice synchronization in packet switching networks, IEEE Network, vol. 7 (5), Sept. 1993, pp. 20–25,.

[15] N. Laoutaris and I. Stavrakakis, An analytical design of optimal playout schedulers for packet video receivers, Computer Communications journal, vol. 26 (4), 2003, pp. 294-203.

[16] D.L. Stone and K. Jeffay, An empirical study of a jitter management scheme for video teleconferencing, Multimedia Systems, vol. 2 (2), 1995, pp. 267-279.

[17] K. Rothermel and T. Helbig, An adaptive stream synchronization protocol, in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Durham, New Hampshire, Apr. 1995, Lecture Notes in Computer Science, pp. 189–202, Springer.

[18] E. Biersack, W. Geyer, and C. Bernhardt, Intra and interstream synchronisation for stored multimedia streams, in ICMCS (IEEE Multimedia Conference), Japan, 1996, pp. 372-381.

[19] M.C. Yuang, S.T. Liang, Y.G. Chen, and C.L. Shen, Dynamic video playout smoothing method for multimedia applications, in Proceedings of the IEEE International Conference on Communications (IEEE ICC), Texas, June 1996, pp. 1365-1369.

[20] M.C. Yuang, P.L. Tien, and S.T. Liang, Intelligent video smoother for multimedia communications, IEEE Journal on Selected Areas in Communications, vol. 15 (2), pp. 136–146, Feb. 1997.

[21] Y. Ishibashi, A. Tsuji, and S. Tasaka, A group synchronization mechanism for stored media in multicast communications, in: Proceed- ings of the Sixth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, Kobe, Japan, April 1997, pp. 692–700

[22] F. Boronat, J.C. Guerri, and J. Lloret, An RTP/RTCP based approach for multimedia group and inter-stream synchronization, Multimedia Tools and Applications Journal, June 2008, vol. 40 (2), 285-319.

[23] Y. Ishibashi and S. Tasaka, A group synchronization mechanism for live media in multicast communications, in: Global Telecommunications Conference, 1997 (IEEE GLOBECOM' 97), November 1997, pp. 746–752.

[24] F. Boronat, J. Lloret, and M. García, Multimedia group and interstream synchronization techniques: A comparative study. Inf. Systems, vol. 34 (1), March 2009, pp. 108-131.

[25] C. Diot and L. Gautier, A distributed architecture for multiplayer interactive applications on the Internet, IEEE Network vol. 13 (4), 1999, pp. 6-15.

[26] Y. Ishibashi and S. Tasaka, A distributed control scheme for causality and media synchronization in networked multimedia games, in: Proceedings of the 11th International Conference on Computer Communications and Networks, Miami, USA, Oct. 2002, pp. 144-149.

[27] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg, Local-lag and timewarp: providing consistency for replicated continuous app., IEEE Trans. Multimedia vol. 6 (l), 2004, pp. 47-57.

[28] Y. Ishibashi and S. Tasaka, A comparative survey of synchronization algorithms for continuous media in network environments, in: Proceedings of the 25th IEEE Conference on Local Computer Networks, Tampa, FL, USA, November 2000, pp. 337–348.

[29] H. Liu and M.E. Zarki, A synchronization control scheme for real-time streaming multimedia applications, in: Proceedings of the 13th Packet Video Workshop, Nantes, France, April 2003.

[30] L. Ehley, B. Furht, and M. Ilyas, Evaluation of multimedia synchronization techniques, in: Proceedings of the International Conference Multimedia Computing and Systems, (ICMCS 94), Boston, MA, USA, May 1994, pp. 514–519.

[31] M.J.P. Luque and T.D.C. Little, A temporal reference framework for multimedia synchronization, IEEE J. Sel. Areas Communications, vol. 14 (1), 1996, pp. 36-51.

[32] F. Boronat , M. Montagud, and V. Vidal, Master selection policies for inter-destination multimedia synchronization in distributed applications, in Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on. pp. 269 – 277

[33] D. Köhler and H. Müller, (Ed.) Multimedia playout synchronization using buffer level control, Multimedia: Advanced Teleservices and High-Speed Communication Architectures, Springer Berlin / Heidelberg, 1994, vol. 868, pp. 167-180.