Zixia Huang, Klara Nahrstedt, Ralf Steinmetz: *Evolution of temporal multimedia synchronization principles: A historical viewpoint.* In: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) - Special Sections on the 20th Anniversary of ACM International Conference on Multimedia, Best Papers of ACM Multimedia 2012, vol. 9, no. 34, p. 23, October 2013.

Evolution of Temporal Multimedia Synchronization Principles: A Historical Viewpoint

ZIXIA HUANG and KLARA NAHRSTEDT, University of Illinois at Urbana-Champaign
RALF STEINMETZ, Technische Universität Darmstadt

The evolution of multimedia applications has drastically changed human life and behaviors. New communication technologies lead to new requirements for multimedia synchronization. This article presents a historical view of temporal synchronization studies focusing on continuous multimedia. We demonstrate how the development of multimedia systems has created new challenges for synchronization technologies. We conclude with a new application-dependent, multilocation, multirequirement synchronization framework to address these new challenges.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*; H.5.1 [**Information Interfaces and Presentations**]: Multimedia Information Systems—*Video*

General Terms: Theory, Design, Performance

Additional Key Words and Phrases: Multimedia synchronization, survey

ACM Reference Format:

Huang, Z., Nahrstedt, K., and Steinmetz, R. 2013. Evolution of temporal multimedia synchronization principles: A historical viewpoint. ACM Trans. Multimedia Comput. Commun. Appl. 9, 1s, Article 34 (October 2013), 23 pages.
DOI: http://dx.doi.org/10.1145/2490821

1. INTRODUCTION

The past century has witnessed generations of multimedia applications, including transitions from analog modulation to digital media, single-audio, single-video playback to multimodal multichannel presentation, and two-party bidirectional communication to large-scale multiparty sharing using the Internet. The scalability and diversity of this evolution have brought about inherent complexity of time dependencies among media data, called *multimedia synchronization*, which must be preserved during computation, distribution, and presentation based on their original time attributes. For example, a motion picture and an audio sample which are captured by the camera and microphone at the same time must be presented at the corresponding output devices synchronously.

Synchronization is important in both continuous and discrete multimedia. Continuous multimedia is characterized by sequences of time-correlated media packets, which are generated by different sensors over time. Video, audio, and haptic data are examples of continuous multimedia. On the contrary, discrete multimedia constitutes the set of static media data (e.g., single images and text) or standalone

Authors' email: Z. Huang (corresponding author), zhuang21@illinois.edu; K. Nahrstedt, klara@illinois.edu; R. Steinmetz: ralf.steinmetz@kom.tu-darmstadt.de.

DOI: http://dx.doi.org/10.1145/2490821

media events (e.g., image pop-up or text animation). Synchronization of discrete multimedia may come with a coarse granularity where only the temporal order needs to be preserved. Hence, it is also called *event synchronization*. There have been numerous synchronization papers for both continuous and discrete multimedia [Boronat et al. 2009; Cronin et al. 2004; Buchanan and Zellweger 2005]. Due to space limitations, we only investigate the continuous case.

The configuration of a continuous multimedia application can be represented in multiple forms of media components (Section 2), where each component requires different temporal synchronization and triggers diverse user interests. However, the time dependencies of these media components when they are captured by the media sensors may lose track in multiple locations during media computation and distribution, due to variations in computation demands and transmission overhead (over the Internet and transport protocols). A synchronization error in one location can be propagated to future locations. In addition, a single multimedia platform may serve multiple application functionalities, so users can exhibit interests in different synchronization requirements. A two-lens stereo camera system with internal mono microphone is a good example. If it is used for 2D video conferencing, people are only interested in synchronization between the audio and one of its lens. But for 3D depth computation, synchronization between the two lens is more important. As next-generation multimedia applications are growing more complex in terms of hardware configurations, more diverse in terms of application functionalities, and more expensive in terms of consumptions of computation and network resources, preserving time correlations of media data in each application location is difficult. A systematic framework is needed to integrate application-dependent, multilocation, multirequirement synchronization problems in order to achieve their final in-sync presentation at the media outputs. We will show that such a framework is unfortunately missing in existing systems.

This article presents a historical view of synchronization studies for continuous multimedia over the past 30 years. Based on synchronization formulations (Section 2), we demonstrate how the development of multimedia systems has created new challenges for synchronization technologies (Section 3). We conclude with a new multidimensional synchronization framework to address these challenges (Section 4).

Surveys in Boronat et al. [2009] and Ishibashi and Tasaka [2000] have considered existing continuous multimedia synchronization results extensively, but mainly focused on comparing and evaluating the functionalities and methodologies of control algorithms. While we will use both surveys as a starting point, we will evaluate multimedia synchronization advancements from a completely different point of view. We clearly convey multimedia technological backgrounds and their historical roles in synchronization modeling, protocols, and human perceptual evaluation. Furthermore, we make an argument, that there is an urgent need for the research community to further evolve and advance existing synchronization practices and standards in the setting of next-generation multimedia applications. These synchronization complexities arise from the inclusion of more advanced computing and communication technologies.

2. SYNCHRONIZATION FORMULATION

Before the discussion of existing literature, we formulate the term *synchronization*. We present the mathematical model to facilitate our description in this article.

2.1 Continuous Multimedia Data Model

The architecture of a continuous multimedia data model can be described in a hierarchical fashion.

—*Session*. A session describes the multimedia communications between two or more sites for a shared collaboration. In this article, we use $\{n^1, \ldots, n^N\}$ to denote N sites within the same session.

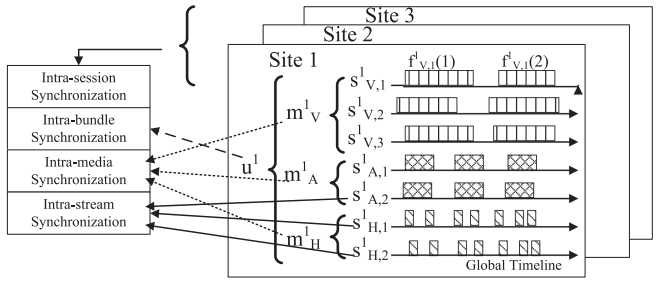

Fig. 1. Four layers of synchronization relations. $f^x_{i,j}(k)$ denotes the frame k in stream $s^x_{i,j}$; $s^x_{i,j}$ denotes the jth sensory stream in media modality i = 'V', 'A', and 'H'; and m^x_i denotes the media modality in bundle u^x at site n^x ($x = 1, 2, 3$).

—*Bundle*. A bundle is a set of time-correlated media data produced by heterogeneous sensors at the same site (i.e., the sender site). We denote the bundle of site n^x as u^x.

—*Media Modality*. To provide users with full-body immersive interactions, each site may incorporate multiple sensors with different modalities: 2D/3D video, audio, haptics, etc. By letting m^x_i be the ith media modality of site n^x, the media bundle u^x can be represented as $u^x = \{m^x_1, m^x_2, \ldots\}$. For example, we can use $i = 1$ or 'V' to represent video modality, $i = 2$ or 'A' for audio modality, $i = 3$ or 'H' for haptic modality, and etc.

—*Sensory Stream*. To preserve directionality and spatiality of the physical room environment, multiple media sensors of the same modality (e.g., microphone array or multi-camera array) can capture an object at the same time, but from different angels. Each sensor produces a sensory stream $s^x_{i,j}$ (j is the stream index), that is, $m^x_i = \{s^x_{i,1}, s^x_{i,2}, \ldots\}$. For instance, $m^x_{\mathrm{A}} = \{s^x_{\mathrm{A},1}, s^x_{\mathrm{A},2}, s^x_{\mathrm{A},3}\}$ represents the audio modality at site n^x with three audio streams captured by a microphone array. In case of multiple sensor arrays of the same media modality at a sender site, we simplify the problem by merging them into a single array.

—*Media Frame*. A sensory stream constitutes a sequence of media frames (i.e., motion images and audio samples), captured by the same sensor over time. We denote the kth media frame produced by $s^x_{i,j}$ as $f^x_{i,j}(k)$. Hence, $s^x_{i,j} = \{f^x_{i,j}(1), f^x_{i,j}(2), \ldots\}$. For example, $s^x_{\mathrm{V},2} = \{f^x_{\mathrm{V},2}(1), f^x_{\mathrm{V},2}(2), \ldots, f^x_{\mathrm{V},2}(k)\}$ represents the second video stream at site n^x with k media frames.

2.2 Layers of Synchronization Requirements

Because of the hierarchical multisite multisensory nature of multimedia data, four layers of synchronization relations are required, where each *synchronization layer* is depicted in Figure 1.

—*Intra-stream synchronization* prescribes the synchronous presentation of media frames within each sensory stream at the receivers, according to their original captured timeline at the multimedia sensors. A synchronization error in this layer can cause temporal media distortion (e.g., image jerkiness or audio pitch).

—*Intra-media synchronization* represents the synchronization of sensory streams from multiple media devices of the same media modality within a media bundle. A synchronization skew in this layer can violate spatial correlations during media presentation (e.g., a visual mismatch between two multiview images).

—*Intra-bundle synchronization* prescribes the synchronization of multiple media modalities within a bundle. This layer evaluates the time consistency across different media modalities. Audiovisual lip synchronization is a frequently studied example of intra-bundle synchronization.

—*Intra-session synchronization* represents either *inter-receiver* or *inter-sender* synchronization within a multimedia session. The inter-receiver synchronization, also named *group synchronization*, has been extensively studied by the community [Blakowski and Steinmetz 1996; Bulterman 1993]. It describes the synchronization of media bundles from the same sender site (or a media server) to multiple receivers. An out-of-sync presentation can cause unfairness when multiple people at different receiver sites get a timing privilege to conduct an activity. The inter-sender synchronization, a new requirement imposed by interactive and immersive activities, represents the in-sync presentation of media bundles from multiple senders at the same receiver. A synchronization error may lead to the confusion of the receiver user when she is watching the senders conducting a highly collaborative activity.

2.3 Definition of Synchronization Skews

A *synchronization skew* in continuous multimedia is defined as the delay difference of two time-correlated *media objects* (media frame, sensory stream, media modality, or participating site), traveling from the media sources to the current location. One of the objects is usually the *synchronization reference*, that is, the (most important) media object that other objects need to be synchronized against. Because of the multilayer synchronization hierarchy, a media object can be represented in multiple forms, meaning that the synchronization references must change accordingly at different layers. Thus, it is not possible to use a single skew to describe the whole multimedia session. Here, we adopt a more reasonable approach by defining multiple skews for different layers respectively.

Intra-Stream Synchronization Skew. The skew within a sensory stream $s^x_{i,j}$ is evaluated by computing the delay difference of a media frame $f^x_{i,j}(k)$ with respect to the *reference frame* $f^x_{i,j}(*)$. We denote $D(f^x_{i,j}(k), n^y)$ as the experienced latency of $f^x_{i,j}(k)$ from its captured time, when it is delivered to the receiver site n^y. Thus, the skew is defined as

$$\forall x, y, i, j,\ \ f^x_{i,j}(k) \in s^x_{i,j}:\ \ \Delta D\big(f^x_{i,j}(k), n^y\big) = D\big(f^x_{i,j}(k), n^y\big) - D\big(f^x_{i,j}(*), n^y\big). \tag{1}$$

Intra-Media Synchronization Skew. We denote $D(s^x_{i,j}, n^y)$ as the experienced latency of $s^x_{i,j}$ when delivered to n^y. Note that due to potential computation and Internet *jitter* (i.e., variations of latency)[1] across media frames within a sensory stream, we use the latency of the reference frame to represent that of the stream, that is, $D(s^x_{i,j}, n^y) = D(f^x_{i,j}(*), n^y)$. Hence, the intra-media synchronization skew $\Delta D(s^x_{i,j}, n^y)$ with respect to the *reference stream* $s^x_{i,*}$ is defined as

$$\forall x, y, i, j,\ s^x_{i,j} \in m^x_i:\ \ \Delta D\big(s^x_{i,j}, n^y\big) = D\big(s^x_{i,j}, n^y\big) - D\big(s^x_{i,*}, n^y\big). \tag{2}$$

Intra-Bundle Synchronization Skew. Because sensory streams within a media modality can experience heterogeneous latencies, we prescribe that the latency of a media modality is defined as the latency of the intra-media synchronization reference (i.e., reference stream) within this modality, in order to best match human perceptual interests, that is, $D(m^x_i, n^y) = D(s^x_{i,*}, n^y)$. Thus, the intra-bundle synchronization skew of m^x_i with respect to the *reference modality* m^x_* is defined as

$$\forall x, y, i,\ m^x_i \in u^x:\ \ \Delta D\big(m^x_i, n^y\big) = D\big(m^x_i, n^y\big) - D\big(m^x_*, n^y\big). \tag{3}$$

Note that previous studies [Blakowski and Steinmetz 1996; Little and Ghafoor 1991; Meyer et al. 1994] usually combine intra-media and intra-bundle synchronization requirements into a single layer called *inter-stream synchronization* (i.e., synchronization of multiple multimodal streams within a media bundle). The *inter-stream synchronization skew* in these studies is defined regardless of media

[1]In this article, Internet jitter describes the variations of latency caused by Internet propagation and transport-layer protocols.

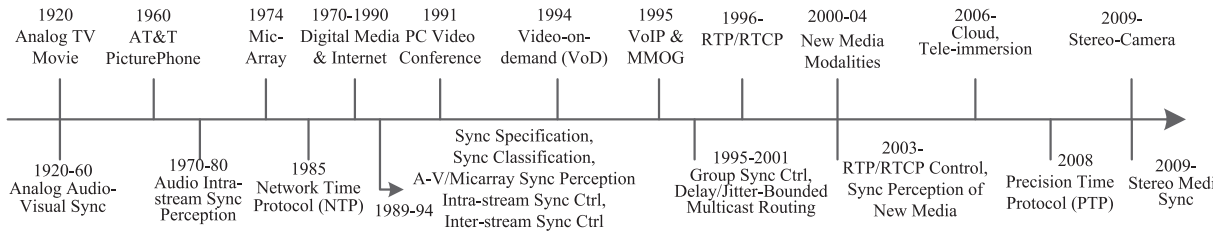

Fig. 2. Advancement timeline of multimedia and synchronization technologies. RTP: real-time protocol; RTCP: real-time control protocol; MMOG: massively multiplayer online game.

modalities. In other words, it uses a single reference stream (denoted as s_*^x) for all other streams of different media modalities within the same bundle. The inter-stream skew is formulated as

$$\forall x, y, i, j: \quad \Delta D\big(s_{i,j}^x, n^y\big) = D\big(s_{i,j}^x, n^y\big) - D\big(s_*^x, n^y\big). \tag{4}$$

There is no skew constraint between two nonreference streams in inter-stream synchronization. For example, we are unable to bound the skew between two video streams (from a multi-camera system) which use the same audio stream as the reference. This is why we propose intra-media and intra-bundle synchronization layers separately. The issue has been neglected even in the work finished within the past 5–6 years [Boronat et al. 2009], when camera/microphone arrays were deployed, mainly because of the community's stereotyped view of synchronizing a single video and a single audio stream in the most common on-demand or conferencing multimedia systems. In next-generation multimedia systems with increasing diversity in multimedia sensors, intra-media and intra-bundle synchronization errors can introduce very different impacts on human perception (Section 3). This difference cannot be captured by traditional inter-stream synchronization.

Intra-Session Synchronization Skew. Similar to the intra-bundle layer, we prescribe that the latency of a bundle is defined as the latency of the intra-bundle synchronization reference within the bundle, that is, $D(u^x, n^y) = D(m_*^x, n^y)$. Given the *reference site* n^*, the inter-sender synchronization skew as to a receiver site n^{y_0} is

$$\forall x: \quad \Delta D\big(u^x, n^{y_0}\big) = D\big(u^x, n^{y_0}\big) - D\big(u^{n^*}, n^{y_0}\big). \tag{5}$$

Accordingly, the inter-receiver synchronization (group synchronization) skew as to a sender site n^{x_0} is

$$\forall y: \quad \Delta D\big(u^{x_0}, n^y\big) = D\big(u^{x_0}, n^y\big) - D\big(u^{x_0}, n^*\big). \tag{6}$$

In continuous multimedia, the synchronization skews are usually evaluated at specific time points, called *synchronization points*. Multiple control approaches utilize the concept of synchronization points to perform and evaluate intra-stream and inter-stream synchronization [Steinmetz 1990].

3. A HISTORICAL VIEW OF SYNCHRONIZATION STUDIES

Multimedia technologies have experienced multiple generations of evolution, with different synchronization requirements in each generation. In our study, we divide them into four stages based on the temporal order. In each stage, we discuss the role of multimedia technologies in advancing synchronization research. Figure 2 shows a timeline of multimedia and synchronization development.

3.1 Years of Birth: On and before 1988

The rise of electronic technologies gave birth to a number of analog and digital multimedia applications before 1988. The rapid deployment of digital computing and communication technologies with

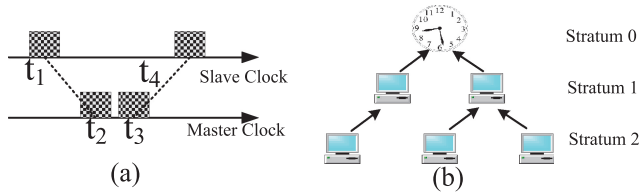

Fig. 3. (a) NTP clock offset computation; (b) NTP multi-stratum hierarchy.

unreliable characteristics, such as PCs and the Internet, brought people's attention to the problem of digital multimedia synchronization. But in those early years, the synchronization concept mainly stayed at the fidelity or intelligibility of multimedia signals.

Historical Background. Back to the 1920s, the broadcast analog TV service and sound film technology developed rapidly. Later in the 1960s, AT&T Bell Labs demonstrated its own analog picturephone which supported a video frame rate of up to 30 fps [BellLabs 1969]. In 1974, the microphone array (or microphone antenna) technique was invented by Billingsley [Michel 2006]. Analog multimedia synchronization between audio and video was an issue in those early years. In sound films, synchronization was achieved by synchronous motors and marked synchronization points in both film and recorded sound [Barrios 1995]. In broadcast TV service, it was solved by taking the analog audio and video signals, multiplexing them and transmitting them over a controlled communication channel [BellLabs 1969]. In addition, operations of these application functionalities, and the quality of analog audio and video intelligibility were themselves immature. Hence, they became the priority problems to solve.

Start of Synchronization Perception Studies. It was not until the 1970s and 1980s that digital multimedia synchronization was realized as a problem. The invention of digital computers fostered the development of digital media, while the introduction of the best-effort Internet protocols brought people's attention to the concept of Internet jitter. Researches became interested in how Internet jitter affected digital media fidelity and human perception. Several preliminary results were developed to discuss the impact of jitter on intra-media synchronization of digital audio. For example, Dannenberg [Blesser 1978] offers a few references and results. Based on his work, the maximum tolerable intra-stream skew for 16-bit high-quality audio is 200 ns in one sampling period. Similar results can also be found in Stockham [1972], which recommends a maximum allowable intra-stream skew of no more than 5–10 ns.

NTP: A Clock Synchronization Protocol. In 1985, David Mills proposed the first version of the Network Time Protocol (NTP), a protocol designed for synchronizing clocks on distributed computers connected by the Internet. To synchronize one computing machine (called a *slave*) against the other (called the *master*), the NTP slave computes the round-trip delay by sending a set of UDP packets to the remote master. We assume the time that a packet leaves the slave is t_1 and arrives at the remote master is t_2 (Figure 3(a)). We also denote the time that the packet leaves the master is t_3 and returns to the slave is t_4. All times are measured based on local clocks. Hence, the clock offset between the two machines is

$$\delta = \frac{(t_2 - t_1) + (t_3 - t_4)}{2}. \tag{7}$$

Eqn. (7) implies that NTP assumes symmetrical round-trip delay. But in reality, the unequal bidirectional latency and jitter can degrade the clock synchronization accuracy. In addition, time measurement is at the application layer whose accuracy depends on the underlying operating system. In general, NTP can lead to a skew error up to the range of 10 ms on wide area network [Steinmetz and

Nahrstedt 1995]. To minimize the impact of jitter on synchronization accuracy and address the issue of computing machine scalability, NTP adopts a multi-stratum hierarchy (Figure 3(b)), where machines in a stratum layer l are synchronized to the corresponding masters in the higher stratum layer $l-1$.

NTP is important in multimedia applications, because it provides a solution for accessing the global clock across distributed machines, so that we can identify the time correlations of two media objects, which are produced or are operating at different physical systems. We will show that existing studies rely heavily on the global timing state in order to achieve multimedia synchronization.

3.2 Years of Understanding: 1989–1994

Owing to the technological advances of the Internet protocols (IP), many Internet-based digital multimedia systems emerged and were commercialized in the late 1980s and early 1990s. Multimedia synchronization became a known and important topic to the research community, and extensive research was done to understand the synchronization problem. These studies covered a broad synchronization area including classification, specification, subjective perception evaluation, and synchronization control algorithms.

Historical Background. In 1991, IBM and PictureTel introduced the first PC-based black-and-white video conferencing system [PictureTel 1991]. In 1992, teleorchestration was invented as a stream-oriented interface for continuous media presentation across multiple distributed systems [Campbell et al. 1992], while a real-time virtual multichannel acoustic environment was invented by Gardner based on microphone arrays [Gardner 1992]. The video-on-demand (VOD) service was also started under the Cambridge project, offering Internet streaming videos at a bandwidth up to 25 Mbps [Wallis 1995].

The proliferation of new Internet-based multimedia systems and the improvement of digital audio-visual fidelity encouraged the researchers to address the synchronization problem. The one-way delay variations between the (single) audio and (single) video streams in both video conferencing and on-demand video systems, and between multiple audio streams in a microphone array setup, exhibited a need for intra-bundle and intra-media synchronization. The development of teleorchestration brought people's attention to the inter-receiver/group synchronization. Multimedia synchronization studies thus became a hot topic during this period.

Synchronization Classification. To understand the heterogeneous requirements of multimedia synchronization, a classification model is needed for investigating the structure of synchronization mechanism and comparing runtime controls that guarantee in-sync presentation of multimedia data. Many classification models were proposed, with views from different aspects of the synchronization problem.

—*Little and Ghafoor Model [1991].* This classification model covers both intra- and inter-stream synchronization, in spite of random network delays. Timed discrete media objects like still images and text are also included in the model.

—*Steinmetz et al. Model [Meyer et al. 1994; Blakowski and Steinmetz 1996].* Based on the type of synchronization requirements, this model is divided into four layers: (1) *media layer*, that is, intra-stream synchronization; (2) *stream layer*, including inter-stream and inter-receiver/group synchronization; (3) *object layer*, describing synchronization of both continuous and discrete media objects; and (4) *specification layer*, prescribing applications and tools for synchronization specifications.

—*Ehley et al. Model [1994].* This model classifies the synchronization technologies based upon the synchronization locations, that is, the places where synchronization control schemes are performed. Only inter-stream synchronization is investigated in each location.

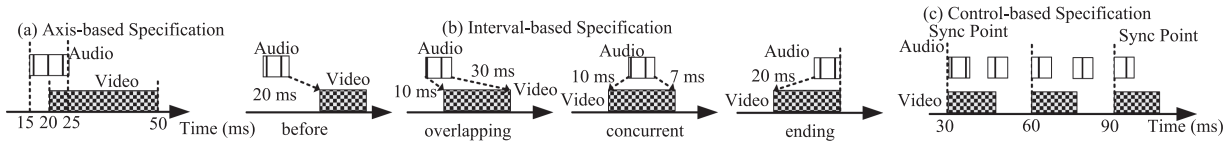

Fig. 4. Three synchronization specification models. Each box or circle represents a media frame.

As one can see, these three synchronization classification models are, in nature, either aligned with each other or mutually orthogonal.

Synchronization Specification. A further understanding of multimedia synchronization topics requires more systematic specification methods to describe synchronization problems. This promotes a number of specification models that can generally be grouped into three categories (Figure 4) [Blakowski and Steinmetz 1996]. A comparison table is presented in Table I of the Online Appendix.

—*Axis-Based Specification [Hodges et al. 1989].* This specification model aligns multimedia objects in either a real or virtual global timeline axis, based on the start and finish times of each object. The accessibility of a real timeline axis is owed largely by the wide deployment of NTP, and a virtual axis can be obtained by referencing the clock skews across distributed machines. The duration of each media object must be described in the specification. For example, in Figure 4(a), we specify that a video frame is presented between the 20th and the 50th ms, while another audio frame is played between the 15th to the 25th ms. The axis-based specification offers a direct view of time relations and synchronization skews of media objects in a global setting, thus facilitating its implementation in real multimedia systems. Media objects in the specification can be added and removed freely due to their mutual independence. However, media data with unknown start and finish times cannot be integrated into the axis-based method, and take advantage of benefits that the specification provides.

—*Interval-Based Specification [Wahl and Rothermel 1994].* This specification model presents the logical temporal relations between two media objects. The exact start and finish times of each media object is unspecified. A total of 29 interval relations are defined, indicating whether a media object is before, after, or overlapping with another object. Figure 4(b) shows an example of four relations with different delay parameter inputs. Similar to the axis-based approach, the interval-based specification is easy to understand, and adding/removing mutually dependent media objects is relatively simple. But because it does not require a knowledge of the duration of each media object, the real specification implementation can be difficult.

—*Control-Based Specification [Steinmetz 1990].* In this model, multimedia data are synchronized over a set of connected synchronization points, based on which a system detects synchronization errors and realigns multimedia data. Oftentimes, these time points are placed periodically in order to allow consistent and manageable media resynchronization. Figure 4(c) shows a sequence of synchronization points every 30 ms. The major advantage of this method is that it can explicitly tell users when the synchronization should be performed. It also allows the integration of new media objects without major efforts. Its drawbacks are that we require an additional mechanism to specify the synchronization skews, and that a timer is required to realize the periodic synchronization points.

Synchronization Perception. As people noticed audiovisual synchronization skews in VOD and conferencing systems over the Internet, researchers became interested in understanding how large an audiovisual skew can be noticed by humans. A subjective study conducted by Steinmetz and Engler [1993] recommends that a lip skew less than 80 ms is not detectable, and a skew greater than 160 ms is

unsatisfactory. In addition, it also concludes that people are less tolerant to a skew when the video signal is behind the audio than to a skew when the audio is behind. The findings can be explained by the fact that the speed of light is much faster than the speed of sound, so people are accustomed to a late audio signal over growing distances.

In the same year, skews between multiple acoustic streams within a microphone array were also studied by Dannerberg and Stern [1993]. The authors claim that a skew of 17 ms between stereo audio signals is perceivable, and that a maximum skew of 11 ms is preferable.

Intra-Stream and Inter-Stream Synchronization Control. Researchers began to investigate the control framework in the early 1990s, exclusively for intra-stream and inter-stream synchronization of video conferencing or on-demand systems, owing to the rapid commercialization of these Internet-based applications. Most studies in those early years focused on synchronization of a single audio and a single video stream, where the audio stream was always selected as the reference stream in master (audio) - slave (video) synchronization prototype. Audio was chosen because human perception is more sensitive to the degradations of audio signals. A global time clock is also assumed to be available between video and audio signals.

This article groups different studies based on the location and functionality of synchronization control mechanisms. For synchronization location, we investigate control algorithms at both sender and receiver sides. In terms of functionality, we classify synchronization approaches that can either be shared universally by any media modality, or applied only to one specific modality.

(1) *Receiver-Based Synchronization* Buffering compensation is the most common approach to accommodate intra-stream jitter and to minimize the inter-stream skew. To facilitate our description, we prescribe that the sender site is n^x, and the receiver site is n^y. The network delay of a media frame $f^x_{i,j}(k)$ (within the sensory stream $s^x_{i,j}$) is $D_{\text{net}}(f^x_{i,j}(k), n^y)$, the buffering delay $D_{\text{buf}}(f^x_{i,j}(k), n^y)$, and the resulting end-to-end latency $D_e(f^x_{i,j}(k), n^y) = D_{\text{net}}(f^x_{i,j}(k), n^y) + D_{\text{buf}}(f^x_{i,j}(k), n^y)$. Hence, between two buffer status updates, we must satisfy the following two requirements.
—Intra-stream synchronization. $\forall k$, $D_e(f^x_{i,j}(k), n^y)$ must remain equal, that is, $D_e(f^x_{i,j}(k), n^y) = D_e(s^x_*, n^y)$.
—Inter-stream synchronization. $\forall\ i,\ \ j,\ |D_e(s^x_{i,j}, n^y) - D_e(s^x_*, n^y)| < \delta_s$ must satisfy, where s^x_* is the inter-stream reference, and δ_s is the *synchronization threshold* of the inter-stream skew.

When D_e is decided, the buffering delay of each media frame (D_{buf}) is decided based upon the network latency statistics (D_{net}). Computation heterogeneity is usually not considered.

The abrupt adaptation of the buffering delay during a transition period of two consecutive updates can introduce discontinuity in a media presentation. Most studies address this issue by implementing an algorithm [Anderson and Homsy 1991; Cluver and Noll 1996; Little 1993; Ravindran and Bansal 1993; Rothermel and Helbig 1995; Woo et al. 1994; Yavatkar 1992; Bailey et al. 1998] to minimize the degradations of the intra-stream synchronization quality:

Increase Buffering Latency.
—*Shared approach*. (1) Replicating past media frames; (2) interpolating media information by bidirectional data prediction based on neighboring media data.
—*Video only*. Increasing inter-frame period.
—*Audio only*. (1) Time-scale modification without pitch change (expanding the playout duration of each audio sample); (2) inserting silent packets.

Decrease Buffering Latency.
—*Shared approach*. Skipping media frames during presentation.
—*Video only*. Decreasing inter-frame period.

—*Audio only*. (1) Time-scale modification without pitch change (shrinking the playout duration of each audio sample); (2) dropping silence packets.

(2) *Sender-Based Synchronization*. Network bandwidth estimation and the resulting media data management are the two key components. The reason is that insufficient bandwidth can exert Internet congestion jitter and losses which can affect both intra-media and inter-media synchronization. Based on the estimated bandwidth [Hu and Steenkiste 2002; Ramanathan and Rangan 1993], the sender performs multiple options of the data management schemes [Qiao and Nahrstedt 1997; Ravindran and Bansal 1993; Rothermel and Helbig 1995; Woo et al. 1994; Bailey et al. 1998] which include the following.
—Reducing media sampling rate (e.g., changing audio sampling frequency from 16000 Hz to 8000 Hz, or video frame rate from 20 fps to 10 fps).
—Downgrading media encoded quality (e.g., downgrading video/audio encoded data rate).
—Skipping media data of low priority (e.g., only sending I frames of MPEG videos).
—Discarding media frames that cannot meet the receiver presentation deadline (based on feedback messages from the receiver indicating current playout buffer status).

Boronat et al. [2009] and Ishibashi and Tasaka [2000] have both summarized that the sender and receiver synchronization methods can be employed jointly, and that each method can be performed either passively in response to Internet quality changes, or actively so as to prevent potential Internet degradations.

3.3 Years of Blossoms: 1995–2001

Multimedia synchronization continued to be a hot topic due to the evolutionary change of Internet quality (i.e., increased bandwidth and reduced latency).

Historical Background. Broadband Internet became widely available in the late 1990s. This promoted the development of multiple real-time applications, for example, the world's first commercial VoIP service by VocalTec in 1995 [Tov 2005], the first 3D massively multiplayer online game (MMOG) by 3DO Company in 1995 [Damer 1998], and the Caltech-CERN project in 1997 which built a virtual room videoconferencing system connecting research centers over the world [Bunn et al. 1998]. The evolution of these multimedia applications had sparked the massive interests in realizing the inter-receiver/group synchronization, for the purpose of preserving the fairness and the time relations among receiver users.

Inter-Receiver/Group Synchronization Control. Similar to the intra-stream and inter-stream synchronization, inter-receiver synchronization control schemes can also be classified based on synchronization locations and synchronization control methodologies. To facilitate the description, we prescribe that the sender site is n^{x_0}, and the list of receiver sites is $\{n^1, n^2, \ldots\}$. We also denote the network delay between the sender n^{x_0} and any receiver n^y is $D_{\text{net}}(u^{x_0}, n^y)$ (where u^{x_0} is the media bundle sourced at n^{x_0}), the buffering delay $D_{\text{buf}}(u^{x_0}, n^y)$, and the resulting end-to-end latency $D_e(u^{x_0}, n^y) = D_{\text{net}}(u^{x_0}, n^y) + D_{\text{buf}}(u^{x_0}, n^y)$. By denoting the synchronization reference site as n^*, the synchronization goal can be formulated as

$$\forall y: \ \left|D_e\left(u^{x_0}, n^y\right) - D_e\left(u^{x_0}, n^*\right)\right| < \delta_{\text{rcv}}, \tag{8}$$

where δ_{rcv} is the synchronization threshold of the inter-receiver synchronization skew. To further simplify the problem, we assume zero Internet jitter in our discussion. Table II summarizes different group control methods in the Online Appendix.

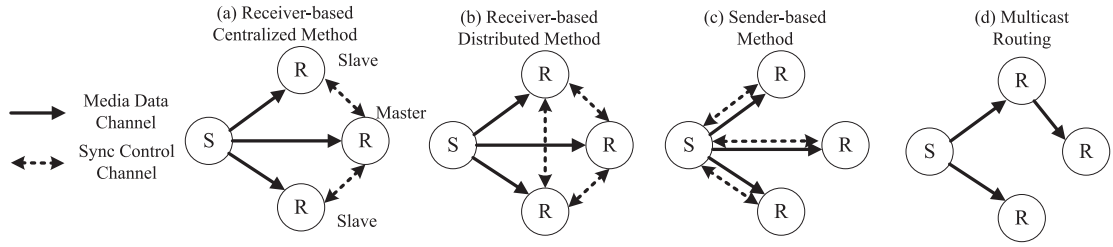

Fig. 5. Group synchronization control algorithms. S: sender site; R: receiver site.

(1) *Receiver-Based Synchronization*. One or multiple receivers need to decide the buffering delay D_{buf} without the information from the sender site. Based on synchronization methodologies, the receiver-based approaches can be further divided into two categories.

—*Centralized (master-slave) method* (Figure 5(a)). In this method, one master receiver is selected as the synchronization reference site n^*, and all other receiver sites are the slaves [Akyildiz and Yen 1996; Ishibashi et al. 1997]. Usually, n^* is chosen as the receiver with the longest D_{net} from the sender, that is, $n^* = \arg\max_y D_{\text{net}}(u^{x_0}, n^y)$. The detailed procedures are described as follows. (1) n^* first decides the one-way latency $D_e(u^{x_0}, n^*) = D_{\text{net}}(u^{x_0}, n^*)$ ($D_{\text{buf}}(u^{x_0}, n^*) = 0$). (2) All other slave receivers unicast individual $D_{\text{net}}(u^{x_0}, n^y)$ to n^*. (3) n^* decides for each n^y its target D_e by

$$\forall y: \quad D_e(u^{x_0}, n^y) = \max\left\{D_{\text{net}}\left(u^{x_0}, n^y\right), \quad D_e\left(u^{x_0}, n^*\right) - \delta_{\text{rcv}}\right\}. \tag{9}$$

n^y's buffering delay D_{buf} is also updated correspondingly. (4) n^* sends the computed D_e and D_{buf} values back to each slave. While it is simple to implement the centralized method in the real multimedia systems, serious drawbacks may hinder its efficient operation. First, the connectivity between master and slave receivers cannot be guaranteed due to poor Internet conditions and firewall blocking issues. Second, timely synchronization adaptation in response to sudden Internet quality changes is not possible, because of the bidirectional latency between master and slave sites. Third, scalability is a common problem in the centralized method, where the computation and network resources may be bottlenecked at the master receiver. Fourth, receiver sites can easily join and leave the session in some multimedia applications like MMOG. When the master site suddenly leaves without announcement, group synchronization may fail immediately.

—*Distributed method* (Figure 5(b)). In this method, each receiver site decides its own buffering delay D_{buf} in a distributed fashion [Ishibashi and Tasaka 2000], by periodically broadcasting its D_e value to each other. When a receiver (denoted as n^1) receives a message from another site (denoted as n^2), it compares the $D_e(u^{x_0}, n^2)$ value carried in the message with its own $D_e(u^{x_0}, n^1)$ value. If $D_e(u^{x_0}, n^1) \geq D_e(u^{x_0}, n^2)$, n^1 simply neglects the message. Otherwise, it assumes n^2 as the temporary synchronization reference ($D_e(u^{x_0}, n^*) = D_e(u^{x_0}, n^2)$), and follows Eqn. 9 to update its own buffer status. Compared to the centralized method, frequent message exchanges among receivers due to full-mesh communication can raise bandwidth overhead tremendously.

(2) *Sender-Based (Maestro) Synchronization* (Figure 5(c)). The receiver sites unicast individual D_{net} information to the sender, which is then responsible for deciding on the receiver buffering delay D_{buf} and the target end-to-end latency D_e of each receiver [Boronat et al. 2009]. The derivation is exactly the same as the algorithm used in the receiver-based centralized approach. The values of D_{net}, D_{buf} and D_e can be piggybacked onto the media packet header during bidirectional data transmission between the sender and receivers. The resulting message exchanges can be effectively minimized. In addition, reliability is not a problem when receiver sites are joining and leaving a session, as long as the sender is consistently sending media data to the receivers. Sender-based synchronization is,

by far, the best method to realize inter-receiver/group synchronization in real systems, due to its flexibility, reliability, and ease of implementation. However, timely synchronization adaptation is still not possible because of the round-trip latency incurred during the synchronization information exchanges.

(3) *Multicast Routing with Bounded Delay and Delay Variation* (Figure 5(d)). A third method is to control D_{net} to bound inter-receiver skews incurred over the Internet rather than introducing buffering latency to compensate for the skews. In multisite applications, the distribution of media data from a sender to each receiver may be routed through specific intermediate sites. We call it a *multicast overlay*. In designing such a topology, there can be multiple path options from the same sender to the same receiver, but via different intermediate sites. These path options may feature unequal network latencies that will lead to heterogeneous inter-receiver skews. Multicast overlay with a bounded inter-receiver skew is required by many synchronization control algorithms [Rouskas and Baldine 1997; Shi et al. 2001; Zimmermann and Liang 2008]. In general, the overlay design can be formulated as an optimization problem in the following form.
   —*Goal*. Minimize average D_{net} for all sender-receiver pairs.
   —*Synchronization constraint (optional)*. Bound the resulting delay (i.e., D_{net}) and delay variation (i.e., inter-receiver skew).
   —*Bandwidth constraint (optional)*. Bound the inbound/outbound bandwidth utilization of each site.

The preceding problem has been proven NP-hard [Zimmermann and Liang 2008]. The optimization goal can be achieved by combining the shortest bounded path options based on Dijkstra's algorithm. Synchronization and bandwidth constraints are realized by iterating over k-shortest path options between each sender and receiver sites to find the one which can bound synchronization skews and/or bandwidth utilization [Rouskas and Baldine 1997; Shi et al. 2001].

Note that in utilizing these multicast studies, one must assume that multimodal sensory streams from a sender site follow the same distribution path to the same receiver.

3.4 Years of Leaps: On and After 2002

Modern multimedia systems became more powerful in terms of the accessibility of computation and network resources, more complex in terms of both hardware and software configurations, and more versatile in terms of the functionalities that can be performed. The development of modern multimedia and networking technologies has led to many open synchronization problems that await researchers to investigate.

Historical Background. Because of increasing computation power (e.g., multicore processor and cloud) and better Internet bandwidth availability, multimedia sensors such as haptics, accelerometer, and body sensor have won wide acceptance in modern multimedia systems. These media modalities provide users with a completely new experience of synchronization perception. Their heterogeneous computation requirements also introduce very different end-to-end latencies that contribute to synchronization errors.

In parallel of these new technologies, many well-known protocols have been developed for synchronization adaptation, including the Real-time Transport Protocol (RTP) specifying the standardized packet format for delivering streaming media over the Internet, the Real-time Control Protocol (RTCP), defining the control information for RTP data [RFC 2003], and the Precision Time Protocol (PTP) [IEEE 2008]. At the transport layer, the Datagram Congestion Control Protocol (DCCP) [Kohler et al. 2006] is also proposed for real-time congestion adaptation. Compared to the Transmission Control

Protocol (TCP), DCCP reduces abrupt changes of sending rate during congestion controls, and allows a smoother transmission jitter that facilitates application-layer synchronization.

Precision Time Protocol (PTP). To address the issue of NTP's subpar synchronization accuracy, IEEE presents the 1588 standard: Precision Time Protocol (PTP) [IEEE 2008]. PTP is able to achieve a clock accuracy up to the range of submicroseconds on the local area network (LAN). PTP adopts the same multi-stratum hierarchy as in NTP, but makes three improvements for better synchronization precision. First, time measurement is at the specialized hardware close to the physical transmission medium, thus providing much better precision than NTP's application-layer measurement. Second, PTP employs a best master clock algorithm to select the synchronization master, where multiple candidate clocks are prioritized by user predefined configurations as well as clock traceability, accuracy and variance. Third, a time-interval field is introduced in PTP messages to compensate for the residence time of the network devices between the master and slave machines.

RTP/RTCP-Based Synchronization Control Implementation. RTP and RTCP have been used extensively in real-time multimedia streaming and synchronization [Boronat et al. 2008; Leroux et al. 2007]. RTP defines the packet format for media data encapsulated in an IP packet. Three fields are included in the RTP header that are directly related to synchronization: (1) payload type, indicating the media modality of the payload; (2) sequence number, representing the index of the RTP packet in a stream for intra-stream synchronization; (3) time stamp, describing the local (relative) time stamp of media frames within each stream, required to satisfy various synchronization requirements. RTP does not specify a global time status. In other words, the time correlation across different streams cannot be specified without the help of other clock synchronization algorithms or protocols (e.g., NTP or PTP).

On the other hand, RTCP provides a communication channel to send synchronization control information between sites. RTCP has two types of packets. (1) Receiver report (RR): receiver sends RR messages to the sender specifying the packet loss rate, jitter statistics, or receiver buffer status. The sender can perform synchronization adaptations (e.g., bandwidth provisioning) dynamically by referencing real-time RR feedback. (2) Sender report (SR): a sender sends SR messages periodically to all receivers. An NTP/PTP (global) timestamp field is included in the SR message to compute the one-way latency between each sender and receiver.

An IETF proposal [Brandenburg et al. 2012] discusses the use of RTCP for inter-receiver synchronization based on the sender-based (maestro) approach (Section 3.3). The proposal uses NTP to synchronize all receiver sites. It also prescribes that the receiver with the largest one-way latency is selected as the reference site. Both NTP clock sources and media clock sources are identified by the signalling of the Session Description Protocol (SDP) [Williams et al. 2013]. Here, a media clock indicates relative time among the media data. It is provided either by media contextual interface or by genlock-like reference signal [Williams et al. 2013].

Synchronization Perception of New Media. There are also a number of subjective studies investigating the perceptual impact of multimedia synchronization in modern applications.

Curcio and Lundan [2007] evaluate synchronization in a mobile terminal with a maximum image size 176x144. They show that in the mobile setting with a video frame rate below 15 fps, people are more tolerant of a synchronization error when the video spatial resolution is reduced. They also describe that a lip skew can be as large as 200–300 ms before it annoys a user, because of the degraded motion smoothness.

Fujimoto et al. [2008] evaluate the subjective quality of the skew between haptic and video data. They show that a skew below 40–80 ms is hardly perceptible, and that a skew greater than 300 ms is annoying.

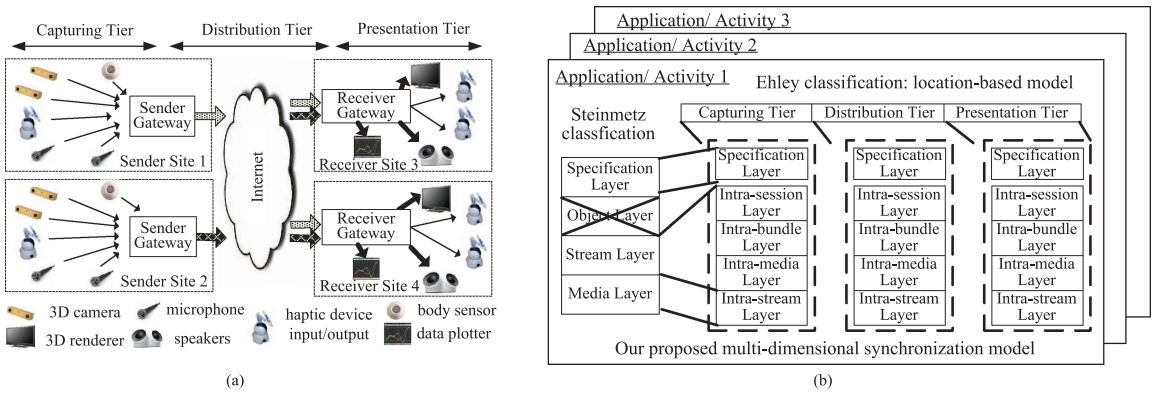

Fig. 6. (a) General architecture of TI systems; (b) multidimensional synchronization model.

Ghinea and Ademoye [2010] conduct a perceptual measurement of the impact of a synchronization error between smell sensory data (i.e., olfaction) and audiovisual content, assuming the audiovisual lip skew is zero. Their results show a synchronization threshold of 30 s when olfaction is ahead of audiovisual data, and of 20 s when olfaction is behind.

Hoshino et al. [2011] also measure the impact of an olfactory-haptic skew. They present that the annoying threshold is in the range of 1–3 s.

For (intra-media) synchronization quality of 3D stereoscopic videos, Goldmann et al. [2010] evaluate four video samples with different scenes, all at a frame rate of 25 fps and a spatial resolution of 1920x1080. The authors claim that a skew below 80 ms leads to a good 3D visual quality, while a skew larger than 200 ms annoys the visual experience.

3.5 Remarks

Several remarks can be made. First, there is no classification model that captures all synchronization requirements in multiple locations of a single multimedia system (Section 3.2). Second, the synchronization reference is usually chosen statically (e.g., the audio for inter-stream synchronization). However, new multimedia systems are not limited to traditional conferencing and on-demand applications, so audio may not be the most important media data. Third, most studies focus on the skews incurred over the Internet. None of them investigates the heterogeneity of computation demands, and its impact on synchronization. The next section presents solutions to address these issues.

4. SYNCHRONIZATION IN NEXT-GENERATION MULTIMEDIA SYSTEMS

Next-generation multimedia systems (NG-MS), like interactive 3D teleimmersive (TI) applications, provide geographically distributed users with a realistic and immersive experience. Synchronization in these new systems and applications is characterized by the following three attributes.

(1) *Demands of scale and device heterogeneity*. Multiple sensory devices with heterogeneous media modalities can be configured in a NG-MS (e.g., multiview videos or spatial audios). This requires both intra-media and intra-bundle synchronization. The immersive environment adds the demand of inter-sender synchronization to preserve seamless interaction among participants.

(2) *Multi-location synchronization controls*. A NG-MS can generally be divided into multiple locations, each of which may affect synchronization skews. Consider a TI system, as shown in Figure 6(a). At the *capturing tier*, a sender site captures time-dependent multimodal media frames and encodes them in real time. The computation heterogeneity can contribute to the skews. At the

distribution tier, the sender gateway serves as a rendezvous point that simply forwards multimodal, multistream data to multiple receivers. Synchronization skews are mainly caused by Internet jitter and the use of an overlay network for distribution. At the *presentation tier*, multimedia streams are decoded and played at the corresponding output devices. Buffering latency is introduced to compensate for the accumulated synchronization skew.

(3) *Diverse applications on a single multimedia platform*. A variety of applications can be served on a single TI platform, including media consulting, remote education, and collaborative gaming. Different media modalities and sensory streams can have varying contributions to the functionality of each application, so they will impact the synchronization perception differently at end users [Huang et al. 2010]. Because synchronization references usually represent the most important media information against which to synchronize, they must be selected depending on user activities and their specific underlying application functionalities.

Existing practices and standards, as discussed in Section 3, focus only on a single dimension of the three synchronization attributes (e.g., Ehley's model considers only the location, while Steinmetz's model considers only device heterogeneity). As TI system is growing more complex, the combined interaction of all three dimensions must be addressed. This prevents the propagation of synchronization errors to different locations, and facilitates in-sync multimedia presentation at the output devices with minimal buffering compensation (for better interactive quality). Hence, we propose a new classification model to capture their multidimensional characteristics.

4.1 A New Multidimensional Synchronization Classification Model

We present a multidimensional synchronization model (Figure 6(b)) which includes the following.

(1) *Dimension of scale and device heterogeneity*. This dimension is based on Steinmetz's model. It includes the four synchronization requirements that we have discussed: intra-stream, intra-media, intra-bundle, and intra-session synchronization layers. The object layer in Steinmetz's model is removed because we only focus on continuous multimedia streams. The specification layer remains for the multilayer skew formulation.

(2) *Dimension of multi-location synchronization controls*. The orthogonal location-based dimension is directly extended from Ehley's model. The multidimensional synchronization model adds synchronization controls at each location together with temporal support for large numbers of heterogeneous devices.

(3) *Dimension of application-dependent synchronization*. We argue that there is a strong demand to add this dimension, to describe the impact of application heterogeneity on human synchronization perception. It is not possible to use uniform synchronization references to represent a multimedia platform. Each application developed for a platform must identify its own references based upon the functionality of performed activities and end user requirements.

4.2 Multi-Location Synchronization Control in TI system

To demonstrate the usage of the multidimensional synchronization model in Figure 6(b), we present an example of collaborative synchronization control framework at multiple locations (tiers) of the interactive TI systems. To the best of our knowledge, the framework is the first to investigate the skews arising from heterogeneous computation demands of multimodal media data. Computation skews can be effectively bounded at the capturing tier, owing to multi-machine multicore cloud deployment. In addition, inter-sender synchronization is new to the research community, and application-dependent synchronization references (Section 2) become an integral part of our framework.

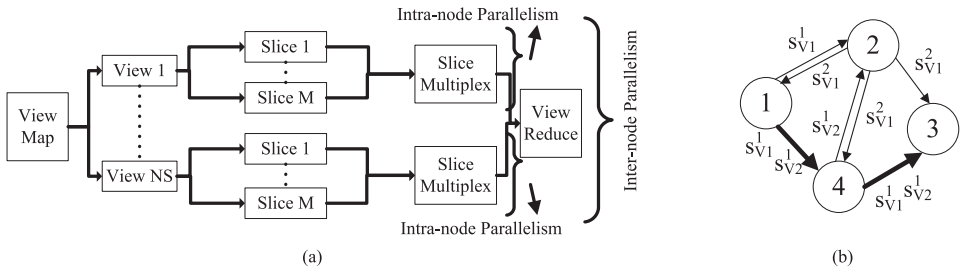

Fig. 7. (a) CloudStream framework design; (b) SyncCast distribution topology.

We rely on RTP/RTCP to implement TI synchronization. The implementation also includes an extension of sender and receiver reports to allow specification of synchronization references selected in our interactive TI systems. The Online Appendix presents an example of the reference selection policy [Huang et al. 2010]. To identify the time correlations of media data sourced at distributed sender sites, we use NTP to perform clock synchronization across the sites. The global time is used for inter-sender and inter-receiver synchronization.

Capturing Tier Control. The purpose is to bound synchronization skews arising from computation heterogeneity at each TI sender site. The heterogeneity is due to the fact that multiple time-correlated media frames can carry different amounts of media information that require unequal CPU resources. The resulting variations in computation overhead within/across the sensory streams cause the intra-stream, intra-media and intra-bundle skews.

We develop CloudStream [Huang et al. 2011], a cloud/grid-based media encoding parallelization and scheduling scheme, to incur only minimal (computation) cost towards multimedia synchronization. The intra-bundle skews are reduced by parallelizing media computation tasks in the cloud infrastructure to speed up the encoding process of computation-intensive media like 3D multiview videos. The intra-stream and intra-media synchronization are realized by deciding the amounts of cloud resources in order to smooth computation jitter. Due to negligible computation overhead of audio, haptics, and etc., we only focus on the parallelization of 3D multiview videos.

In CloudStream (Figure 7(a)), we *map* multiview video streams to multiple compute nodes, and encode different views in parallel without encoding dependency (called *inter-node parallelism*). For each view, we further divide the video frames into multiple non-overlapping partitions to facilitate the multicore parallelism on each compute node (called *intra-node parallelism*). Assuming δ_{cap} is the computation time upper bound of a video frame, $\tilde{T}_{cap}$ is its estimated encoding time using a single CPU (estimated by profiling and content analysis [Huang et al. 2011]), and T_{seq} is the processing time of the sequential portion of the job which cannot be parallelized, the minimal required number of CPUs of the requested node can be computed by $\lceil \frac{\tilde{T}_{cap}-T_{seq}}{\delta_{cap}-T_{seq}} \rceil$. The encoded data are multiplexed into a single stream in the view *reduce* component (Figure 7(a)).

Distribution Tier Control. We present SyncCast [Huang et al. 2011], a synchronized multicast overlay for TI systems (Figure 7(b)) to bound synchronization skews during media distribution over the Internet. SyncCast simplifies the synchronization problem by only focusing on multiview video distribution and the resulting intra-session and intra-media (video) synchronization. It assumes audio, haptics, and other media modalities require negligible bandwidth resources, so their packets can be multiplexed and follow the same distribution path as the video reference stream in the same media bundle. In other words, the intra-bundle and intra-media (e.g., audio, haptics, and etc.) skews are

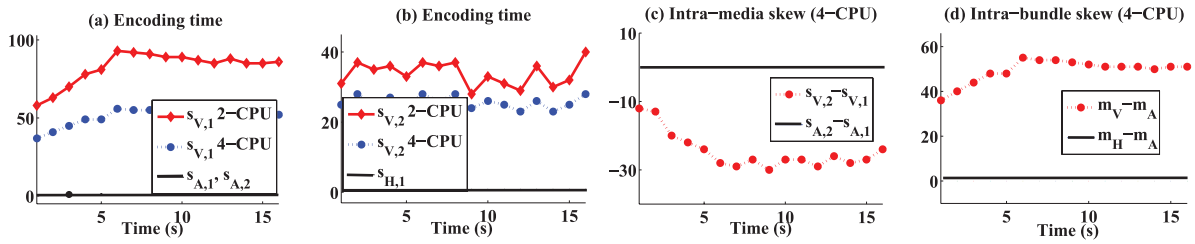

Fig. 8. Evaluation of CloudStream. (a) and (b) Encoding overhead of two video (up to four CPUs), two audio and one haptic streams (unit: ms); (c) and (d) resulting intra-media and intra-bundle skews in case of 4-CPU parallelization (unit: ms). x-label: a duration of 16 seconds. $s_{V,1}$ and $s_{A,1}$: intra-media references, m_A: intra-bundle reference. Superscript representing the site index is omitted.

negligible during media distribution. Internet jitter and the resulting intra-stream synchronization are not studied. We assume they will be solved by buffer compensation at the presentation tier.

The goal is to find a multicast topology with bandwidth and (intra-media, intra-bundle, and inter-sender/receiver) synchronization constraints at the distribution tier. It minimizes the average network latency D_{net} (Section 3.3) of all intra-media (video) synchronization references, rather than the average of all media data in the session. The reason is that reference streams usually carry the most important information to end user perception, so they are given priority in distribution for better interactive quality. Compared to existing delay-bounded multicast overlay studies (Section 3.3), sensory streams within the same media bundle are allowed to follow different paths. For example, in Figure 7(b), site 2 receives the video stream $s^1_{V,1}$ directly from site 1 and $s^1_{V,2}$ via the intermediate site 4.

SyncCast offers a heuristic solution to the NP-hard optimization problem. We first find distribution paths for video reference streams because of their importance. For a reference stream to each receiver site, we follow Rouskas and Baldine [1997] and Shi et al. [2001] (Section 3.3) and iterate k-shortest path options. We add to SyncCast topology the shortest path that can satisfy both synchronization and bandwidth constraints (called a *constraint path*), based on the topology that has been constructed. In case a search does not return any successful constraint path because of bandwidth bottleneck or synchronization violation issues, a multicast adjustment algorithm is used [Huang et al. 2011] to change the existing topology to accommodate new streams. The adjustment is realized by finding alternative constraint paths among k-shortest path options, for reference streams whose constraint paths have been included in the topology. We then decide the shortest constraint path for any other video stream which is not a synchronization reference. Unlike reference streams, we directly discard the unlucky non-reference streams without further adjustment, if no successful constraint path can be found.

Presentation Tier Control. The goal is to add buffering latency to compensate for synchronization errors that are propagated from capturing and distribution tiers. The problem has been extensively studied for intra-stream, inter-stream, and inter-receiver/group synchronization, as discussed in Sections 3.2 and 3.3. Two extensions are needed for TI synchronization: (1) separation of reference stream and reference modality during buffer control for intra-media and intra-bundle synchronization (as opposed to single reference stream in past inter-stream synchronization studies), and (2) use of NTP and global time for inter-sender synchronization (i.e., the receiver adapts buffer to achieve in-sync presentation of media packets from multiple sender sites, by referencing their global captured time specified in the RTP header). The buffer control algorithm is discussed in Huang [2012]. We will not present its details because of space limitations.

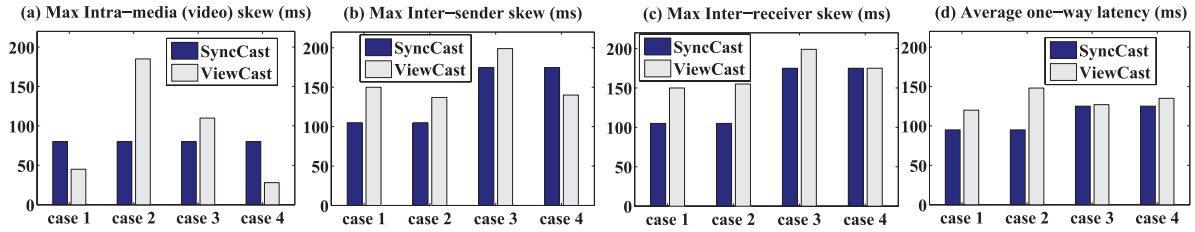

Fig. 9. A comparison between SyncCast and ViewCast. (a)–(c) Maximum intra-media, inter-sender, inter-receiver skews (unit: ms); (d) average latency of video references (unit: ms). Both 5-site and 9-site setups are studied. x-label represents four cases with different combinations of site number and intra-bundle skew bound of SyncCast or one-way delay bound of ViewCast. Case 1: 5-site 200-ms skew/delay bound, case 2: 5-site 300-ms bound, case 3: 9-site 200-ms bound, case 4: 9-site 300-ms bound.

Performance Evaluation. We present a brief evaluation of TI multi-location synchronization control. Without loss of generality, we assume each TI site outputs four QVGA 3D video, two audio, and one haptic streams. Due to space limitations, we focus on CloudStream and SyncCast, and leave out the presentation tier control.

CloudStream. In our cloud testbed, we use up to four CPUs (Intel Xeon 2.8 GHz) for each compute node [Huang et al. 2011]. Figures 8(a)–(b) shows an example of computation time for two video streams ($s^1_{\mathrm{V},1}$, $s^1_{\mathrm{V},2}$), two audio ($s^1_{\mathrm{A},1}$, $s^1_{\mathrm{A},2}$), and one haptics ($s^1_{\mathrm{H},1}$), when using different number of CPUs. The 3D video streams are encoded using the Berkeley codec [Huang 2012], and the overhead of audio/haptic streams is negligible. We find variations in computation time within/across the video streams, because different video frames may carry heterogeneous amounts of visual information. The resulting intra-media and intra-bundle synchronization skews are shown in Figure 8(c)–(d), when we pick $s^1_{\mathrm{V},1}$ and $s^1_{\mathrm{A},1}$ as intra-media synchronization references, and m^1_{A} as intra-bundle reference. Figure 8 validates the use of computation parallelization to reduce the computation jitter.

SyncCast. We evaluate SyncCast using a multisite TI emulator [Huang et al. 2011]. Real network latencies are measured between Planetlab sites, and the mean latency statistics is used as an input of connectivity setups. Internet jitter is not considered. Bandwidth availability is represented as the unit of video stream number. We assume fixed bandwidth overhead for all video streams and negligible overhead for audio and haptic data.

Figure 9 presents an example of five-site and nine-site setups, where sites are evenly distributed in the U.S. Europe, and Asia. Both setups use a maximum allowable intra-media (video) skew of 80 ms, and variable intra-bundle skew bound. As a comparison, we also evaluate ViewCast [Yang et al. 2010], which extends existing delay-bounded multicast studies by allowing video streams within a media bundle to follow different distribution paths. However, ViewCast does not give priority to synchronization references during their distribution. To compare SyncCast with ViewCast, both topologies prescribe same intra-session skew bound. This can be achieved by using SyncCast's intra-bundle skew bound as ViewCast's one-way delay bound. Audio and haptic streams follow the same path as the video reference stream in both SyncCast and ViewCast, and inbound/outbound bandwidth upper bound is set to be ten video streams in the example.

Figure 9(a) shows that SyncCast can consistently achieve intra-media (video) synchronization within the preset 80-ms constraint by dropping video streams which would otherwise incur unbounded skews. The maximum intra-media (video) skews in ViewCast can be as large as the one-way delay bound. Figures 9(b)–(c) presents the bounded intra-session (inter-sender and inter-receiver) skews in both

Table I. Comparisons of Three Specification Models (Section 3.2)

Specification models	Axis-based	Interval-based	Control-based
Implementation	Easy	Complex	Easy
Media objects	Independent	Dependent	Independent
Adding/Removing media objects	Easy	Complex	Easy
Media object duration	Required	Not required	Not required
Sync skew	Supported	Supported	Additional Mechanism Needed

Note: Axis-based: [Hodges et al. 1989], interval-based: [Wahl and Rothermel 1994], control-based: [Steinmetz 1990].

Table II. Comparisons of Inter-Receiver/Group Synchronization Control Algorithms Discussed in Section 3.3

Control algorithms	Receiver-based (Centralized)	Receiver-based (Distributed)	Sender-based (Maestro)	Multicast routing
Centralized/Distributed	Centralized	Distributed	Centralized	Centralized
Adding/Removing receivers	Complex if master is changed	Easy	Easy	Complex
Amounts of ommunication overhead	Medium	Large	Small	Large
Adaptation responsiveness	Round-trip time	Slow	Round-trip time	N/A

Note: Receiver-based centralized: [Akyildiz and Yen 1996; Ishibashi et al. 1997], receiver-based distributed: [Ishibashi and Tasaka 2000], sender-based: [Boronat et al. 2009], multicast routing: [Rouskas and Baldine 1997; Shi et al. 2001; Zimmermann and Liang 2008].

topologies. Figure 9(d) demonstrates that the average latency of video reference streams in SyncCast is lower because of its priority differentiation.

5. CONCLUSION

Distributed multimedia systems and their unique features, such as scalability and heterogeneity of multimodal devices, diversity of applications, and activities, and complexity of spatial and other contextual information are becoming reality in much broader application usage space, due to major advancement of multimedia devices, distributed computing, and network technologies, and due to the drop in cost in putting these technologies together. New applications lead to new requirements for multimedia synchronization. This article reviews the past and current synchronization practices and standards and presents a new multidimensional synchronization model for next-generation multimedia environments. Readers can use this article to study the evolution of synchronization research under the background of multimedia technological advancement and to understand new synchronization challenges that will arise in future multimedia applications.

APPENDIXES

A. COMPARISON SUMMARY OF SYNCHRONIZATION STUDIES

We summarize two comparison tables for the synchronization studies we have presented in Section 3. Table I discusses the synchronization specification models in Section 3.2, and Table II evaluates the inter-receiver/group synchronization control algorithms in Section 3.3.

B. SYNCHRONIZATION REFERENCE SELECTION IN TELEIMMERSIVE (TI) SYSTEM

In this section, we present an example of synchronization reference selection methodology in our current TI implementation. Note that the selection rule is policy-based, meaning that it can vary depending on specific end user interests in different multimedia applications.

Intra-Stream Synchronization. The reference frame or the intra-stream synchronization reference is usually selected as the first media frame within a sensory stream at each system control update.

Hence, other media frames behind it can be played at the output devices by consulting their original captured inter-frame period at the media sensor.

Intra-Media Synchronization. The intra-media synchronization reference is selected as the reference stream which has the largest contribution to end user interests within a media modality. The media contribution varies depending on the characteristics of each modality. Here, we discuss four commonly deployed media modalities that we have used.

Multiview-Videos. Multiview video streams capture the same physical object at the same time, but from different viewpoints. The importance of each video stream is decided by their contributions of 3D image pixels to the end user viewpoint [Huang et al. 2010], which can be computed using the orientation difference between the sender camera and the receiver view. Given the sender n^x's camera orientation of a video stream $s^x_{\mathrm{V},i}$ (denoted as $\vec{O}(s^x_{\mathrm{V},i})$), and the desired receiver n^y's view orientation of n^x's videos (denoted as $\vec{O}^{x,y}$), the visual contribution or the *contribution factor* (CF) of $s^x_{\mathrm{V},i}$ to the receiver n^y is

$$\mathrm{CF}\left(s^x_{\mathrm{V},i}, n^y\right) = \vec{O}\left(s^x_{\mathrm{V},i}\right) \cdot \vec{O}^{x,y} \tag{10}$$

Hence, the video reference stream is elected as the video stream with the largest CF within the video modality for each receiver.

Spatial Audios. Multiple omnidirectional microphones concurrently record the same physical ambient environment. The contribution of each audio stream is decided by its signal-to-noise ratio (SNR), a metric indicating the intelligibility of the speaker's utterances. SNR can be computed online by estimating the noises during silence periods. We prescribe that the audio reference stream is the audio stream with the largest SNR within the audio modality.

Haptic or Body Sensory Streams. Multiple haptic or body sensory streams may record different parts of a physical object. In TI systems, we decide the haptic/body reference stream as the one with the largest data rate within the haptic/body sensory modality, because a larger data rate for these sensory streams usually means higher-precision information.

Intra-Bundle Synchronization. The importance of media modalities can vary at different applications, and the intra-bundle synchronization reference is defined as the most important reference modality. Empirically in TI systems, we classify different applications based on real user perceptual feedback. (1) Users attach more importance to the intelligibility of audio signals in a conversation-oriented application (e.g., conferencing or remote education), so the reference modality is the audio. (2) The clarity of video signals is of the greatest significance in a collaborative task with fine motor skills (e.g., rock-paper-scissor gaming or cyber-archeology), so the video is elected as the reference modality. (3) The body sensory streams have the largest contribution in telehealth or remote rehabilitation applications, because the doctor needs to evaluate a patient's health status by consistent body sensory feedback. Thus, we choose the body sensory modality as the reference.

Intra-Session Synchronization. In multisite interactive multimedia systems, the most active site usually demands higher-quality streaming bundles in order to guarantee uninterrupted collaborations in a session. The intra-session synchronization reference of inter-sender or inter-receiver synchronization is, thus, selected as the media bundle corresponding to the most active user among the senders or receivers. In TI systems, for example, this user usually takes the lead in multimedia applications (e.g., a trainer in the remote education, a director in the conferencing, or a doctor in the telehealth). The selection of the lead person is context-dependent, so it must be specified explicitly by multimedia applications.

REFERENCES

AKYILDIZ, I. F. AND YEN, W. 1996. Multimedia group synchronization protocols for integrated services networks. *IEEE J. Select. Areas Commun. 14,* 1, 162–173.

ANDERSON, D. P. AND HOMSY, G. 1991. A continuous media I/O server and its synchronization mechanism. *IEEE Comput. 24,* 10, 51–57.

BAILEY, B., KONSTAN, J., COOLEY, R., AND DEJONG, M. 1998. Nsync—a toolkit for building interactive multimedia presentations. In *Proceedings of the ACM International Conference on Multimedia*. 257–266.

BARRIOS, R. 1995. Examination of early sound musicals, with extensive coverage of Vitaphone. In *A Song in the Dark*, Oxford University Press.

BELLLABS. 1969. The picture of the future. *Bell Labs Record 47*, 134–186.

BLAKOWSKI, G. AND STEINMETZ, R. 1996. A media synchronization survey: Reference model, specification, and case studies. *IEEE J. Select. Areas Commun. 14,* 1, 5–35.

BLESSER, B. 1978. Digitization of audio: A comprehensive examination of theory, implementation, and current practice. *J. Audio Engi. Soci. 26*, 739–771.

BORONAT, F., CEBOLLADA, J. C. G., AND MAURI, J. L. 2008. Study of delay jitter with and without peak rate enforcement. *J. Multimedia Tools Appli. 40,* 2, 285–319.

BORONAT, F., LLORET, J., AND GARCIA, M. 2009. Multimedia group and inter-stream synchronization techniques: A comparative study. *Inf. Syst. 34*, 108–131.

BRANDENBURG, R., STOKKING, H., VAN DEVENTER, O., VAISHNARI, I., BORONAT, F., AND MONTAGUD, M. 2012. IETF draft: RTCP for inter-destination media synchronization. http://tods.ietf.org/agenda/80/slides/artcore-5.pdf.

BUCHANAN, M. AND ZELLWEGER, P. 2005. Automatic temporal layout mechanisms revisited. *ACM Trans. Multimedia Comput. Commun. Appl. 1,* 1, 60–88.

BULTERMAN, D. 1993. Specification and support of adaptable networked multimedia. *Multimedia Syst. 1,* 2, 68–76.

BUNN, J., NEWMAN, H., AND WILKINSON, R. 1998. Status report from the Caltech/CERN/HP "GIOD" joint project: Globally interconnected object databases. In *Proceedings of the Conference on Computing in High Energy Physics*.

CAMPBELL, A., COULSON, G., GARCLA, F., AND HUTCHISON, D. 1992. Orchestration services for distributed multimedia synchronisation. In *Proceedings of the IFIP International Conference on High Performance Networking*.

CLUVER, K. AND NOLL, P. 1996. Reconstruction of missing speech frames using sub-band excitation. In *Proceedings of the IEEE International Symposium on Time-Frequency and Time-Scale Analysis*. 277–280.

CRONIN, E., KURC, A. R., FILSTRUP, B., AND JAMIN, S. 2004. An efficient synchronization mechanism for mirrored game architectures. In *Proceedings of the 1st Workshop on Network and Support Systems for Games.*

CURCIO, I. AND LUNDAN, M. 2007. Human perception of lip synchronization in mobile environment. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*.

DAMER, B. 1998. *Avatars: Exploring and Building Virtual Worlds on the Internet.* Peachpit Press.

DANNENBERG, R. AND STERN, R. 1993. Experiments concerning the allowable skew of two audio channels operating in the stereo mode. Personal Communications.

EHLEY, L., FURTH, B., AND ILYAS, M. 1994. Evaluation of multimedia synchronization techniques. In *Proceedings of the International Conference on Multimedia Computing and Systems*. 110–119.

FUJIMOTO, T., ISHIBASHI, Y., AND SUGAWARA, S. 2008. Influences of inter-stream synchronization error on collaborative work in haptic and visual environments. In *Proceedings of the IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator System*. 113–119.

GARDNER, B. 1992. A realtime multichannel room simulator. In *Proceedings of the 124th Meeting of the Acoustical Society of America*.

GHINEA, G. AND ADEMOYE, O. A. 2010. Perceived synchronization of olfactory multimedia. *IEEE Trans. Syst. Man, Cybernet. 40,* 4, 657–663.

GOLDMANN, L., LEE, J.-S., AND EBRAHIMI, T. 2010. Temporal synchronization in stereoscopic video: Influence on quality of experience and automatic asynchrony detection. In *Proceedings of the IEEE International Conference on Image Processing*. 3241–3244.

HODGES, M., SASNETT, R., AND ACKERMAN, M. 1989. Athena Mouse: A construction set for multimedia applications. *IEEE Software*.

HOSHINO, S., ISHIBASHI, Y., FUKUSHIMA, N., AND SUGAWARA, S. 2011. Qoe assessment in olfactory and haptic media transmission: Influence of inter-stream synchronization error. In *Proceedings of the IEEE International Workshop on Communications Quality and Reliability*. 1–6.

HU, N. AND STEENKISTE, P. 2002. Estimating available bandwidth using packet pair probing. Tech. rep. CMU-CS-02-166. Carnegie Mellon University.

HUANG, Z. 2012. Synchronized distribution framework for high-quality multi-modal interactive teleimmersion. Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

HUANG, Z., MEI, C., LI, L., AND WOO, T. 2011. Cloudstream: Delivering high-quality streaming video through a cloud-based H.264/SVC proxy. In *Proceedings of the IEEE International Conference on Computer Communications*.

HUANG, Z., WU, W., NAHRSTEDT, K., AREFIN, A., AND RIVAS, R. 2010. Tsync: A new synchronization framework for multi-site 3D tele-immersion. In *Proceedings of the ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*.

HUANG, Z., WU, W., NAHRSTEDT, K., RIVAS, R., AND AREFIN, A. 2011. Synccast: Synchronized dissemination in multi-site interactive 3D tele-immersion. In *Proceedings of the ACM Multimedia Systems Conference*.

IEEE. 2008. IEEE 1588 standard: Precise time synchronization as the basis for real time applications in automation. http://www.nist.gov/el/isd/ieee/ieee1588.cfm.

IETF. 2003. IETF RFC3550 standard - RTP: A transport protocol for real-time applications. http://www.ietf.org/rfc/rfc3550.txt.

ISHIBASHI, Y. AND TASAKA, S. 2000. A comparative survey of synchronization algorithms for continuous media in network environments. In *Proceedings of the IEEE Conference on Local Computer Networks*. 337–348.

ISHIBASHI, Y., TSUJI, A., AND TASAKA, S. 1997. A group synchronization mechanism for stored media in multicast communications. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies*. 692–700.

KOHLER, E., HANDLEY, M., AND FLOYD, S. 2006. RFC4340: Datagram congestion control protocol (DCCP).

LEROUX, P., VERSTRAETE, V., DE TURCK, F., AND DEMEESTER, P. 2007. Synchronized interactive services for mobile devices over ipdc/dvb-h and umts. In *Proceedings of the IEEE/IFIP International Workshop on Broadband Convergence Networks*. 1–12.

LITTLE, T. 1993. A framework for synchronous delivery of time-depdent multimedia data. *Multimedia Syst. 1,* 2, 87–94.

LITTLE, T. AND GHAFOOR, A. 1991. Spatio-temporal composition of distributed multimedia objects for value-added networks. *IEEE Computer 24,* 10, 42–50.

MEYER, T., EFFELSBERG, W., AND STEINMETZ, R. 1994. A taxonomy on multimedia synchronization. In *Proceedings of the IEEE Workshop on Future Trends of Distributed Computing Systems*. 97–103.

MICHEL, U. 2006. History of acoustic beamforming. In *Proceedings of the Berlin Beamforming Conference*.

PICTURETEL. 1991. Picturetel in project with I.B.M. *New York Times*, October 22, 1991.

QIAO, L. AND NAHRSTEDT, K. 1997. Lip synchronizatioin within an adaptive VoD. In *Proceedings of the SPIE Multimedia Computing and Network*. 170–181.

RAMANATHAN, S. AND RANGAN, P. 1993. Feedback techniques for intra-media continuity and inter-media synchronization in distributed media systems. *Computer J. 36,* 1, 19–31.

RAVINDRAN, K. AND BANSAL, V. 1993. Delay compensation protocols for synchronization of multimedia data streams. *IEEE Trans. Knowl. Data Eng. 4,* 5, 574–589.

ROTHERMEL, K. AND HELBIG, T. 1995. An adaptive stream synchronization protocol. In *Proceedings of the ACM International Workshop on Network and Operating System Support for Digital Audio and Video*. 189–202.

ROUSKAS, G. N. AND BALDINE, I. 1997. Multicast routing with end-to-end delay and delay variation constraints. *IEEE J. Select. Areas Commun. 15,* 3, 346–356.

SHI, S. Y., TURNER, J. S., AND WALDVOGEL, M. 2001. Dimensioning server access bandwidth and multicast routing in overlay networks. In *Proceedings of the ACM International Workshop on Network and Operating System Support for Digital Audio and Video*. 83–92.

STEINMETZ, R. 1990. Analyse von synchronisation mechanismen mit anwendung im multimedia-bereich. In *Proceedings of the GI ITG Workshop Sprachen und System zur Parallelverarbeitung*. 39–47.

STEINMETZ, R. AND ENGLER, C. 1993. Human perception of media synchronization. Tech. rep. 43.9310, IBM European Networking Center Heidelberg.

STEINMETZ, R. AND NAHRSTEDT, K. 1995. *Multimedia Computing, Communications and Applications*. Prentice Hall.

STOCKHAM, T. 1972. A/D and D/A converters: Their effect on digital audio fidelity. In *Digital Signal Processing, IEEE Press*, 55–66.

TOV, S.-Y. 2005. Happy 10th birthday, VoIP. Haaretz, June 16, 2005.

WAHL, T. AND ROTHERMEL, K. 1994. Representing time in multimedia systems. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*. 538–543.

WALLIS, A. 1995. Cambridge corners the future in networking. TWANZ 5, 10.

WILLIAMS, A., GROSS, K., BRANDENBURG, R., AND STOKKING, H. 2013. RTP clock source signalling. IETF.

WOO, M., QAZI, N., AND GHAFOOR, A. 1994. A synchronization framework for communication of pre-orchestrated multimedia information. *IEEE Netw. 1,* 8, 52–61.

YANG, Z., WU, W., NAHRSTEDT, K., KURILLO, G., AND BAJCSY, R. 2010. Enabling multi-party 3d tele-immersive environments with viewcast. *ACM Trans. Multimedia Comput. Commun. Appl. 6,* 2.

YAVATKAR, R. 1992. MCP: A protocol for coordination and temporal synchronization in collaborative applications. In *Proceedings of the IEEE International Conference Distributed Computing Systems*. 606–613.

ZIMMERMANN, R. AND LIANG, K. 2008. Spatialized audio streaming for networked virtual environments. In *Proceedings of the ACM International Conference on Multimedia*. 299–308.

Received October 2012; revised March 2013; accepted March 2013