Quality of Service (QoS) Metrics for Continuous Media

Duminda Wijesekera[*]and Jaideep Srivastava[†]
Distributed Multimedia Research Center
Department of Computer Science, University of Minnesota
Minneapolis, MN 55455.
e-mail: {wijesek|srivasta}@cs.umn.edu

June 27, 1996

Keywords: continuous media, multimedia, QoS, synchronization, media mixing

Abstract

This paper presents quality of service (QoS) metrics for continuity and synchronization specifications in continuous media (CM). Proposed metrics specify continuity and synchronization, with tolerable limits on average and bursty defaults from perfect continuity, timing and synchronization constraints. These metrics can be used in a distributed environment for resource allocation. Continuity specification of a CM stream consists of its sequencing, display rate and drift profiles. The sequencing profile of a CM stream consists of tolerable aggregate and consecutive frame miss ratios. Rate profiles specify the average rendition rate and its variation. Given a rate profile, the ideal time unit for frame display is determined as an offset from the beginning of the stream. Drift profile specifies the average and bursty deviation of schedules for frames from such fixed points in time. Synchronization requirements of a collection of CM streams are specified by mixing, rate and synchronization drift profiles. Mixing profiles specify vectors of frames that can be displayed simultaneously. They consist of average and bursty losses of synchronization. Rate profiles consist of average rates and permissible deviations thereof. Synchronization drift profiles specify permissible aggregate and bursty time drifts between schedules of simultaneously displayable frames. It is shown that rate profiles of a collection of synchronized streams is definable in terms of rate profiles of its component streams. It is also shown that mixing and drift profiles of a collection of streams are non-definable in terms of sequencing and drift profiles of its constituents. An important consequence of the mutual independence of synchronization and continuity specification is that, in a general purpose platform with limited resources, synchronized display of CM streams may require QoS tradeoffs. An algorithm that makes such tradeoffs is presented as a proof of applicability of our metrics in a realistic environment.

---

[*] Thanks to Deepak R. Kenchamanna-Hosekote, Sujal S. Parikh, Youjip Won and an anonymous referee for many valuable suggestions and numerous corrections.

[†] This work is sponsored by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

Contents

List of Figures

List of Tables

1 Introduction

Preliminary descriptions of *high quality* are frequently non quantitative, although improvement in quality requires a quantitative understanding of the service/object under discussion. The cost of high quality services/products are of paramount importance, because, although unlimited resources may fetch the best of all possible worlds, it may be unaffordable or non profitable to do so. Consequently, the quality vs. cost tradeoffs have to be considered.

Multimedia (MM) is no exception to this rule. Although unlimited resources may get the best possible quality of multimedia services, such luxury may not be needed for a particular application. Consequently, a need exists for a specification method of *quality of service* (QoS) for on demand continuous media (CM) services. The current paper proposes metrics to quantitatively describe QoS in CM presentations.

CM streams are distinguished from the non continuous streams of data by their *continuous nature* of progression, which is qualitatively well understood. For example, there already exists an intuitive understanding of the continuity of flow in audio and video streams, which form the major types of CM streams. Thus, one of the main tasks of quantifying QoS is defining a specification metric for *continuity*. In addition, most CM presentations consist of a synchronized flow of a collection of streams, such as TV broadcasts consisting of audio and video. A qualitative definition of a synchronization specification is well known as *lip synchronization*. Thus, quantifying synchronization requires defining metrics for it.

The utility of a metric designed for computer usage depends on three simple, but stringent requirements. The first is that they should be simple enough to be specified. The second is that they must be implementable with ease. The third, which is required of all metrics is that they should measure relevant characteristics, thereby faithfully quantifying qualitative descriptions.

We claim that our metrics are simply specifiable, and easy to implement. Although the first claim has to stand the test of time, we intend to back up our second claim by providing a simple display site algorithm. Our thesis is that QoS metrics proposed in this work faithfully quantify parameters important to CM applications.

A basic characteristic of continuous media is the *continuous nature* of media streams, which entails delivering large amounts of data with real time deadlines. Because of operating systems, networks, disks etc, general purpose computing environments are inherently non deterministic in nature. Hence, unless special precautions are taken, there can be delays and disruptions in jitter free lossless delivery and display of media frames. Consequently, the solution proposed by research and industrial communities to overcome such disruptions is to have a guaranteed CM service with application specific QoS parameters such as display rates, synchronization granularity and allowable media losses, etc.

In light of this proposal, we examine application needs of continuous media. Firstly, continuity and synchronization requirements are application dependent. For eg., TV broadcast requires tight lip synchronization while video browsing does not. Also, the granularity of synchronization is application dependent, eg. stereo voice mixing is of finer granularity than audio-text synchronization. Furthermore, tolerable limits of defaults from rigidly specified continuity and synchronization requirements such as the frequency and amount of frame delays, glitches, skips, pauses and synchronization losses are application dependent. These parameters affect CM presentations in different ways. For eg., frame losses and skips result in unexpectedly short display durations, bursty losses results in clutter, and loss of synchronization results in poor comprehensibility. Consequently, in order for an application to convey its requirements to the underlying system, QoS metrics must include all the above mentioned parameters.

1.1 Problem Description

The problem addressed by this paper is to formulate QoS metrics relevant for CM applications. Specification metrics should be such that they can be used by a general purpose computing environment with limited resources to intelligently isolate inherent non determinacy and limitations of the underlying systems from degrading application QoS. Naturally, specification metrics should be *faithful* in the sense that they convey parameters relevant for applications and be easily specified by application writers and implemented by system developers. In this paper our attention is focused on metrics that can specify average values and tolerable variations thereof for relevant parameters.

1.2 Relevant Work

Issues related to QoS can be broadly categorized into the following three categories [VKvBG95]:

1. Assessing QoS in terms of user's subjective wishes or satisfaction with the quality of application-performance, synchronization cost etc,
2. Mapping results of the assessment onto QoS parameters for various system components.
3. Negotiation between system components or layers (embedded protocols) to ensure that all system components can meet the required QoS parameter values consistently.

According to this characterization, the current paper falls into the *assessing user QoS requirements* category. Although we do not directly report experimental results of user requirements, we present a model for user QoS metrics and their ramifications. Published work in the general area of QoS for CM services are numerous. For lack of space, we summarize only some, with all due apologies for the authors of the others. First of all [VKvBG95] provides an up to date survey of issues related to QoS in distributed multimedia systems. In the area of user QoS assessment, an important experiment in estimating the tolerable levels of media drifts in human perception of media synchronization is described in [SE93]. While it provides a wealth of data, tolerable limits of synchronization in the presence of lossy media and higher order effect of drifts remain to be investigated. In the same area, perceptual effects caused by changing frame rates in dynamic QoS are described in [AFKN94]. They develop a video classification schema (VCS) based on dimensions of temporality, audio content and video content. In this study, based on user surveys and physiological tests, each dimension of video quality, audio quality and frame rate is categorized as *high* or *low*. They show the ratings that a select group of viewers provided for logos/test patterns, snookers, talk shows and stand up comedies. Hence, this paper addresses QoS issues at a higher level than from ours, and is oriented towards physiological issues in audio visual QoS metrics and user studies to decide their importance. In subsequent work [AFKN95], an experiment to measure user responses to video frame rate degradation without affecting audio quality has been reported. These experiments provide valuable information because in addition to setting a precedence of experimental methodology they also provide values for user QoS parameters.

In the area of modeling user QoS requirements, application level QoS metrics given in [FSS95] consist of two classes. They are intra-media continuity and inter-media synchronization. Latency (delay from expected display time), temporal crops (error rate in the temporal scale), spatial crops (rate of skipped information in the spatial scale) and continuity (smoothness of processing information) are stated as parameters of intra-media continuity. Latency of the beginning and the allowable time lag between start times of multiple streams are categorized as parameters of inter-media synchronization. This paper claims that latency, temporal/spatial crops and continuity can be translated to delay, bandwidth and delay jitter of the underlying transport system. The paper also identifies the necessity to have two service classes for delivery and client site management: *guaranteed* and *best effort*. Although this paper addresses most aspects of CM *quality*, it does not offer explicit metrics to measure them.

In the same area [RTP94] presents a context free grammar (CFG) to describe a functional model of a CM service and a user interacting with its presentation. The chosen QoS parameters are throughput, delay, transmission reliability and inter-channel relationships. The paper shows CFG expressions for freezing and restarting, scaling the presentation speed and spatial requirements, handling spatial clashes, skipping events, navigating in time, and reverse presentations. Although relevant, this paper addresses QoS issues at a higher level than that of ours.

A communicating sequential process (CSP) based specification language to specify CM processes is presented in [SW94]. Events in these processes have to be explicitly mapped to time intervals of a global clock by scripts. Events that can be omitted or added and conditions not satisfied by any trace of the process can be specified externally. Although such mechanisms are in general capable of specifying almost any QoS measures for single streams, the paper does not explicitly offer any metrics of either continuity or synchronization. Secondly, more work is necessary to extend this framework to include inter-stream relationships and relative timings of events, and such relationships are of paramount importance in specifying synchronization information. Furthermore, this framework has a global clock built into its semantic interpretation, which has been challenged by other researches [RR93] in this area.

The review of QoS based resource management reported in [NS95] provides a number of relevant QoS parameters for CM streams and synchronization. At the application level, given QoS metrics for audio are sample size, rendition rate and playback point. For video, application level QoS are frame rate and dimensions, color resolution, aspect ratio and compression ratio. For synchronization, the only QoS metric provided is temporal skew. Our QoS metrics for CM streams have rendition rate as a parameter, whereas audio and video specific metrics are left out as stream specific parameters. For synchronization, our metrics have average and bursty temporal skews.

The QoS broker [SN95] is a comprehensive framework to utilize application specified QoS metrics throughout a distributed multimedia system. Although any QoS metric can be specified in this framework in general, the explicit metrics offered for CM streams are divided into two groups; media quality and transmission character. Media quality metrics are sample size and rendition rate. Transmission character metrics are end-to-end delay, sample loss rate and *importance*. Except for importance, our metrics include the rest.

The quality of service architecture proposed in [CCH94], is another comprehensive solution to utilize application specified QoS to manage resources in a distributed environment. There, an application specifies its QoS in the form of a *contract*. A *contract* consists of flow, commitment, adaptation, maintenance, connection and cost specifications. Flow specifications consist of frame sizes, rates, bursts, losses and jitter in an interval. Hence, like ours, they are interval based specifications. Commitment specifications determine the class of service commitments the application demands from the service provider and consists of three: deterministic, statistical and best-effort. These service commitments are applied to throughput, loss, delay and jitter. Adaptation metrics are possible corrective action specifications in case of QoS changes. Maintenance metrics specify if the QoS metrics should be monitored by the service provider. Connection specifications parameterize start and termination times and *service negotiation types*, such as fast non-negotiated, negotiated or a future service reservation. Although our metrics specify *only* continuity and synchronization parameters, our parameters of loss and delay consist of average and bursty components over specified intervals, and subsume metrics proposed in this paper.

In [RR93], continuity is parameterized by frame rates with a permissible variation thereof. Thus media misses, skips and pauses are not modeled as continuity parameters. This paper, being one of the first to consider continuity and synchronization of CM streams, develops algorithms to service a display site with limited intelligence, controlled by servers connected through lossless networks with bounded delays. The proposed synchronization QoS is the maximum permissible time lag between simultaneously displayable frames. The authors further go on to integrate their model with server design and media mixing in later papers [RV93], [Ran93].

A significant amount of work has been done in the related area of network scheduling with specified QoS metrics for CM data delivery; for eg., [Tok92], [Tow93], [Mil95] and [Fer93]. In [Tow93], QoS metrics for network traffic have been characterized as those that need deterministic, probabilistic, or best effort guarantees. They consider delay, delay jitter and packet loss as parameters that require such guarantees. The paper, which is an exhaustive survey, goes on to show how the guarantees are translated into routing, congestion control and bandwidth allocation requirements in the underlying service layers of a network. Hence, the main emphasis of this paper is towards providing QoS guarantees in packet switched networks, which is relevant, but of tangential importance to the current work. Nevertheless, their metrics for CM delivery are of importance in our context, because QoS metrics specified at the application level need to be translated and supported by underlying layers.

The Tenet Group provides a set of schemes and protocols for multimedia delivery in the more general context of real time communication [FBZ92]. They have two transport protocols, *Real-Time Message Transport Protocol (RMTP)* [FBZ92], [Fer93], [FV90] and *Continuous Media Transport Protocol (CMTP)* [WM91], [FGMW92] running on top of the packet transmission protocol *Real-Time Internet Protocol (RIP)* [FV90], [FBZ92] on virtual circuits setup by *Real-Time Channel Administration Protocol (RCAP).* The QoS metrics used in these protocols are bounds on delay, delay jitter and probabilities of delay violation and buffer overflow. Furthermore, *channel groups* [GM93] have been proposed as a new abstraction to specify inter channel relationships. Specifiable metrics include inter-stream synchronization, and sharing. The client being able to specify such relationships results in lower cost, better resource utilization and improved scalability of communication. Although this work is primarily addresses issues relevant to network channels, our work indicates a strong need to have relationships between synchronized CM streams even at the application interface.

The work reported in [Mil95] describes QoS parameters and their translation through the layers of the *XTPX* protocol in the *RACE* [Bau92] project. The *XTPX* QoS parameters consist of data throughput, delay, delay jitter, data loss rate, other traffic requirements such as multiplexing possibilities, and reliability requirements. This paper discusses a specific implementation of QoS parameters, and is oriented towards network issues.

Work reported in [RB93], focuses on protocols that synchronize CM data streams across packet switched networks. They specify QoS metrics for three layers of CM delivery protocols, namely application layer, transport layer and network layer. In [RB93] application level continuity is parameterized by an overall rate, inter-glitch spacing, inter frame pause and divergence thereof, where missing data frames are said to result in glitches. Thus, inter-glitch spacing specifies the permissible frequency of glitches. Inter-frame pause and divergence specifies permissible deviations from specified rendition time of frames. Synchronization QoS parameters are vectors of frames that are to be displayed together, and the temporal divergence between their start times. A significant amount of work has been done in [RB93] to translate application QoS to network QoS. This work is directly relevant to us. Our QoS metrics for the application layer subsume the corresponding metrics of [RB93], although we do not describe QoS parameters for other layers.

[NK82] is about audio communication. It discusses two policies to handle audio packets arriving after a specified deadline at a destination node in a packet switched network. The policies are:(1) wait until a late packet arrives, causing a delay in the whole queue (called the *I policy*), or (2) discard any late packets and continue processing succeeding packets (called the *E policy*). The paper shows the effect of buffering and delaying the beginning of display on minimizing the delay jitter of the output stream. In this work, QoS metrics of a CM stream are inter-packet delay and delay jitter. This study is limited to lossless packet switched networks. Hence, media misses, skips and pauses are not considered. Also, there are no QoS metrics for synchronization.

In [JS95], building upon the foundations of [NK82], continuity parameters of audio streams are specified by permissible delays, and satisfied by using different scheduling policies. The authors show that their history based buffer overflow handling algorithm (called *queue monitoring*) performs significantly better than both policies (I policy and E policy) used in [NK82], in case of history based simulations taken from video conferencing applications. Consequently, from our standpoint, this paper provides a significantly better buffer overflow handling algorithm to eliminate delay jitter of the output stream, but does not offer explicit metrics to measure continuity of CM streams or synchronization between them.

In work surveyed so far, either synchronization is considered with lossless media streams, or we have lossy CM streams without tolerable levels of synchronization. Our work attempts to fill that gap by formulating QoS specifications that encapsulate both these concerns.

1.3 Our Contributions

We define QoS parameters for continuity and synchronization specification of CM presentations. Our metrics are average frame rate and its variation, aggregate and bursty losses of frame misses, timing defaults, rate defaults, and synchronization. We show that our synchronization metrics cannot be subsumed by sufficiently stringent continuity metrics. Further we show that, in the absence of dedicated resources, delivery of a synchronized collection of CM streams requires tradeoffs between qualities of continuity and synchronization. As a proof of applicability of our specifications, we offer a QoS based integrated scheduling algorithm that can be utilized for presentation management. The scheduling algorithm depends upon recovery policies to deal with server site starvation. Consequently, the quality of a CM presentation depends upon both the specified QoS and the recovery policy, as shown in Fig. 1(A).

1.4 Organization of the Paper

Sections 2 defines QoS parameters for continuity. Section 3 defines QoS parameters for synchronization. Section 4 describes relationships between continuity and synchronization QoS metrics. Section 5 describes the solution space available for a QoS based display site manager and compromises needed to be made in synchronized delivery of CM streams. It provides a QoS based scheduling algorithm. Finally, Sect. 6 concludes the paper. A road-map of sectional dependencies of the paper appears in Fig. 1(B).

2 QoS Parameters of Stream Continuity

The displayable units of CM data are called samples, i.e. sound samples and video frames. Rendition of a continuous media stream consists of displaying a sequence of samples with a regular frequency. Some streams may record naturally evolving phenomena such as a sunset or a developing storm and hence controlling or rolling back is beyond the realm of the recorder or the audience. Other streams, such as graphical animations, may be within controllable limits of human influence.

The *quality* of a CM stream is important for its rendition and recording, and is justifiably receiving considerable attention. A number of factors contribute to the quality of a video stream,

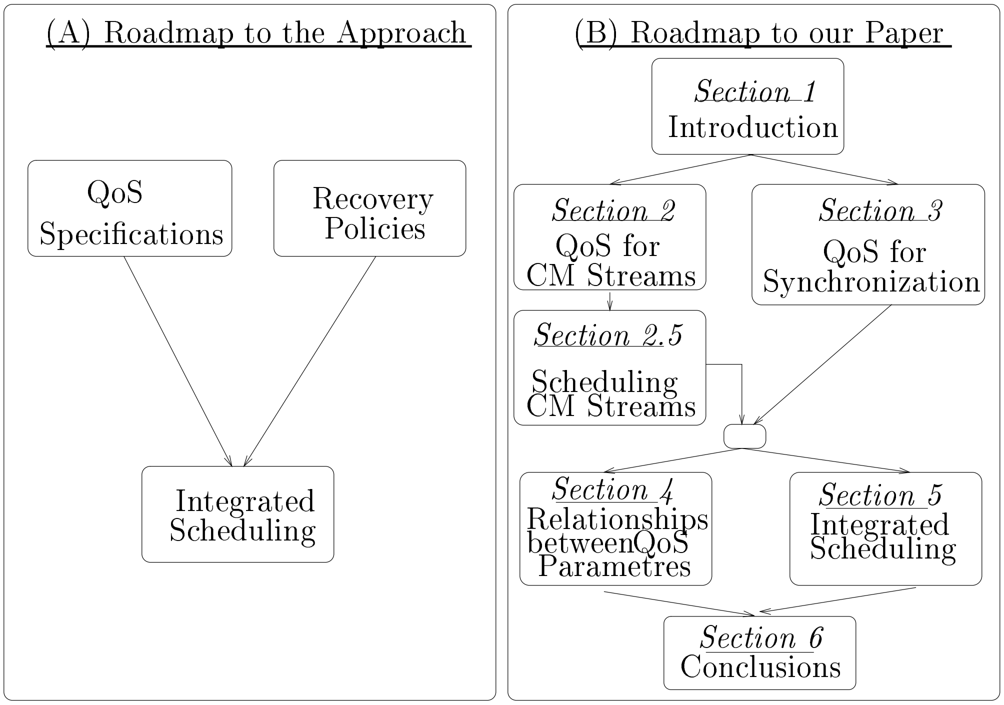

Figure 1: Roadmaps

e.g. frame size, RGB values, hue, brightness, contrast and pixel size etc. For audio, these include encoding law, volume, and amplification factor etc.

Sampling frequency, hence rendition rate is an important parameter which measures the granularity of a discrete approximation to a continuous phenomenon. Although higher frequency yield better approximations, it also increases cost, and thus needs to be only within limits of application requirements. Eg., if the consumer of a video stream is a motion detection program which can handle only up to 30 frames per second, it is a waste of resources to sample an evolving scene at a higher frequency.

In a CM stream with a fixed sampling frequency, the first sample determines the sequence of subsequent samples. This leads to the concept of an *ideal sample sequence* of a CM stream. Modulo the discretization of a continuous phenomena, the ideal sample sequence of a stream records the sequence of all samples that perfectly capture the phenomenon in the proper order of its evolution.

Some applications can tolerate infrequently missing a few samples, without losing the continuity of the rendered phenomenon. Since losing too many samples too frequently may be a serious hindrance, there is a need to specify the maximum number and frequency of acceptable sample droppings. Such a specification is called a *sequencing profile*.

Acknowledging that an ideal sample sequence has a fixed rate of recording, in light of the inability of a general purpose system to control timing points with arbitrary precision, there needs to be a mechanism to specify rendition rates and their acceptable variations, called a *rate profile*.

Due to a large number of factors within a computer system, precise adherence to a fixed schedule is almost impossible, while bounded drifts occurring infrequently may be acceptable by some applications. *Drift profile* is a mechanism to specify acceptable drifts of schedules and their frequencies.

In this study, our continuity metric for CM streams is called a *continuity profile*. It consists of sequencing, rate and drift profiles.

2.1 Media Granules

In media such as video, it is possible for a presentation system to process, schedule, transport and render CM data one sample at a time. In other types of media, eg. audio, this may not be so, since:

- Frequency of processing/scheduling samples becomes too high to handle individual samples.
- Process context switch times are too large for any corrective action in the interval between successive samples.
- Many devices require more than one sample to begin rendering, eg. MPEG

Consider an audio stream with a nominal rate of 44kHz, where to fetch and render a single sample of audio at such a high frequency the rendering process must execute once every 1/44000 seconds, i.e 0.02 milliseconds. This is impossible to achieve in a presentation platform with an average context switch time of 10 milliseconds, which is the time to render 88 audio samples, because it takes 20 milliseconds to switch context twice. Also, there may be hardware requirements, such as an audio buffer requiring at least 256 samples before the device starts rendering.

For stated reasons, the unit of processing/scheduling CM data will often be an integral multiple of samples, called a *media granule*, with the number of samples depending on media type. The number of samples in a media granule is the *media granularity* of the media type. An *ideal granule sequence* with media granularity g, is an ideal sample sequence where every successive g samples are packaged into media granules. The ideal granule sequence provides a baseline to measure content losses of a CM stream.

2.2 Sequencing Information

In order to discuss sequencing information, envision the evolution of a CM stream as a train of slots with successive slot numbers. We denote a stream by $s(\cdot)$, where the sequence of successive slots are numbered $s((0)), s((1)), \ldots$. As for their contents, if $s(\cdot)$ is an ideal granule sequence, then $s(j)$ is the j^{th} media granule. Consequently, for an ideal granule sequence, slot $s((j))$ contains media granule $s(j)$.

In any given rendition of stream $s(\cdot)$, not all slots may be filled, or they may not be filled in the proper order; i.e. it may not be the case that slot $s((j))$ is occupied by media granule $s(j)$. Our notation for subsequence of a CM stream is the index with respect to its ideal sequence, i.e, if $\{s(i) : 1 \leq i\}$ is the ideal granule sequence of a stream, then $(2, 4, 6, 8, \ldots)$ is our notation for the granule subsequence $\{s(2), s(4), s(6), s(8), \ldots\}$. Omission of granules is modeled by the special symbol $\perp$. For eg. $\{2, \perp, 4, 6, 8, \ldots\}$ is our notation for the subsequence $\{s(2), \perp, s(4), s(6), s(8), \ldots\}$.

Specification of which subsequences are acceptable by continuity requirements is the *sequencing profile* of a CM stream. We consider only non-decreasing sequences of media granules. Hence any potential media granule sequence can deviate from its ideal sequence due to three causes:

- Skipping media granules: Eg., the stream *1,4,5,8,10,12* skips media granules *2,3,6,7, 9* and *11.* Skipping some media granules does not leave empty slots on the time line of a media stream.
- Repeating media granules: Eg., the stream *1,2,2,3,3,4,4,4,5* repeats media granules *2, 3* and *4.*

- Missing media granules altogether: Eg. the stream *1,⊥,2,⊥,3,4,⊥* misses media granules for the second, fourth and seventh time slots on the display time line. Missing media granules leaves empty slots on the time line of a CM stream.

All three of these causes can combine to produce a stream of media granules with sequencing disruptions. E.g., the stream $1, 1, 1, \perp, 2, 4, 5, \perp$ skips media granule 3 altogether, repeats media granule 1 twice and misses media granules for fourth, and eighth slots.

In order to measure sequencing disruptions caused by skips, repetitions and misses, we introduce *unit sequencing loss* (USL) . To define unit sequencing loss, envision a CM stream as a train of slots with successive slot numbers, as given in Fig 2. Some slots may be filled with media granules. We define a unit sequencing loss only for slots that are non empty, i.e. they are filled with some media granule. Suppose $s(k)$ is the media granule at slot $s((i))$ of stream $s(\cdot)$. Suppose the immediately previous non empty slot to slot $s((i))$ is slot $s((i \perp l))$, where $l > 0$, and it is occupied by media granule $s(j)$. In case there are no skips, repeats or misses, if slot $s((i))$ is occupied by media granule $s(k)$, then slot $s((i \perp l))$ should be occupied by media granule $s(k \perp l)$. Hence the unit sequencing loss incurred at slot $s((i))$ due to skips and repeats is $\|k \perp l \perp j\|$. The unit sequencing loss due to missing media granules at slot $s((i))$ is $(l \perp 1)$, precisely because there are $(l \perp 1)$ empty slots in between slots $s((i))$ and $s((i \perp l))$. Hence the maximum of sequencing losses due to skips, repeats and misses at slot $s((i))$, say $USL(i)$, is $\max\{\|k \perp l \perp j\|, l \perp 1\}$. Consequently, we define $\max\{\|k \perp l \perp j\|, l \perp 1\}$ to be the *unit sequencing loss* at slot $s((i))$. In order to measure the sequencing loss at the beginning of a stream, we assume that every stream has a hypothetical slot $s((\perp 1))$ with number $\perp 1$ with a hypothetical media granule $s(\perp 1)$.

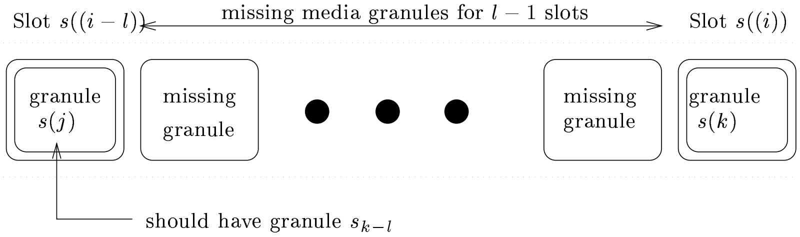

Figure 2: Unit Sequencing Loss

Now, we use unit sequencing losses to specify sequencing profiles. Our sequencing profile specifies allowable average and bursty unit sequencing losses, which are specified by *aggregate loss factor (ALF)* and *consecutive loss factor (CLF).* We use the notation (ALF, CLF) for a sequencing profile.

An ALF of $n//m$ of a stream means that n is the sum of unit sequencing losses allowed within any window of m successive slots for media granules. i.e. $\max\{\sum_{k=i}^{i+m}\{USL(k) : USL(k) \neq \perp\} \leq n$ for any $i \geq 1$.

Consecutive sequencing loss factor (CLF) is the maximum sum of non zero consecutive unit sequencing losses. i.e. max $\{\sum_{k=i}^{i+l}\{USL(k) : USL(k) \neq \perp, 0\ \forall k\ (i \leq k \leq i+l)\} : i, l \geq 1\} \leq CLF$.

For example, consider the stream $\{2, 2, \perp, 3, 5\}$. Its ALF and CLF can be calculated as given in Table 1. Thus according to Table 1 the aggregate loss factor ALF and consecutive loss factor are respectively $4//5$ and 2.

Because our definition of unit sequencing loss accounts for lost media granules, our definition of ALF subsumes metrics of aggregate losses presented in other work such as [Tow93].

Ideal Media granule = Slot Number = $s((j))$	1	2	3	4	5
Presented Media granule $s(j)$	2	2	$\perp$	3	5
Unit Sequencing Loss (USL(j))	1	1	$\perp$	1	1
Aggregate Media Granule Loss (ALF(j))	1	1	2	1	1
Consecutive Media Granule Loss (CLF(j))	1	1	$\perp$	1	1

Table 1: Unit, Aggregate and Consecutive Loss Factors

2.3 Uniformity of Flow

Another important parameter for CM stream delivery is the rendition rate. Important characteristics of such rates include average and first/higher order variations. For the present discussion, we select average rendition rate (in slots per second) and its variation as specifiable characteristics. A CM stream has a *rate profile* of (ρ, σ) when its average rendition rate is ρ and its allowable variation is σ; i.e. the instantaneous rate of rendition lies in the closed interval $[\rho \perp \sigma, \rho + \sigma]$. In any acceptable rendition every constituent media slot must begin within a specific time interval. Consider a stream $s(\cdot)$ with a rate profile (ρ, σ). If the i^{th} slot $s((i))$ of $s(\cdot)$ appeared at time t_i, to be acceptable the $(i+1)^{th}$ slot $s((i+1))$ must appear in the time interval $[t_i + \frac{1}{\rho+\sigma}, t_i + \frac{1}{\rho-\sigma}]$. See Fig. 3.

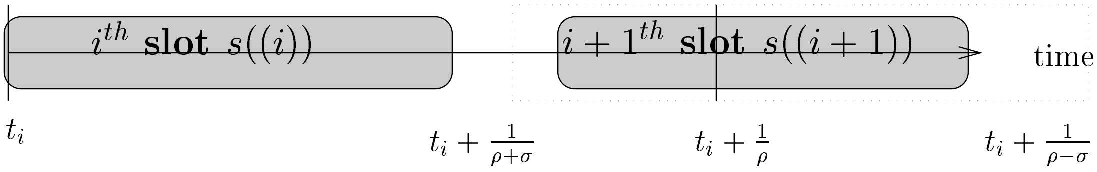

Figure 3: Uniformity of Flow

2.4 Faithfulness to Real Time

For a clock, faithfulness to real time is characterized by drifts and first/higher order rates of drift from some idealized *wall clock* [Chr89]. In face of inherent non determinacy and limitations of hardware/software clocks, the need for such measures arises out of application needs to be within controllable margins of error from idealized wall clock times. Also, similar to clock values, perceivable continuity of CM streams is sensitive to drifts [RB93], which we characterize by *drift profiles.*

Given a rate profile, every granule of a stream has an interval of time to begin its rendition. However, in a pathological case, this can result in an unusually short or long duration of total rendition, as shown in Fig. 4. The drift profile places stronger restrictions on allowable drifts of media granules than that required by rate profiles. We limit the average and bursty behavior of such drifts by specifying the *aggregate drift factor* (ADF) and the *cumulative drift factor* (CDF).

Consider a stream $s(\cdot)$ with a rate profile (ρ, σ), where, with a constant rate ρ slot $s((i))$ should start at time T_i and in practice it starts at time t_i. To be compliant with (ρ, σ), the valid interval for the beginning of $s((i+1))$ is t_{i+1}, where $t_{i+1} \in [t_i + \frac{1}{\rho+\sigma}, t_i + \frac{1}{\rho-\sigma}]$. Define the *unit granule drift at slot* $s((i+1))$, $UGD(i+1)$ as the time difference between t_{i+1} and T_{i+1} i.e. $UGD(i+1) = \|t_{i+1} \perp T_{i+1}\|$, when t_{i+1} is defined. See Fig. 5. If the media granule $s(j)$ is omitted, then t_j is $\perp$, and hence $\|t_j \perp T_j\|$ and $UGD(j)$ are undefined. Using the sequence of UGD's $\{UGD(i) : i \geq 1\}$, we can define the

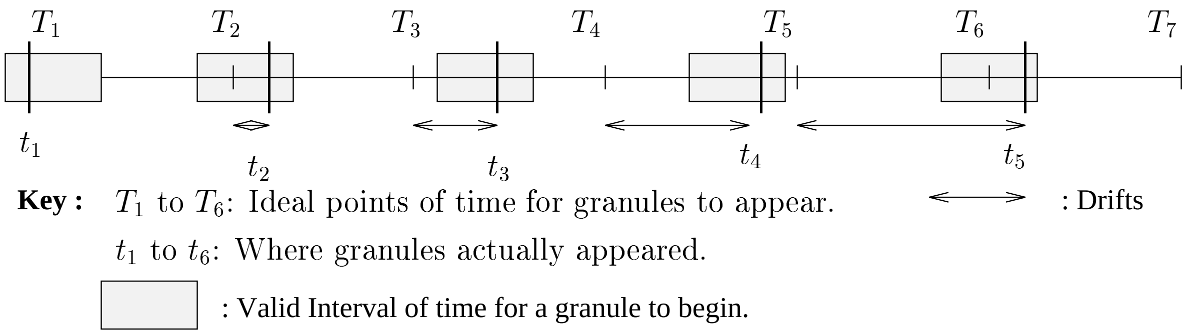

Figure 4: Faithfulness to Real Time

Ideal Rendition Time T_j	1.0	2.0	3.0	4.0	5.0
Actual Rendition Time t_j	0.99	2.02		3.97	5.09
Unit Granule Drift $UGD(j)$	0.01	0.02		0.03	0.09
Aggregate Drift $ADF(j)$	0.01	0.03		0.06	0.15
Consecutive Drift $CDF(j))$	0.01	0.03		0.03	0.12

Table 2: Example of Unit, Aggregate and Consecutive Drift Factors

drift profile (ADF, CDF). An ADF of $d//m$ means that no consecutive m granules can have a sum of more than d time units of granule drift, i.e. $\sum_{k=i}^{i+m}\{UGD(k) : UGD(k) \neq\perp\} \leq d$ for any $i \geq 1$. A CDF of d' means that the sum of consecutive non zero delays can be at most d' time units, i.e. $\max\{\sum_{k=i}^{i+l}\{UGD(k) : UGD(k) > 0 \; \forall k \; (i \leq k \leq i+l)\} : i, l \geq 1\} \leq d'$.

Consider the example given in Fig. 5, where the ideal presentation times, i.e T_1, T_2, T_3, T_4 and T_5 are 1.0, 2.0, 3.0, 4.0 and 5.0, whereas actual presentation times, i.e., t_1, t_2, t_3, t_4, t_5 are $(T_1 \perp 0.01), (T_2 + 0.02), \perp, (T_4 \perp 0.03), (T_5 + 0.09)$. Hence UGD's are 0.01, 0.02, 0.0, 0.03 and 0.09, the resulting ADF is 0.01 + 0.02 + 0.03 + 0.09 = 0.15 per five granules. The largest consecutive drift is 0.03 + 0.09, between $s((4))$ and $s((5))$, and hence the CDF is 0.12. Table 2 clarifies these calculations.

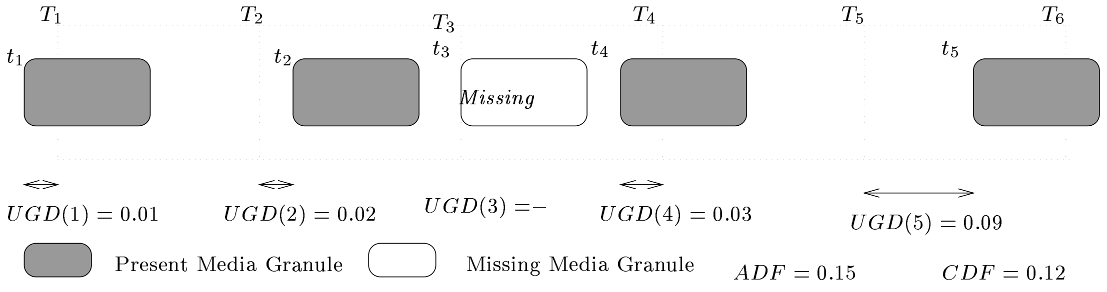

Figure 5: Unit Granule Drifts

2.5 Scheduling CM Streams

We now investigate scheduling media granules of a CM stream so that its display satisfies the specified continuity profile. Two main factors to be determined each time a scheduler is invoked are:

- What to show as the next media granule ?
- When to show the next media granule ?

We show that the starting time of a media granule can be chosen from an interval, and that there is a *best time instant* to start a media granule. Also, any granule from a given set can be chosen and similarly there exists a *best media granule* to be displayed. Because loss and drift factors summarize the history of a rendition, schedulers based on these parameters can make decisions based on content wise and time wise history up to that point. We call this *history based scheduling.*

2.5.1 History Based Scheduling: Content

ALF specifies the aggregate media granule losses, i.e. the number of media granules that can be skipped, repeated or omitted from display. To see how, consider a CM stream $s(\cdot)$ with the sequencing profile $(ALF, CLF) = (l//m, l')$. To schedule $s(\cdot)$ at any given slot $s((p))$ of the display, granules that were shown during the past m slots need to be known. Let $D(k,p)$ be the sum of unit sequencing losses within the window of last p media granule at slot $s((k))$; i.e. as given in (1).

$$D(k,p) = \sum_{i=k-p}^{k} \{USL(i) : \text{ if } USL(i) \neq \perp\} \tag{1}$$

Then, the next media granule can be skipped or repeated if $D(k, m \perp 1) < l$. Generalizing, for $s(\cdot)$ to satisfy an ALF of $l//m$, the maximum number of media granules that can be skipped, repeated or omitted is $l \perp D(k, m \perp 1)$. See Fig. 6.

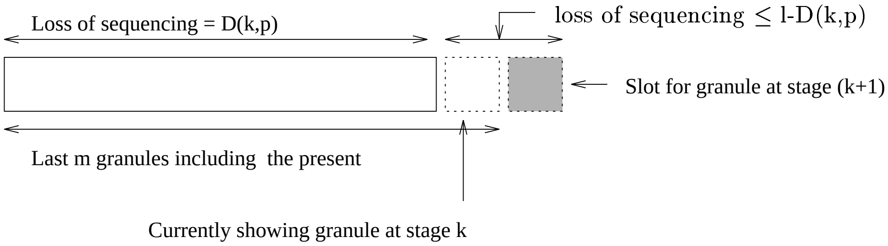

Figure 6: ALF based Constraints

CLF determines the number of consecutive granules that can be skipped at slot k. To see how, consider the stream $s(\cdot)$ with sequencing parameters $(l//m, l')$. Let $C(k)$ be the sum of consecutive non zero unit continuity losses at slot k. To satisfy a CLF specification of l', the maximum number of frames that can be skipped for the succeeding frame slot $s((k+1))$ is $l' \perp C(k)$. Thus, to satisfy both CLF and ALF specifications, the number of media granules that can be skipped, repeated, or missed is $\max\{l' \perp C(k), l \perp D(k, m \perp 1), 0\}$.

Consequently, to be compliant with a sequencing profile, the succeeding media granule must be picked from an interval that is given in (2) and (3). (3) says that either skip the next media granule or pick a candidate from the interval $[u(k), u(k) + L(k)]$. In these equations, $u(k)$ is the last displayed media granule.

$$[u(k), u(k)] \quad \text{if} \quad L(k) = 0 \tag{2}$$

$$\{\perp\} \cup [u(k), u(k) + L(k)] \quad \text{if} \quad L(k) > 0 \tag{3}$$

$$\text{where } L(k) = \max \ \{l' \perp C(k), l \perp D(k, m \perp 1), 0\} \tag{4}$$

However, the granule that has the *best* sequencing properties (i.e. the one that contributes to the least loss factors) is calculated as follows. If there is a media granule q displaying at slot k, then the *best media granule* to show for slot $s((k+1))$ is $q+1$. If not, and $s(j)$ is the last shown media granule at slot $s((i \perp l))$ then $s(j+l+1)$ is the *best media granule* for slot $k+1$. See Fig. 5.

2.5.2 History Based Scheduling: Timing

For a CM stream to be compliant with a drift profile $(d//m, d')$, the next granule to be scheduled is restricted by the history of unit granule drifts. To calculate the interval of rendition imposed by a rate profile, consider a stream $s(\cdot)$ with QoS parameters $((ALF, CLF), (\rho, \sigma), (ADF, CDF))$, where $(ADF, CDF) = (d//m, d')$. If a media slot $s((i))$ of stream $s(\cdot)$ appeared at time t_i, then under a fixed rate ρ the successor of $s((i))$, $s((i+1))$ appears at time $(t_i + \frac{1}{\rho})$. For the rendition to be compliant with the rate profile (ρ, σ), $s((i+1))$ must have a maximum drift of $min\{d, d'\}$ from $(t_i + \frac{1}{\rho})$. Thus, the rate profile restricts the maximum drift to be $\max\{\frac{1}{\rho} \perp \frac{1}{\rho+\sigma}, \frac{1}{\rho-\sigma} \perp \frac{1}{\rho}\} = \frac{\sigma}{\rho(\rho-\sigma)}$. Hence, to satisfy the rate profile, the rendition interval for the next media granule should be within the interval $[t_i + \frac{1}{\rho+\sigma}, t_i + \frac{1}{\rho-\sigma}]$.

To see the effect of drift profiles, for each slot $s((i))$ define $S_{ADF}(i)$ as the sum of unit drifts within the preceding $(m \perp 1)$ slots. Also, define $S_{CDF}(i)$ as the total consecutive non zero unit drifts. Define $D(i) = \min\{d \perp S_{ADF}(i), d' \perp S_{CDF}(i)\}$. Hence, $D(i)$ is the maximum drift slack available for $s((i+1))$, and satisfies $D(i) \geq 0$. Hence, to satisfy the drift profile, the rendition interval for the next media slot should be within the interval $[T_i + \frac{1}{\rho} \perp D(i), T_i + \frac{1}{\rho} + D(i)]$. Hence, if $s((i))$ appeared at time t_i, then the permissible time interval for its successor $s((i+1))$ to begin is given in (5).

$$[\ \max\ \{t_i + \frac{1}{\rho+\sigma}, T_i + \frac{1}{\rho} \perp D(i)\},\ \min\ \{t_i + \frac{1}{\rho \perp \sigma}, T_i + \frac{1}{\rho} + D(i)\}] \tag{5}$$

Consequently, we need to ensure or find conditions under which the interval given in (5) is non empty. Necessary and sufficient conditions for this interval to be non empty are given by

$$t_i + \frac{1}{\rho+\sigma} \quad \leq \quad T_i + \frac{1}{\rho} + D(i) \tag{6}$$

$$t_i + \frac{1}{\rho \perp \sigma} \quad \geq \quad T_i + \frac{1}{\rho} \perp D(i) \tag{7}$$

Equations (6) and (7) reduce to (8) and (9) respectively . They are satisfied if (10) holds. Notice that because $D(i)$ measures the maximum possible drift from the ideal presentation time, it satisfies to satisfy $0 \leq D(i) \leq min\{d, d'\}$. Consequently (11) is sufficient to satisfy (10).

$$(t_i \perp T_i) \quad \leq \quad \frac{\sigma}{(\rho+\sigma)} + D(i) \tag{8}$$

$$(T_i \perp t_i) \quad \leq \quad \frac{\sigma}{(\rho \perp \sigma)} + D(i) \tag{9}$$

$$\parallel t_i \perp T_i \parallel \quad \leq \quad \frac{\sigma}{(\rho+\sigma)} + D(i) \tag{10}$$

$$\parallel t_i \perp T_i \parallel \quad \leq \quad \frac{\sigma}{(\rho+\sigma)} \tag{11}$$

Hence the next timing interval is given by (12).

Statistic	Time Slot	1	2	3	4	5	6	7	8
Content	Ideal media granule	1	2	3	4	5	6	7	*8*
	Presented media granule	1	⊥	2	3	5	8	10	*10 ... 13*
	Unit sequencing Loss (USL)	0		1	0	1	2	1	*0 ... 2*
	Aggregate media granule loss (ALF)	0		1	1	2	4	5	*5 ... 7*
	Consecutive media granule loss (CLF)	0		1	0	1	3	4	*0 ... 6*
Timing	Ideal time of media granule appearance	0	33	66	99	132	165	198	*231*
	Actual Time of media granule appearance	10		56	99	132	145	188	*221 ... 241*
	Unit media granule drift (UGD)	10		10	0	0	20	10	*0 ... 10*
	Aggregate media granule drift (ADF)	10	10	20	20	20	40	50	*50 ... 60*
	Consecutive media granule drift (CDF)	10		10	0	0	20	30	*30 ... 40*

Table 3: History of media granule Rendition

$$[T_i + \frac{1}{\rho} \perp \min\{\frac{\sigma}{\rho(\rho+\sigma)}, D(i)\}, T_i + \frac{1}{\rho} + \min\{\frac{\sigma}{\rho(\rho+\sigma)}, D(i)\}] \qquad (12)$$

2.5.3 An Example of Stream Scheduling

As an example of stream scheduling, consider rendering a video stream with the following parameters.

- A media granule is a frame.
- The sequencing profile is $(8//30, 6)$.
- The rate profile is (33 frames/s, 10 frames/s) $= (\frac{1}{33}, \frac{1}{100})$ in frame/ms.
- The drift profile is (100 ms //30, 40 ms).

Given the rate profile (33 frames/s, 10 frames/s), in an ideal rendition of successive frames they should appear 33.3 milliseconds apart, starting with the first frame and continuing with successive frames. Suppose when the scheduler is invoked for the 8^{th} time slot the history of schedules appears as given in Table 3.

Based on the history of rendition, the ideal eighth media granule to be presented has index 11. Because the consecutive media granule loss at time slots 7 is 4 and the specified CLF is 6, the largest index of the granule that can be presented without violating the specified CLF is 13. The smallest granule that can be shown at time slot 8 is 10. Since the specified ALF is 8//30 and the aggregate media granule loss at time slot 7 is 5, three more misses can be tolerated without violating ALF specification. Notice that 13 and 10 are respectively the largest and smallest k in the interval given by (2) and (3). Hence, the range of media granules to be displayed at the eighth slot is $\perp \cup [10, 12]$, as predicted by (2).

Notice that the aggregate media granule drift at the seventh slot is 50 milliseconds and the specified ADF is 100//30. Thus, there is a slack of 50 milliseconds at time slot 7. Since the CDF is 30 and the specified CDF is 40, there is a restriction placed by consecutive drifts for the eighth media slot to be rendered within 30 milliseconds. Consequently, as per notation of Sect. 2.5.2, maximum slack drift for the eighth media slot, $D(8)$ is $\min\{100 \perp 60, 40 \perp 30\} = 10$. At a rate of 33 frames/second the ideal time of rendition for the eighth media granule is 231 ms. Using (12), the interval of rendition is $[\ 231 - \min\{\frac{100}{30(1/30+1/100)}, D(8)\},\ 231 + \min\{\frac{100}{30(1/30-1/100}), D(8)\}] =$ [221, 241].

3 QoS Parameters of Inter-stream Synchronization

Many continuous media presentations consist of synchronized renditions of multiple CM streams. For example, in TV movies there are audio streams synchronized with video streams. In such synchronized renditions, the granularity of synchronization may differ from application to application, or even between different segments of the same application. For example, one application/segment may require tight lip synchronization between audio and video streams whereas another application/segment may only require spoken words displayed with a changing background. As with CM streams, synchronization requirements of different applications are specified by quality of service (QoS) parameters. This section examines such parameters and their properties.

Consider a synchronized audio-video rendition such as a TV movie. Suppose a 1/2 second portion of the audio stream was missed without skipping the corresponding portion of the video stream. Then the rest of the presentation will suffer from a poor quality of correlation between the audio and video streams. This example shows that simultaneously displayed chunks of streams play a key role in the quality of a synchronized rendition. It is not only the sequence of samples, but also the combination of them across streams that matters. Inherent losses in communication media and unforeseen transient overloads of short durations result in synchronization loss due to mismatches of samples across streams. Some applications may not be able to tolerate such mismatches. Hence, there is a need for a mechanism to specify the amount and frequency of mismatches across streams. We propose such a specification mechanism called *mixing profiles* in Sect. 3.2.

Another factor contributing to the quality of a synchronized presentation is the rate of rendition. As in the case of single streams, rate and its allowable variation can be specified by a *rate profile* for synchronized renditions, as discussed in Sect. 3.3.

Another issue relevant to synchronized renditions is relative time drifts of granules across streams that need to be ideally shown together. Consider the problem of lip synchronization. Although a drift of microsecond magnitude between a video granule and the corresponding audio segment may not be distinguishable by a human audience, frequent drifts of few seconds may be a severe hindrance to the viewers. Hence there is a need to specify acceptable time drifts and their frequencies of occurrence across streams. We propose *synchronization drift profiles* for that purpose, as defined in Sect. 3.4.

Consider again the example of a TV movie. Suppose that a portion of the video stream had to be dropped due to some communication delay. If the missing portion is small, the recovery point of the restored quality can be quick. Thus, the units of synchronization can also contribute to the grossness, granularity, and hence to the overall quality of a synchronized rendition.

3.1 Synchronization Granules

Continuity properties of CM streams are specified in terms of media granules. The number of samples in a media granule depends only upon its type, and hence play out times of different types of media granules may be different. To synchronize between time intervals of different types of media, atomic units of synchrony must have the same play out length. Furthermore, desired granularity of synchrony may differ from one composition to another or within different segments of the same composition. In order to satisfy both these needs, for each segment of a CM presentation in which synchronization granularity does not change, we choose an atomic unit from each synchronized stream, so that units across streams have equal rendition times. We call such a unit a *synchronization granule* of the corresponding stream segment. Because continuity of each stream is maintained in units of media granules, we require that a synchronization granule contain an integral number of media granules, called the *synchronization granularity.* Notice that unlike media granularity, synchronization granularity depends upon the presentation, and perhaps the segment of the presentation and the media type. For synchronization granules of different media types to have equal display times, there must be relationships between rendition rates and synchronization granularities. This is discussed in Sect. 3.3.

Given an integer n to be taken as synchronization granularity, any sequence of media granules can be represented as a sequence of synchronization granules by packaging every successive n media granules into a synchronization granule. The sequence of synchronization granules corresponding to an ideal media granule sequence is called the *ideal synchronization granule sequence* of the segment.

3.2 Mixing Granules for Synchronization

Given two or more CM streams, perfect synchrony between them is obtained when rendition of all i^{th} synchronization granules begin together and end together. Thus, for the sequencing aspect of synchrony, all synchronization granules with the same index must be rendered simultaneously. Any diversion from this ideal situation is measured by means of aggregate and consecutive losses thereof, resulting in controlled average and bursty losses of synchrony. We call our mixing parameters of synchrony as *aggregate mixing loss factor* ($AMLF$) and *consecutive mixing loss factor* ($CMLF$). Also the mixing component of synchronization parameters ($AMLF, CMLF$) is our *mixing profile.*

To provide precise definitions of $AMLF$ and $CMLF$, let $S = \{s_i(\cdot) : i \leq n\}$ be a collection of streams where $s_i(\cdot) = \{s_i(j) : 1 \leq j\}$ is the sequence of rendered synchronization granules of the i^{th} stream $s_i(\cdot)$. Then we define the *unit mixing loss at slot k*, say $UML(k)$ to be the maximum drift within any two concurrent media streams, symbolically defined as max $\{\|s_i(k) \perp s_j(k)\| : 1 \leq i, j \leq n, i \neq j$ where $s_i(k), s_j(k) \neq \perp\}$. $UML(k)$ is a measure of synchronization loss at slot k. For example, in streams 1, 2, and 3 in Fig. 7, unit mixing losses for slots 1 through 12 are given as 1,2,2,1,0,0,0,0,2,1,0 and 0. Notice that the definition of UML ignores measuring synchronization drifts with missed synchronization granules. As a consequence, when only one synchronization granule of a vector is present, unit mixing loss UML is undefined, and denoted by $\perp$.

Also, define the *m-aggregate mixing loss at slot i* (say $UML_m(i)$) as $\{\sum_{k=i-m}^{i} UML(k) \neq \perp\}$. m-aggregate mixing loss at any slot measures the aggregate synchronization loss within a window of m, up to and including the i^{th} slot. In the example of Fig. 7, $UML_2(4)$ is the sum of $UML(4) + UML(3) = 1 + 2 = 3$.

We say that S has an *aggregate mixing loss factor* ($AMLF$) of $l//m$ (where l, m are integers) if m-aggregate mixing loss $UML_m(i)$ is at most l for any slot i, i.e. if and only if $\forall i \; UML_m(i) \leq l$.

We define the *consecutive mixing loss at slot i*, say $CML(i)$ of S to be $max\{\sum_{k=j}^{i} UML(k) : j \leq i$ and $UML(k) > 0$ for $j \leq k \leq i\}$, i.e. it defines the largest aggregate consecutive non zero unit mixing losses up to and including slot i. In the example of Fig. 7, notice that there is a non zero unit mixing loss in slots 1 through 4. Consequently, $CML(4)$ is $\sum_{k=1}^{4} UML(k) = 1+2+2+1 = 6$.

We say that a schedule has a *consecutive mixing loss factor*, say $CMLF$, of l' if it satisfies $CML(i) \leq l'$ for all slots i. We say that S has a mixing profile of $(l//m, l')$ if it has an $AMLF$ of $l//m$ and $CMLF$ of l'.

Consider the example of synchronized rendering of three streams as given in Fig. 7. The index of synchronization granules of each stream, and their ALF's and CLF's are given in Fig. 7. Unit mixing loss for each synchronized vector is given below it. The unit mixing loss for any slot is

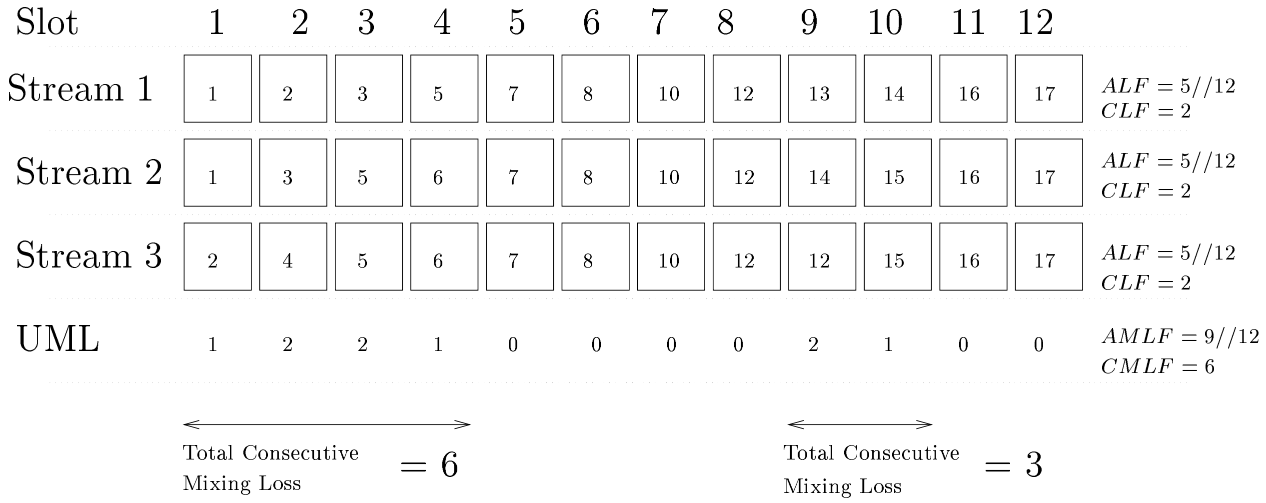

Figure 7: Rendition of Synchronized Streams

calculated by taking the maximum of the differences of the indices of synchronization granules for that slot. For example, for slot 3 the maximum difference between 3, 5 and 5 is 2. Hence, the unit mixing loss for slot 3 is 2. The sequence of unit mixing losses is 1, 2, 2, 1, 0, 0, 0, 0, 2, 1, 0, 0. Based on this sequence, the sum of unit sequencing losses for 12 slots is 9, giving an $AMLF$ of 9//12. Similarly, the largest sum for non zero subsequence of the unit mixing loss sequence is 1+2+2+1 = 6, giving a $CMLF$ of 6.

3.3 Uniformity of Flow

Unlike mixing profiles, uniformity of flow and faithfulness to real time are specifications about rendition times. Thus, the same vector sequence of synchronization granules can be rendered with varying timing sequences to produce different effects. The flow of synchronization granules are specified and controlled through the use of a *schedule*.

A *schedule* of a CM stream is a (potentially infinite) non decreasing sequence of real numbers interspersed with $\perp$'s. A stream $s_i(\cdot)$ given as a sequence of synchronization granules $s_i(\cdot) = \{s_i(j); 1 \leq i\}$ is said to be rendered according to the schedule $t_i(\cdot) = \{t_i(j) : 1 \leq j\}$ if the rendition time of synchronization granule $s_i(j)$ is $t_i(j)$. Because missing synchronization granules need not be scheduled, we require $t_i(j) = \perp$ whenever $s_i(j) = \perp$. We say that *the rendition slot* of $t_i(j)$ is j.

An array of streams $S = (s_1(\cdot), \ldots, s_n(\cdot))$ is said to be rendered according to the schedule $T = (t_1(\cdot), \ldots, t_n(\cdot))$, (where each $t_i(\cdot)$ is a schedule for a stream) if each stream $s_i(\cdot)$ is rendered according to the schedule $t_i(\cdot)$. Thus, given a stream $s_i(\cdot)$ and a schedule of rendition $t_i(\cdot)$, rate and drift profiles of $s_i(\cdot)$ determine relationships between elements of $t_i(\cdot) = \{t_i(j) : 1 \leq j\}$. This was discussed in Sect. 2.5.2. The schedule $T = (t_1(\cdot), \ldots, t_n(\cdot))$ for a synchronous rendition of an array $S = (s_1(\cdot), \ldots, s_n(\cdot))$ of streams specify relationships between all elements of $\{t_i(j) : 1 \leq i \leq n, 1 \leq j\}$.

As stated in Sect. 2.3, uniformity of flow parameters of a CM stream $s_i(\cdot)$ is given by its rate profile (ρ_i, σ_i), where ρ_i is the nominal rate and σ_i is the permissible deviation from the nominal rate ρ_i. For any stream $s_i(\cdot)$, ρ_i and σ_i are given in media granules per time unit. Because synchronization granularities of streams are known, synchronization granules are composed of media granules accordingly and rate profiles can be translated to units of synchronization granules per time unit. This calculation is given in Sect. 4. The translation yields a rate and its variation for synchronization granules.

Because synchronization granularities of an array of streams are chosen so that corresponding granules across streams have the same rendition time, the rate of synchronized rendition is automatically determined by the rates of constituent streams. Similarly, the rate variations for constituent streams can be translated to the basis of synchronization granules. Since they don't have to be equal, the maximum rate variation of a synchronized rendition is taken to be the maximum of the rate variations of the component streams.

Thus, parameters of flow uniformity in a synchronized rendition are rendition rate, say ρ_{sy} and rate variation, say σ_{sy}, which measure the average behavior and burstiness of a schedule. (ρ_{sy}, σ_{sy}) is said to be the *synchronization rate profile* of a synchronized rendition.

A schedule $T = \{t_i(\cdot) : 1 \leq i \leq n\}$ of a synchronized rendition that is compliant with a rate profile (ρ_{sy}, σ_{sy}) of a set of streams $S = \{s_i(\cdot) : 1 \leq i \leq n\}$ has the following relationships among $\{t_i(j) : 1 \leq i \leq n\}$.

$$\forall i, j \leq n, 1 \leq k : \qquad t_j(k+1) \in [t_i(k) + \frac{1}{\rho_{sy} + \sigma_{sy}}, t_i(k) + \frac{1}{\rho_{sy} \perp \sigma_{sy}}] \tag{13}$$

Equation (13) relates scheduling points across streams. The intention being that, no matter what stream provides the synchronization granules, the rate and rate variation calculated between any two of them should satisfy the specified rate profile.

3.4 Relative Time Drifts in Synchronized Schedules

In order to avoid frequent synchronization losses to an unacceptable degree, there is a need to restrict the amount and frequency of relative drifts across synchronized schedules. We propose to specify and control such drifts by *synchronization drift profiles.* A synchronization drift profile has two components: namely, *aggregate synchronization drift factor* ($ASDF$) and *consecutive synchronization drift factor* ($CSDF$), controlling the average and bursty drifts of schedules across synchronized streams.

In order to give precise definitions, consider a collection of n schedules $T = \{t_i(\cdot) : i \leq n\}$, where $t_i(\cdot) = \{t_i(j) : 1 \leq j\}$. Then, define the *unit synchronization drift* of T at slot k as:

$$\begin{aligned} USD(k) &= \perp, \text{ if all } \ t_i(k) = \perp \quad \text{for } \ 1 \leq i \leq n. && (14) \\ &= \max\{\|t_i(k) \perp t_j(k)\| : 1 \leq i \neq j \leq n \ \text{ and } \ t_i(k), t_j(k) \neq \perp\}, \ \text{ otherwise} && (15) \end{aligned}$$

We define the *m-aggregate synchronization drift at slot i* , say $USD_m(i)$, as $\{\sum_{k=i-m}^{i} USD(k) : 1 \leq i$ and $USD(k) \neq \perp\}$. It measures the maximum aggregate synchronization drift within the window of m successive slots ending with i of a schedule. We say that T has an *aggregate synchronization drift factor*, say $ASDF$, of $d//m$ (where d is in time units and m is an integer) if m-aggregate synchronization drift at any slot is at most d, i.e. if $\forall\ i\ USD_m(i) \leq d$. We define the *consecutive synchronization drift at slot i*, say $CSD(i)$, of T to be $max\{\sum_{k=j}^{i} USD(k) : j < i$ and $0 < USD(k)$ for $j \leq k \leq i\}$, i.e. it defines the largest sum of consecutive non zero unit timing drifts of a vector of schedules. We say that a schedule has a *consecutive synchronization drift factor*, say $CSDF$, of d' if its $CSD(i) \leq d'$ for all slots i, where $CSD(i)$ is the consecutive synchronization drift of T at slot i. We say that a schedule T satisfies a *synchronization drift profile*, say SDP, of $(d//m, d')$ if it has a $ASDF$ of $d//m$ and a $CSDF$ of d'.

Consider the example given in Fig. 8. Suppose that the schedules of the first four synchronization granules of streams $s_1(\cdot)$ and $s_2(\cdot)$ are given as $t_1(\cdot) = (t_1(1), t_1(2), t_1(3), t_1(4)) = (1.0, 1.8, 2.8, 3.8)$

and $t_2(\cdot) = (t_2(1), t_2(2), t_2(3), t_2(4)) = (1.2, 2.0, 2.8, 4.1)$. Suppose their ideal schedule is $t_{ideal}(\cdot) = (1, 2, 3, 4)$. Then, the sequence of unit synchronization drifts is (1.2 - 1.0, 2.0 - 1.8, 2.8 - 2.8, 4.1 - 3.8) = (0.2, 0.2, 0.0, 0.3). Hence, the aggregate of unit synchronization drifts is (0.2+0.2+0.0+0.3) = 0.7 for 4 slots, giving an $ASDF$ of 0.70//4. The maximum over sums of consecutive nonzero unit synchronization drifts is $0.2 + 0.2 = 0.4$. Hence the $CSDF$ is 0.4.

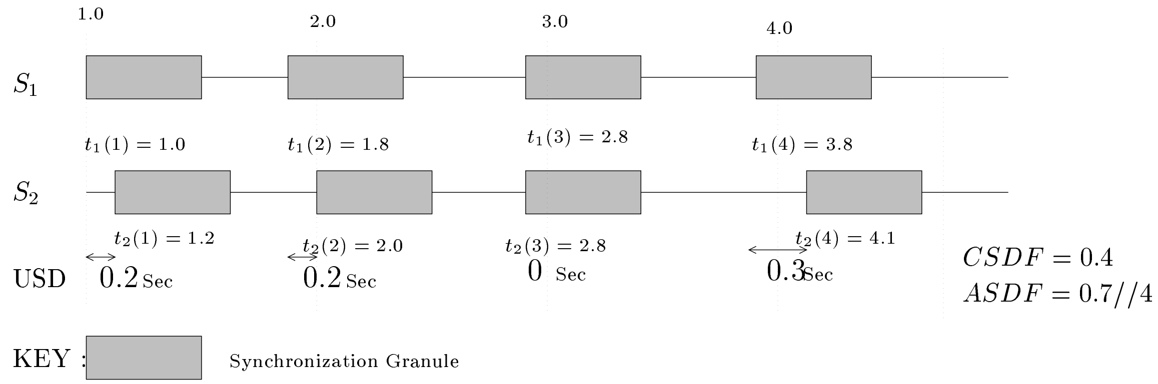

Figure 8: Controlling Drifts of Synchronization Granules

3.5 History Based Schedules for Synchronized Streams

QoS parameters of synchronization are mixing profile $(AMLF, CMLF)$, rate profile (ρ_{sy}, σ_{sy}), and drift profile $(ASDF, CSDF)$. In this section, we describe relationships between them and possible constraints they place on synchronized rendition of streams. As in the case of stream rendition, scheduling consists of the content of the next vector of synchronization granules, and its rendition time. The content is determined by mixing profiles, whereas rendition time is determined by a combination of rate and synchronization drift profiles.

3.5.1 Contents of Synchronization Granules

Suppose we want to determine the contents of synchronization granules at slots $k+1$, $\{s_i(k+1) : 1 \leq i \leq n\}$ of a collection of n streams $S = \{s_i(\cdot) : 1 \leq i \leq n\}$. Towards this end, suppose that at slot k of a synchronized display, the consecutive unit mixing loss is $CMLF(k)$ and the aggregate unit mixing loss over the past r slots of the display is $AMLF(r, k)$. Consequently, they must satisfy conditions given in (16) and (17), collectively given as (18). Inequality (16) states that consecutive mixing drift must be bounded by l', while (17) states that aggregate mixing drift within any m slots must be bounded by l.

$$
\begin{aligned}
UML(k) + CMLF(k) &\leq l' && (16)\\
UML(k) + AMLF(m \perp 1, k) &\leq l && (17)\\
UML(k) &\leq min\{l' \perp CMLF(k), l \perp AMLF(m \perp 1, k)\} && (18)
\end{aligned}
$$

Inequalities (16) and(17) ensure the compliance of $CMLF$ and $AMLF$. Next synchronization granules have to be chosen to satisfy (18). Notice that (18) gives an upper bound on the unit mixing drifts at the succeeding slot. Any choice of component synchronization granules satisfying (18) ensures that synchronization content specifications are met for the succeeding slot of display.

3.5.2 Timing of Synchronization Granule Vectors

Schedules compliant with rate and drift profiles have some restrictions placed on them. The restrictions placed by rate profiles can be statically calculated, while restrictions placed by synchronization drift profiles at any slot depend on the history of unit synchronization drifts up to that slot. The rate and drift profiles play orthogonal roles in determining the compliance of a vector of schedules. The former determines an interval for positioning the scheduling point at any slot. The latter determines the variation between scheduling points across streams within the same slot. Thus, collectively they can be used to determine scheduling points for slots of synchronized schedules.

In order to determine intervals for positioning of scheduling points, consider a schedule $T = \{t_i(\cdot) : 1 \le i \le n\}$ for a collection of streams $S = \{s_i(\cdot) : 1 \le i \le n\}$, where $t_i(\cdot) = \{t_i(j) : 1 \le j\}$ gives the schedules for synchronization granules $\{s_i(j) : 1 \le j\}$ of stream $s_i(\cdot)$. For T to be compliant with the rate profile (ρ_{sy}, σ_{sy}), it must satisfy (19). In (19), $t_i(0) + \frac{k}{(\rho_{sy}+\sigma_{sy})}$ is the time to render k synchronization granules with a constant rate of $(\rho_{sy} + \sigma_{sy})$. The time to render k synchronization granules with a constant rate of $(\rho_{sy} \perp \sigma_{sy})$ is $t_i(0) + \frac{k}{(\rho_{sy}-\sigma_{sy})}$. Hence, (19) gives the interval of rendition for the k^{th} granule.

$$\forall i \le n : \forall k : t_i(k) \in [t_i(0) + \frac{k}{(\rho_{sy} + \sigma_{sy})}, t_i(0) + \frac{k}{(\rho_{sy} \perp \sigma_{sy})}] \tag{19}$$

The bounds in (19) are calculated using the largest and smallest rendition rates for T. Thus, the unit synchronization drift at any slot k, $USD(k)$ is bounded by:

$$(t_i(0) + \frac{k}{(\rho_{sy} \perp \sigma_{sy})}) \perp (t_i(0) + \frac{k}{(\rho_{sy} + \sigma_{sy})}) = \frac{2 \cdot k \cdot \sigma_{sy}}{(\rho_{sy} \perp \sigma_{sy}) \cdot (\rho_{sy} + \sigma_{sy})} \tag{20}$$

For T to be compliant with the drift profile $(ASDF, CSDF) = (d//m, d')$, it must satisfy some history based restrictions. To see how, define $Drift_{sy}(i) = \min\{d \perp USD_m(i), d' \perp CSD(i)\}$. Given a synchronization drift profile (d, d'), $Drift_{sy}(i)$ measures the the *left over slack* of the drift after the current slot i. The slack drift $Drift_{sy}(i)$ can be used up in the succeeding slot. Notice that $Drift_{sy}(i)$ satisfies $Drift_{sy}(i) \ge 0$ and $\{t_{i+1}(l) : 1 \le n\}$ must be chosen so that they satisfy (21).

$$USD(i + 1) \le Drift_{sy}(i) \tag{21}$$

3.5.3 An Example of Synchronous Scheduling

As an example, consider scheduling synchronized rendition of two streams $s_1(\cdot)$, $s_2(\cdot)$ with following parameters.

- A synchronization granule consists of *one* frame.
- Mixing profile is $(8//30, 6)$
- The rate profile is (33 frames/s, 10 frames/s) $= (\frac{1}{33}, \frac{1}{100})$ in frame/ms.
- Synchronization drift profile is $(100\ ms/30,\ 40\ ms)$.

Given the rate profile (33 frames/sec, 10 frames/sec), in an ideal rendition successive frame vectors should appear 33 milliseconds apart. Suppose when the scheduler is invoked for the 8^{th} time the history of schedules appears as given in Table 4.

According to the data given in Table 4, at slot 7, $UML(7)$ is 2, $AMLF(7)$ is 6 and $CMLF(7)$ is 3. Hence, according to (18), synchronization granules should be chosen so that $UML(8) \le \min$

Statistics	Slot Number = ideal synchronization granule	1	2	3	4	5	6	7
	Ideal time of appearance	0	33	66	99	132	165	198
Stream 1	Media granule	1	2	4	$\perp$	8	9	10
	Time of appearance	0	28	56	$\perp$	122	160	193
Stream 2	Media granule	1	4	5	6	8	10	12
	Time of appearance	0	38	66	109	142	160	203
Mixing	Unit mixing loss (UML)	0	2	1	$\perp$	0	1	2
	Aggregate mixing loss ($AMLF$)	0	2	3	3	3	4	6
	Consecutive mixing loss ($CMLF$)	0	2	3	$\perp$	0	1	3
Drift	Unit timing drift USD	0	10	10	$\perp$	20	0	10
	Aggregate synchronization drift (USD_{30})	0	10	20	20	40	40	50
	Consecutive synchronization drift (CSD)	0	10	20	$\perp$	20	0	10

Table 4: History of synchronization granule Rendition

$\{6 \perp CMLF(7), 8 \perp AMLF(7)\} = 2$. The choice of synchronization granules have to be so that the continuity profile of streams $s_1(\cdot)$ and $s_2(\cdot)$ are maintained and their indices differ at most by 2.

For the timing component, $USD(7)$, $USD_{30}(7)$ and $CSD(7)$ are respectively, 10, 50 and 10. Hence, the relative time drifts between the beginning of the eighth synchronization granules must not exceed min$\{100$ - $USD_30(7)$, 40 - $CSD(7)\} = 30$ milliseconds.

4 Relationship Between Continuity and Synchronization QoS Parameters

Two types of specifications must be satisfied in a synchronized rendition of a collection of CM streams. They are, synchronization parameters of a collection of streams and continuity parameters of their components. In the current section we investigate the definability of some of these parameters with respect to others. We show the following facts, and their stated consequences follow.

1. Mixing profiles of a collection of synchronized streams cannot be defined in terms of stream parameters of their components.
   Consequence: It is not possible to control the mixture of samples displayed together only by exercising control over individual streams, without having a mechanism to handle cross-stream effects.

2. Rate profiles of a collection of synchronized streams can be defined in terms of rate profiles of their components.
   Consequence: Rate of a synchronized rendition can be controlled by controlling rendition rates of its component streams.

3. Except for the perfect case, the synchronization drift profile of a collection of streams is not definable in terms of drift profiles of its components, although the aggregate synchronization drifts can be bounded by drift profiles of component streams.
   Consequence: It is possible to control average timing drifts in a synchronized rendition by controlling timing drifts of its component streams.

4. Consecutive synchronization drift of a collection of synchronized streams is not definable in terms of drift profiles of its component streams.

Consequence: It is not possible to control bursty timing drifts between a collection of synchronized streams by controlling the individual timing drifts of its component streams.

To state our results precisely, some notation is in order. Given any sequencing profile (ALF, CLF) for a CM stream, there is a set of (potentially infinite) sequences of synchronization granules that satisfy it, say $\mathcal{S}(ALF, CLF)$. Any member of $\mathcal{S}(ALF, CLF)$ is accepted as a presentation compliant with (ALF, CLF). For eg., streams $s_1(\cdot) = 1, 2, 3, 5, 7, 8, 10, 12, 13, 14, 16, 17$, $s_2(\cdot) = 1, 3, 5, 6, 7, 8, 10, 12, 14, 15, 16, 17$ and $s_3(\cdot) = 2, 4, 5, 6, 7, 8, 10, 12, 12, 15, 16, 17$ of Fig. 7 all belong to $\mathcal{S}(5//12, 2)$.

Similarly, for a given mixing profile $(AMLF, CMLF)$ and an integer n for the number of component streams, there is a set of sequences of n dimensional vectors of synchronization granules (one per each stream) that satisfy it, say $\mathcal{SS}(AMLF, CMLF)$. Any member sequence of $\mathcal{SS}(AMLF, CMLF)$ is acceptable as a synchronous rendition compliant with $(AMLF, CMLF)$. For eg., streams 1, 2 and 3 of Fig. 7 as a collection belong to $\mathcal{SS}(9//12, 6)$.

Given any drift profile (ADF, CDF), there is a set of (potentially infinite) sequences of rendition times (i.e. schedules) for synchronization granules that satisfy it, say $\mathcal{T}(ADF, CDF)$. For example, a stream $s_i(\cdot)$ given as a sequence of synchronization granules $\{s_i(j) : j \geq 1\}$ can have a schedule $t_i(\cdot) = \{t_i(j) : j \geq 1\}$ where synchronization granule $s_i(j)$ is rendered at time $t_i(j)$. Any rendition of a stream having a schedule in $\mathcal{T}(ADF, CDF)$ is accepted as a rendition compliant with (ADF, CDF). Similarly, for a given synchronization drift profile $(ASDF, CSDF)$ and an integer n for the number of streams, there is a set of n dimensional vectors of schedules for synchronization granules that satisfies it, say $\mathcal{TT}(ASDF, CSDF)$, where the i^{th} component of the schedule vector comprises a schedule for the i^{th} constituent stream. For eg, $[(1, 2, 3, \ldots), (1.1, 2.1, 3.1, \ldots)]$ and $[(1.5, 2.5, 3.5, \ldots), (1.6, 2.6, 3.6, \ldots)]$ are two vector schedules for a synchronous rendition of two streams, whereas $(1, 2, 3, \ldots)$ and $(1.1, 2.1, 3.1, \ldots)$ are schedules for components streams of the first synchronized rendition. As in the case of streams, any rendition having a schedule from $\mathcal{TT}(ASDF, CSDF)$ is considered to be compliant with $(ASDF, CSDF)$.

For a collection of synchronized streams, we need a notation for vector schedules. Towards that end, let $\{A_i : 1 \leq i \leq n\}$ be any finite collection of sets of sequences. Then define the product $\prod_{i=1}^{n} A_i = \{(t_i, \ldots, t_n) : t_i \in A_i\}$. For eg., if $A_1 \in \mathcal{S}(5//10, 3)$ and $A_2 \in \mathcal{S}(4//15, 2)$, then $\prod_{i=1}^{2} A_i$ is the vector of synchronization granules, where each i^{th} stream in the vector satisfies continuity specifications of stream i.

4.1 Non Definability of Mixing Profiles

In this section we show that, in general, the mixing profile of a collection of CM streams is non-definable and unspecifiable in terms of sequencing profiles of its constituent streams. Our claim of non-definability, precisely stated in Theorem 1, says that any mixing synchronization profile (other than perfect synchronization) cannot be obtained by simply specifying sufficiently stringent sequencing profiles for its component streams.

Theorem 1 (Non Definability of Mixing Profile) *For any given mixing profile* $(AMLF, CMLF)$ $\neq (0//m, 0)$ *and any integer* n *for the number of component streams, there do not exist* n *sequencing profiles* $\mathcal{S}(ALF_i, CLF_i)$ *of CM streams that satisfy (22); i.e. no mixing synchronization profile (other than the perfect one) can be obtained by only specifying sequencing profiles for component streams.*

$$\mathcal{SS}(AMLF, CMLF) = \prod_{i=1}^{n} \mathcal{S}(ALF_i, CLF_i) \tag{22}$$

Proof:
In order to justify our claim, consider synchronizing two streams $s_1(\cdot)$ and $s_2(\cdot)$ with a mixing profile of $(AMLF, CMLF) = (p//q, r)$, where $p, r \geq 1$. Suppose contrary to our claim, there are sequencing profiles (ALF_1, CLF_1) and (ALF_2, CLF_2) satisfying

$$\mathcal{SS}(AMLF, CMLF) = \prod_{i=1}^{2} \mathcal{S}(ALF_i, CLF_i) \tag{23}$$

Then, it is not the case that $CLF_1 = 0$ and $CLF_2 = 0$, because, if so, then by (23), $CMLF = 0$. Without loss of generality, assume $CLF_1 \geq 1$. Then (s_1, s_2) where $s_1(\cdot) = \langle 1, 2, 3, 4, 5, \ldots \rangle$ and $s_2(\cdot) = \langle 2, 3, 4, 5, \ldots \rangle$ belongs to $\prod_{i=1}^{2} \mathcal{S}(ALF_i, CLF_i)$, but not to $\mathcal{SS}(AMLF, CMLF)$. $s_1(\cdot) \prod s_2(\cdot) \notin \mathcal{S}(AMLF, CMLF)$ because the unit mixing loss at every slot i, $UML(i)$ is 1, and hence adds up to be more than $CMLF$. See Fig. 9. Furthermore, the preceding example shows that the only definable subclass of $\mathcal{SS}(AMLF, CMLF)$ is $\{ \langle 1, 2, 3, 4, \ldots \rangle \prod \langle 1, 2, 3, 4, \ldots \rangle \}$, thus showing the non-definability of any non trivial subclass thereof. ■

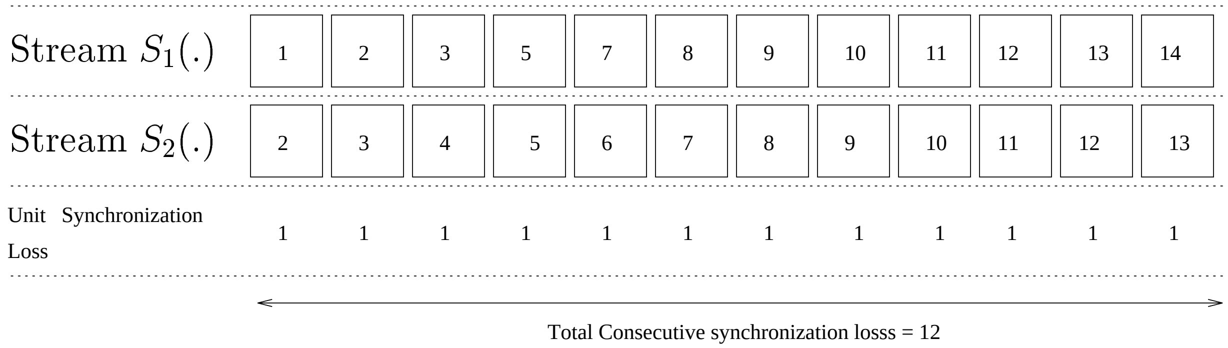

Figure 9: Non Definability of $AMLF$ and $CMLF$

4.2 Definability of Rate Profiles for Synchronized Streams

Synchronizing CM streams requires synchronization granularities and rendition rates to be related. Unlike mixing profiles, rate profile of a collection of synchronized streams is definable in terms of rate profiles of its component streams. Definability of rate profiles is a consequence of the fact that *in order to synchronize, component streams must display the same number of synchronization granules in a given interval of time.* This requirement is precisely stated in Theorem 2. In addition, rendition rate and its variation for a synchronized collection can be computed from rate profiles of their component streams, as shown in Theorem 2.

Theorem 2 (Definability of Rate Profiles) *Let $S = \{s_i(\cdot) : i \leq n\}$ be a collection of streams.*

1. *A necessary and sufficient condition for S to be synchronized is given by $\forall i, j : \rho_i.g_i = \rho_j.g_j$ where stream $s_i(\cdot)$ has a rate profile (ρ_i, σ_i) and synchronization granularity g_i.*
2. *The maximum drift of any synchronization granule of $s_i(\cdot)$ is $\frac{g_i.\sigma_i}{\rho_i.(\rho_i - \sigma_i)}$.*

Proof:

1. Rendition rate of synchronization granules of stream $s_i(\cdot)$ is $\rho_i.g_i$. Because all media granules from streams of S have equal rendition times, $\forall i, j : \ \rho_i.g_i = \rho_j.g_j$ holds. This is the rendition rate, say ρ_{sy}, for synchronization granules.
2. To calculate the rate variation, σ_{sy} of $s_i(\cdot)$ with synchronization granularity g_i, consider the last (g_i^{th}) media granule of a synchronization granule. It can drift from the ideal rendition time of $\frac{g_i}{\rho_i}$ by max $\{\frac{g_i}{\rho_i} \perp \frac{g_i}{(\rho_i+\sigma_i)}, \frac{g_i}{(\rho_i-\sigma_i)} \perp \frac{g_i}{\rho_i}\} = \frac{g_i.\sigma_i}{\rho_i.(\rho_i-\sigma_i)}$. Thus, we take the maximum drift of a synchronization granule of $s_i(\cdot)$ to be max $\{\frac{g_i.\sigma_i}{\rho_i.(\rho_i-\sigma_i)} : 1 \leq i \leq n\}$ (say σ_{sy}). Then the rate profile for the synchronized streams is (ρ_{sy}, σ_{sy}). See Fig. 10.

■

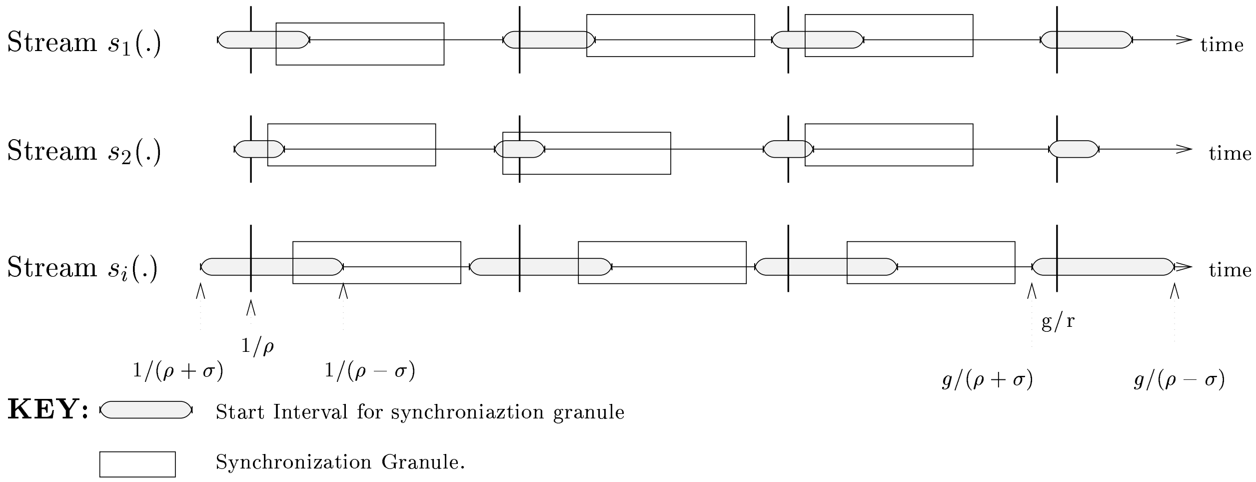

Figure 10: Definability of Rate Profiles of Synchronization Granules

4.3 Non Definability of Synchronization Drift Profiles

In this section we show that, in general, synchronization drift profiles cannot be defined only in terms of drift profiles of their component streams. This is a consequence of two facts. The first being that because drift profiles for component streams limit synchronization granules drifting by themselves, the unit synchronization drift, USD at any slot can be controlled by choosing sufficiently stringent ADF's. The second fact is that $CSDF$'s cannot be bounded by choosing sufficiently stringent ADF's or CDF's for component streams. Firstly, we justify the claim of non definability of synchronization drift profiles and secondly, show a method of estimating ASDs. Finally, we show the non definability of $CSDF$, despite the ability to compute a bound for $ASDF$.

4.3.1 Non Definability of $(ASDF, CSDF)$

In this section we show that the drift profile of a synchronized rendition of a collection of streams cannot be defined only by means of drift profiles of its component streams, without reference to any cross effects. The precise statement of the non definability of $(ASDF, CSDF)$ is given in Theorem 3.

Theorem 3 (Non Definability of ($ASDF, CSDF$)) *Given a schedule for synchronized renditions $\mathcal{TT}(ASDF, CSDF)$ and an integer n for the number of streams, there do not exist drift profiles (ADF_i, CDF_i) for n streams such that (24) holds.*

$$\mathcal{TT}(ASDF, CSDF) = \prod_{i=1}^{n} \mathcal{T}(ADF_i, CDF_i) \tag{24}$$

i.e. merely specifying drift profiles of component streams is insufficient to define the drift profile of a synchronized collection of streams.

Proof:
In order to justify our claim, suppose that for any given $\mathcal{TT}(ASDF, CSDF) = (d//m, d')$ where $d, d' > 0$, there is a set of schedules $T = \{t_i(\cdot) : 1 \leq i \leq n\}$ for a set of streams $S = \{s_i(\cdot) : n \geq 1\}$ with drift profiles $\{(ADF_i, CDF_i) : i \leq n\}$ satisfying (24). i.e. assume that the given synchronized drift profile can be specified by only specifying T. Assume that $(ADF_i, CDF_i) = (d_i//m_i, d'_i)$, and that the synchronization granularity of each $s_i(\cdot)$ is 1. (24) implies that there is some integer $l \leq n$ such that $d_l, d'_l > 0$, i.e. the drift profile of at least one stream is non-perfect. Consider a set of schedules $t_i(\cdot) = \{t_i(j) : 1 \leq j\}$ defined by (25) and (26), where

$$\begin{aligned} t_i(j) &= (\sum_{k=1}^{n} d'_k) + 1 \text{ for } j = 1 \text{ and } 1 \leq i \leq n && (25) \\ &= (j \perp 1).g \text{ for } j > 1 \text{ and } 1 \leq i \leq n && (26) \\ & \quad \text{where } g \text{ is the ideal rendition time for a synchronization granule} \end{aligned}$$

Notice that given schedules $\{t_i(\cdot) : i \geq 1\}$ are constructed as follows. The first synchronization granules of all n streams are late by more than their ADF's. The other synchronization granules are exactly on time. Then $(t_i(\cdot), \ldots, t_n(\cdot))$ belongs to $\mathcal{TT}(ASDF, CSDF)$, but not to $\prod_{i=1}^{n} \mathcal{T}(ADF_i, CDF_i)$. The reasons being, that because relative drifts of synchronization granules are zero $(t_i(\cdot), \ldots, t_n(\cdot)) \in \mathcal{T}(ASDF, CSDF)$, but because the drift of the first synchronization granule of any stream s_i is larger than d'_i, $t_i(\cdot) \notin \mathcal{T}(ADF_i, CDF_i)$. ■

4.3.2 Controlling Aggregate Synchronization Drifts

A CM stream with a given continuity profile and synchronization granularity can be envisioned as a flow of synchronization granules by packaging the appropriate number of media granules into a synchronization granule. We calculate the drifts of such *logically packaged* synchronization granules in terms of the drift profile of their media streams. Estimating synchronization drifts is handled in two steps. Firstly, in Lemma 1 we find an upper bound on the maximum permissible delay between two *successively packaged* synchronization granules. Using this bound, in Theorem 4, we find an upper bound on $ASDF$ of a collection of streams in terms of ADF's of streams in the collection.

Lemma 1 (Upper Bound for Synchronization Drifts) *For stream $s_i(\cdot)$ with a rate profile (ρ_i, σ_i) drift profile $(d_i, d'_i) = (l_i//m_i, l'_i)$ and synchronization granularity g_i, the maximum delay between two successive synchronization granules is max$\{\frac{g.\sigma}{\rho_i.(\rho_i - \sigma_i)}, d_i.\frac{[g_i/m_i]}{\lceil m_i/g_i \rceil}\}$.*

Proof:
To calculate drifts in synchronization granules, consider the following two cases:
Case 1: ($g_i > m_i$)

Then, a synchronization granule of $s_i(\cdot)$ contains $\lfloor \frac{g_i}{m_i} \rfloor$ units of m_i media granules. In each such unit there can be a maximum drift of $d_i.\lfloor \frac{g_i}{m_i} \rfloor$. But, the additional fraction of a unit (if there is one) can suffer a maximum aggregate drift of d_i. Consequently, the beginning of the next synchronization granule suffers a maximum drift of $d_i.\lceil \frac{g_i}{m_i} \rceil$.

Case 2: ($g_i < m_i$)

Thus, a unit of g_i media granules contain $\lfloor \frac{m_i}{g_i} \rfloor$ synchronization granules. If each synchronization granule is allowed a drift of d''_i, then the total aggregate drift that can be experienced by the $(g_i+1)^{th}$ granule is as large as $d''_i.\lfloor \frac{m_i}{g_i} \rfloor$, plus one more unit of d''_i in the fraction of a unit (if there is one). Thus, we get a total drift of $d''_i.\lceil \frac{m_i}{g_i} \rceil = d$. Consequently, the maximum allowable drift is $d''_i = \frac{d_i}{\lceil m_i/g_i \rceil}$.

Thus, in both cases (that is $g_i < m_i$ and $g_i > m_i$) the maximum drift is $d_i.\frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}$. In Sect. 4.2 it was shown that, in order to be compliant with the rate profile (ρ_i, σ_i) of stream $s_i(\cdot)$, the maximum drift between synchronization granules is $\frac{g_i.\sigma_i}{\rho_i.(\rho_i - \sigma_i)}$. Consequently, the maximum permissible delay between two successive synchronization granules is $\max\{\frac{g_i \cdot \sigma}{\rho_i.(\rho_i - \sigma_i)}, d_i.\frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\}$. ■

Using results from Lemma 1, Theorem 4 computes an upper bound for aggregate synchronization drifts.

Theorem 4 (Upper Bound on Aggregate Synchronization Drift) *The aggregate synchronization drift factor (ASDF = d//m) of a set of streams $S = \{s_i(\cdot) : i \le n\}$ satisfies the bound:*

$$d \le 2 \cdot m \cdot max\{max\{\frac{g_i \cdot \sigma_i}{\rho_i.(\rho_i \perp \sigma_i)}, d_i.\frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\} : i \le n\} \tag{27}$$

In (27), m_i is the synchronization granularity and $(l_i//m_i, l'_i)$ is the drift profile of stream $s_i(\cdot)$.

Proof:

To compute a bound for $ASDF$ of S by choosing sufficiently stringent ADF's, suppose that there are $T = \{t_i(\cdot) : 1 \le n\}$ schedules for n streams $S = \{s_i(\cdot) : 1 \le i \le n\}$, where $t_i(j)$ is the schedule of $s_i(j)$. Then the total drift for m successive synchronization granules is given by $\sum_{k=1}^{m} max\{\|t_i(k) \perp t_j(k)\| : 1 \le i, j \le n\}$. This bound can be estimated as follows: In the following inequalities, $u_i(\cdot) = \{u_i(j) : 1 \le j\}$ is the *ideal schedule* for synchronization granules $s_i(\cdot) = \{s_i(j) : 1 \le j\}$ with a rate profile (ρ_{sy}, σ_{sy}), i.e. one that satisfies $u_i(j+1) = u_i(j) + 1/\rho_i$ for all $i \le n$ and $j \ge 1$. Because of the metric inequality $\|t_i(k) \perp t_j(k)\| \le \|t_i(k) \perp u_i(k)\| + \|u_i(k) \perp u_j(k)\| + \|u_j(k) \perp t_j(k)\|$, we get (28). The purpose of (28) is to compute upper bounds for drifts of k^{th} synchronization granules of streams $s_i(\cdot)$ and $s_j(\cdot)$ by relating them to their ideal schedules $u_i(k)$ and $u_j(k)$.

$$\begin{aligned} &\sum_{k=1}^{m} max\{\|t_i(k) \perp t_j(k)\| : 1 \le i, j \le n\} \le \\ &\quad \sum_{k=1}^{m} max\{\|t_i(k) \perp u_i(k)\| + \|u_i(k) \perp u_j(k)\| + \|u_j(k) \perp t_j(k)\| : 1 \le i,\ j \le n\} \end{aligned} \tag{28}$$

Because both $u_i(k)$ and $u_j(k)$ are ideal schedules for k^{th} synchronization granules of two synchronized streams $s_i(\cdot)$ and $s_j(\cdot)$ starting at the same time and rendering at the same rate, we get $u_i(k) = u_j(k)$. Hence (28) reduces to (29).

$$\sum_{k=1}^{m} max\{\|t_i(k) \perp t_j(k)\| : 1 \leq i,j \leq n\} \leq \sum_{k=1}^{m} max\{\|t_i(k) \perp u_i(k)\| + \|u_j(k) \perp t_j(k)\| : 1 \leq i,j \leq n\} \tag{29}$$

From the proof of Lemma 1, maximum drift between synchronization granules of stream $s_i(\cdot)$ is $\max\{\frac{g_i \cdot \sigma_i}{\rho_i \cdot (\rho_i - \sigma_i)}, d_i \cdot \frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\}$. We get (30) by substituting $\max\{\frac{g_i \cdot \sigma_i}{\rho_i \cdot (\rho_i - \sigma_i)}, d_i \cdot \frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\}$ and $\max\{\frac{g_i \cdot \sigma_i}{\rho_j \cdot (\rho_j - \sigma_j)}, d_j \cdot \frac{\lceil g_j/m_j \rceil}{\lceil m_j/g_j \rceil}\}$ respectively for $\|t_i(k) \perp u_i(k)\|$ and $\|u_j(k) \perp t_j(k)\|$ in (29).

$$\begin{aligned} &\sum_{k=1}^{m} max\{\|t_i(k) \perp t_j(k)\| : 1 \leq i,j \leq n\} \leq \\ &\quad \sum_{k=1}^{m} max\{max\{\frac{g_i \cdot \sigma_i}{\rho_i.(\rho_i \perp \sigma_i)}, d_i.\frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\} + max\{\frac{g_j \cdot \sigma_j}{\rho_j.(\rho_j \perp \sigma_j)}, d_j.\frac{\lceil g_j/m_j \rceil}{\lceil m_j/g_j \rceil}\} : i,j \leq n\} \end{aligned} \tag{30}$$

Inequality (30) simplifies to (31), justifying our claimed upper bound in (27).

$$\sum_{k=1}^{m} max\{\|t_i(k) \perp t_j(k)\| : 1 \leq i,j \leq n\} \leq 2 \cdot m \cdot max\{max\{\frac{g.\sigma}{\rho_i.(\rho_i \perp \sigma_i)}, d_i.\frac{\lceil g_i/m_i \rceil}{\lceil m_i/g_i \rceil}\} : i \leq n\} \tag{31}$$

■

Theorem 4 shows that $ASDF$ of a set of synchronized stream can be bounded by choosing sufficiently small ADF's for component CM streams. Here the $ASDF$ can be made smaller by choosing rate variations σ_i and aggregate drifts d_i for component stream $s_i(\cdot)$ to be as small as required. However, it should be noted that the bound given in Theorem 4 does not imply definability in the sense of Sect. 4.1.

4.3.3 Non Definability of Consecutive Synchronization Drifts

In this section, we show that the consecutive synchronization drifts, (i.e. $CSDF$'s) of a collection of synchronized streams cannot be specified in terms of drift profiles of its component streams. The precise statement of our claim is given in Theorem 5. In order to state our claim precisely, let $\mathcal{TT}(CSDF)$ be the collection of n streams that satisfy a consecutive synchronization drift factor of $CSDF$.

Theorem 5 (Non Definability of Consecutive Synchronization Drifts) *Given $\mathcal{TT}(CSDF)$ and an integer n for the number of component streams, there do not exist drift profiles (ADF_i, CDF_i) for n CM streams that satisfy (32).*

$$\mathcal{TT}(CSDF) = \prod_{i=1}^{n} \mathcal{T}(ADF_i, CDF_i) \tag{32}$$

Proof:
To show the non definability of consecutive synchronization drifts of synchronized schedules in terms of drift profiles of their component streams, suppose that we have 3 streams $s_i(\cdot)$ for $1 \leq i \leq 3$, (i.e. $n = 3$) with drift profiles $(ADF_i, CDF_i) = (d_i//m_i, d_i')$. Thus, the maximum drift allowed

by each $s_i(\cdot)$ is d_i units for every m_i synchronization granules. Without loss of generality assume $m_i > 3$. Then consider the three schedules given as follows. In the schedule $t_i(\cdot)$ of $s_i(\cdot)$, every third synchronization granule suffers a drift of $min\{d_i, d'_i : 1 \leq i \leq 3\}/(m_1 + m_2 + m_3)$, say D, and all others suffer a zero drift. In the terminology of Sect. 4.3.2, $t_i(\cdot)$ can be given as:

$$t_i(j) = u_i(j) + D \text{ if } j = 3 \cdot k + i \text{ for some } k \tag{33}$$

$$t_i(j) = u_i(j), \text{ if not} \tag{34}$$

These schedules are shown in Fig. 11. Hence, *unit synchronization drift at every slot* j, say $USD(j)$, is $max\{\|t_i(j) \perp t_k(j)\| : 1 \leq i, k \leq 3\} = D$. Thus, $\sum_{j=1}^{k} USD(j) = D \cdot k$. Consequently, $\sum_{j=1}^{k} USD(j)$ can be made as large as possible by increasing k. Hence, $\langle t_1(\cdot), t_2(\cdot), t(\cdot)\rangle \not\in \mathcal{T}\mathcal{T}(CSDF)$ for any $CSDF$. Nevertheless, $CDF_i = D < d'_i$ and aggregate drift per m_i media granules in schedule $t_i(\cdot)$ is bounded above by $D.m_i < \frac{d_i}{m_i}$. Consequently, $\langle t_1(\cdot), t_2(\cdot), t(\cdot)\rangle \in \prod_{i=1}^{3} T(CDF_i)$, justifying the non-definability of $\mathcal{T}\mathcal{T}(CSDF)$ as claimed. ■

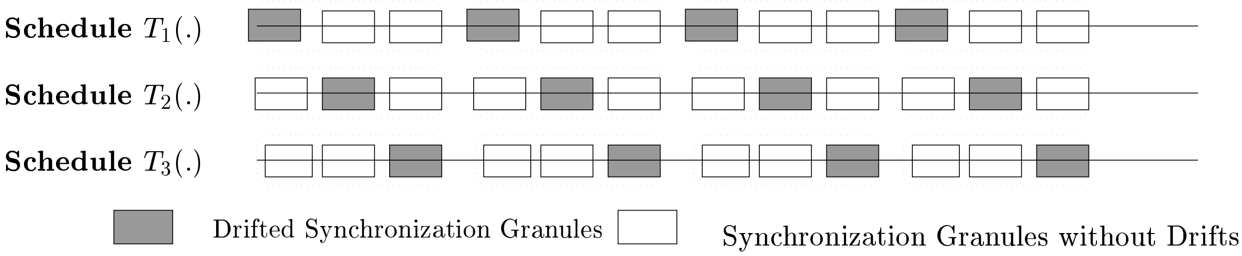

Figure 11: Non Definability of Cumulative Synchronization Drifts

5 Integrated Scheduling

Up to now we have discussed ways of specifying continuity and synchronization requirements of continuous media streams and how to schedule streams compliant with such requirements individually. The current section discusses how such specifications can be used in integrated scheduling; i.e. scheduling a collection of synchronized streams satisfying all specified continuity and synchronization specifications.

As discussed earlier, there is a class of CM streams that satisfy a given continuity specification. Similarly, for any given number of component streams and a synchronization specification, there is a set of vectors of CM streams that satisfy it. Consequently, there is a large design space from which to choose a rendition of media streams to satisfy a given set of continuity and synchronization specifications. We describe that space in the first part of this section and present scheduling policies and algorithms to meet specific choices in the second part.

5.1 Design Space

The design space for integrated scheduling consists of two components. They are the content component and the timing component. As discussed earlier, for a synchronized rendition, the content component has to satisfy synchronization specifications of the collection and continuity specifications of its component streams. In order to observe the combined effect of these two classes of requirements, we want to schedule a collection of streams $S = \{s_i(\cdot) : 1 \leq i \leq n\}$.

Further suppose that S has to satisfy mixing, rate and drift profiles of $(AMLF, CMLF)$, (ρ_{sy}, σ_{sy}) and $(ASDF, CSDF)$, respectively. Also the rendition of each stream $s_i(\cdot)$ must satisfy continuity, rate and drift profiles of (ALF_i, CLF_i), (ρ_i, σ_i) and (ADF_i, CDF_i) respectively. Consequently, assuming the availability of synchronization granules at the display site, content component of the design space consists of n-dimensional vectors of synchronization granules that belong to the set of streams given in (35).

$$\mathcal{SS}(AMLF, CMLF) \cap \prod_{i=1}^{n} \mathcal{S}(ALF_i, CLF_i) \tag{35}$$

The content component of our design space consists of streams for synchronization granule vectors from (35). To satisfy rate and drift profiles, they must be displayed in n-dimensional vectors of timing schedules given in (36).

$$\mathcal{TT}(ASDF, CSDF) \cap \prod_{i=1}^{n} \mathcal{T}(ADF_i, CDF_i) \tag{36}$$

Accordingly, the schedulers we propose consist of two components: the content selection component and the timing selection component. For stream rendition to be compliant with synchronization and continuity specifications, any potential content selecting component of a scheduler has to extend a string of vectors belonging to the set in (35) to a longer string belonging to the same set. Details of the content selecting component are presented in Sect. 5.2. Similarly, the timing selecting component of any potential scheduler has to extend a string of time vectors in the set in (36) to a longer string in the same set. Details of the timing selection component are presented in Sect. 5.3.

Consequently, the task of any schedulers is, given any string of schedules that satisfy some specification, to select an extension for it to satisfy the same specification. Once the scheduler selects the extended string of schedules, it is the task of the underlying delivery system to make sure that the selected string of synchronization granules are available at the display site at the appropriate time. Conversely, when the scheduler seeks extensions of strings of schedules, it may look for only those extensions consisting of strings that are already available at the display site. These correspond to different policies. Also an issue is the actions taken when the given strings of schedules cannot be extended to strings that satisfy requested specifications. Thus, there are major issues with the policy used to make sure that appropriate synchronization granules are available, and recovery policies in case of starvation.

5.2 Continuity and Synchronization Parameters

The space of possible schedules compliant with a given continuity profile as discussed in Sect. 3.5.1, consists of a *best next media granule* and an interval of possible media granules. Recall also that the best next media granule is the one that, if chosen to be displayed, results in the least amount of unit continuity loss.

Similarly, for a given synchronization specification, Sect. 3.5.1 gives the *best next vector of synchronization granules* and a range of possible synchronization granules. Consequently, to be compliant with all synchronization and continuity specifications, the solution spaces provided in Sects. 3.5.1 and 2.5.1 need to be incorporated into one integrated schedule.

In finding an integrated schedule, a problem faced in using the solution given for continuity requirements in Sect. 2.5.1 is that the best granule and the interval of next possible granules were computed for media granules. We revise those calculations in Sect. 5.2.1 to be applicable to synchronization granules.

5.2.1 Sequencing Profile: Revision for Synchronization Granules

Suppose we want to schedule a CM stream $s_n(\cdot)$ with a sequencing profile $(ALF, CLF) = (l//m, l')$ and media granularity g. Following notation from Sect. 2.5.1, let $D(k,p)$ be the sum of unit continuity losses within a window of p media granules at slot k. Hence, the next synchronization granule can be skipped if and only if $D(k, m \perp 1) + g < l$. Generalizing, for $s_n(\cdot)$ to satisfy an ALF of $l//m$, the maximum number of synchronization granules that can be skipped, paused or omitted is $\lfloor (l \perp D(k, m \perp 1))/g \rfloor$. Let $C(k)$ be the sum of consecutive non zero unit continuity losses at slot k. To satisfy a CLF of l', the maximum number of media granules that can be skipped for the succeeding slot is $l' \perp C(k)$, hence $\lfloor (l' \perp C(k))/g \rfloor$, synchronization granules. Thus, to satisfy both CLF and ALF specifications, the number of granules that can be skipped, paused, or missed is $\max\{min\{\lfloor (l' \perp C(k))/g \rfloor, \lfloor (l \perp D(k, m \perp 1))/g \rfloor\}, 0\}$, say $L(k)$.

Accordingly, the valid interval for synchronization granules to be displayed at slot k is given by (37) and (38).

$$\{\perp\} \cup [q, q + L(k)] \text{ if } L(k) > 0 \tag{37}$$

$$[q+1, q+1] \text{ if } L(k) = 0 \tag{38}$$

However, the synchronization granule that has the *best* sequencing properties (i.e. the one that contributes to the least loss factors) is calculated as follows: If there is a synchronization granule $s(k)$ displaying at slot k, then the best synchronization granule to show at slot $s(k+1)$ is $s(k)+1$. If not, and $s(j)$ is the last displayed synchronization granule at slot $s(i \perp p)$ then $s(j + i \perp p)$ is the best choice for the synchronization granule.

5.2.2 Visualizing the Design Space

Consider the synchronous rendition of a collection of streams $S = \{s_i(\cdot) : 1 \leq i \leq n\}$. Suppose S has to satisfy a mixing profiles of $(AMLF, CMLF)$ and each stream $s_i(\cdot)$ has to satisfy the continuity profile (ALF_i, CLF_i). Further suppose that at slot k of a synchronized display each stream $s_i(\cdot)$ of S is rendering synchronization granule $s_i(k_i)$. In order for the display to be compliant with mixing and sequencing profiles, the choices for the succeeding vector of synchronization granules $s_i((k+1)_i)$ have to satisfy (37) ,(38) and (18). Notice that (37) and (38) provide intervals for possible synchronization granules.

For example, the situation for two streams can be visualized as in Fig. 12. Consider two streams $s_1(\cdot)$ and $s_2(\cdot)$ that have synchronization granules $s_1(k_1)$ and $s_2(k_2)$ displaying at slot k. Suppose that the intervals calculated from (37) and (38) for $s_1(\cdot)$ and $s_2(\cdot)$ are $[A_1, B_1]$ and $[A_2, B_2]$ respectively. Hence the possible space for the next pair of synchronization granules say, $(s_1((l_1), s_2(l_2))$ compliant with continuity profiles is the hyper-rectangle $[A_1, B_1] \times [A_2, B_2]$. Let us call this the *continuity content space.* Assume that the largest value of $UML(k)$ that satisfies (18) is ϵ. The latter restriction places the requirement $\|s_1(l_1) \perp s_2(l_2)\| \leq \epsilon$. That means they should be at most distance ϵ apart from each other, i.e. the solution space for $(s_1(l_1), s_2(l_2))$ consists of a strip centered around the diagonal $s_1(\cdot) = s_2(\cdot)$ with width 2ϵ. Let us call this the *synchronization content space.* Hence the solution space (if it exists) compliant with both continuity and mixing profiles is the intersection of the continuity content space with the synchronization content space; i.e the intersection of the rectangle $[A_1, B_1] \times [A_2, B_2]$ with the strip. This is shown shaded in Fig. 12 as the *design content space.* Also, for each stream $s_1(\cdot)$ and $s_2(\cdot)$, there is the *best choice* for a synchronization granule. The values of best choices for $s_1(\cdot)$ and $s_2(\cdot)$ are shown respectively as horizontal and vertical lines. Notice that best choices always intersect and the point of intersection

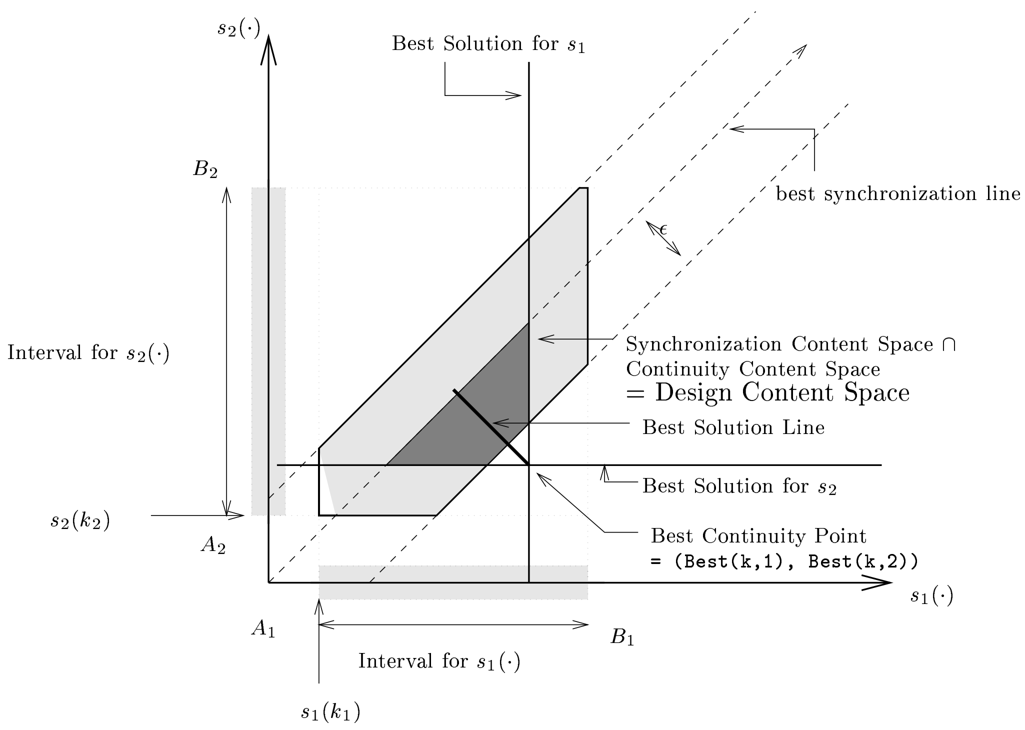

Figure 12: Design Space of Continuity and Mixing Profiles

is always inside the hyper-rectangle $[A_1, B_1] \times [A_2, B_2]$. Call the point of intersection of the best synchronization granules the *best continuity point*. A potential problem is that the best continuity point may not be within the synchronization content space. The importance of the best continuity point is that it gives the least CLF's and ALF's and hence the best continuity for the component streams. The best synchronization point for the rendition lies on the diagonal line $s_i(\cdot) = s_2(\cdot)$. Hence one choice for a potential scheduling algorithm is to drop the perpendicular from the best continuity point to the best synchronization line and choose the point nearest to the best continuity point on this perpendicular that lies within the synchronization content space.

Abstracting out from the example above, using (37) and (38) for each stream, we can compute an interval for choosing synchronization granules satisfying its continuity requirements. The product space of these intervals gives a hyper-rectangle from which to choose any combination of synchronization granules that satisfy the continuity requirements of component streams. We call this the *continuity content space*. Given the synchronization granules that are being displayed at any slot, from (18) we can compute ($UML(k)$ for slot k in (18)) the maximum difference between sequence numbers of any two synchronization granules to be displayed for different streams at the succeeding slot. Thus, the space of all synchronization granules satisfying (18) forms a cylinder centered around the main diagonal with radius $UML(k)$. We call this space the *synchronization content space*. Any vector of synchronization granules chosen from the synchronization content space satisfies the mixing profile of the collection. Consequently, in order for the next vector of synchronization granules to satisfy continuity profiles of component streams and the mixing profile of the collection it must be chosen from the intersection of the continuity content space and the synchronization content space, which we call the *design content space*. The algorithm presented in Sect. 5.2.3 makes such a choice.

5.2.3 Algorithm for Content Selection

To state a content selection algorithm, let `Sync(k)` be the largest value of *UML(k)* satisfying (18) at the k^{th} slot that the scheduler is invoked. For each stream i, let $[A_i(k), B_i(k)]$ be the k^{th} possible synchronization granule interval calculated in (37) and(38). Suppose that `Best(k,i)` is the best synchronization granule for stream i at slot k.

The algorithm that we present favors synchronization more than continuity requirements, i.e. it attempts to find the vector of synchronized granules with the least unit mixing loss. Our algorithm uses the auxiliary function `find(`$\vec{r_n}, \epsilon$`)`, which computes a sequence of integral coordinate points in a n-dimensional sphere with center $\vec{r_n}$ and radius ϵ, starting at the center and going towards the surface. Here, ϵ is a real number and we assume that the sequence `find(`$\vec{r}, \epsilon$`)` has length $\|find\|$. The algorithm is as follows:

Algorithm 1 Scheduling: Contents at slot k

```
▷ To Compute the $k^{th}$ synchronization granule vector
$\vec{r_n} = (\sum_{i=1}^{n} Best(k,i)/n, \ldots, \sum_{i=1}^{n} Best(k,i)/n)$
for each($\vec{z} = (z_1, \ldots, z_n) \in$ find($\vec{r_n}, \epsilon$))
    if $\vec{z} \in \prod_{i=1}^{n} [A_i(k), B_i(k)]$ and available at display site,
        display the synchronization granule $z_i$ for stream $s_i(k)$
    end If
end for each
if no synchronization granule found yet
    m = min$\{A_i(k) : 1 \leq i \leq n\}$
    M = max$\{A_i(k) : 1 \leq i \leq n\}$
    for($\vec{r} = (m, \ldots, m)$; $r_i = r_i + 1$; $\vec{r} = (M, \ldots, M)$ )
        for($\vec{z} = (z_1, \ldots, z_n) \notin$ find($\vec{r_n}, \epsilon$) and $\vec{z} \in \prod_{i=1}^{n} [A_i(k), B_i(k)]$)
            if ($\vec{z}$ is available at display site)
                display the synchronization granule $z_i$ for stream $s_i(k)$
            end if
        end for
    end for
end if
else /* no synchronization granule available */
    nothing available to be displayed
end
```

Explanation

(a) Search for a synchronization granule from the center going towards the surface on the best line. If found display it.

(b) Else search for *any* synchronization granule from the design content space. If found display it.

(c) Else nothing available to be displayed.

5.3 Selecting Timing Parameters

The timing parameters have properties similar to those of content parameters. For each component stream $s_i(\cdot)$ of a collection of synchronized streams $S = \{s_i(\cdot) : 1 \leq i \leq n\}$, rate and drift profiles of $s_i(\cdot)$ need to be satisfied, and as a collection the synchronization drift parameters of S have to be maintained. As described in Sect. 2.5.2, to be compliant with the drift profile of $s_i(\cdot)$, at each slot the timing component of our scheduler is invoked, it has a choice of picking a time from an interval of possible values as in (5). Also based on the rate profile (ρ_i, σ_i), the *best time* to present the k^{th} synchronization granule of stream $s_i(\cdot)$ is $1/\rho_i$ time units after the beginning of the $(k \perp 1)^{th}$ synchronization granule. Consequently, for each stream $s_i(\cdot)$, there is an interval $[U_i(k), V_i(k)]$ and a best time $t_i(k)$ to display the k^{th} synchronization granule. Thus, the design space of schedules that satisfy rate requirements of all streams of S forms a hyper-rectangle in the space of scheduling points, which we call the *continuity timing space.*

Based on the synchronization drift profiles and the history of unit synchronization drifts up to the display of the k^{th} synchronization granule, the difference between display times of any synchronization granules of component streams are bounded by $Drift_{sy}(k)$, as calculated in (21). Thus, analogous to the design space of synchronization content space, scheduling time points compliant with synchronization specifications form a cylinder with radius $Drift_{sy}(k)$ around the *ideal time axis* $t_1(\cdot) = \ldots = t_n(\cdot)$. We call this the *synchronization timing space.* The intersection of the continuity timing space and the synchronization timing space is called the *design timing space.*

Thus, a similar figure can be used to visualize timing points to schedule synchronization granules. The updated figure, with required revisions is given in Fig. 13. But the main difference between the content selection algorithm and the time selection algorithm is that in the successive choices of content, some indices of components of vectors of synchronization granules may decrease. But that cannot be allowed in the time selection algorithm, as going back in time cannot be achieved. Hence the traversal of the design timing space has to be done in such a way such that all components of successive choices for time vectors are non-decreasing.

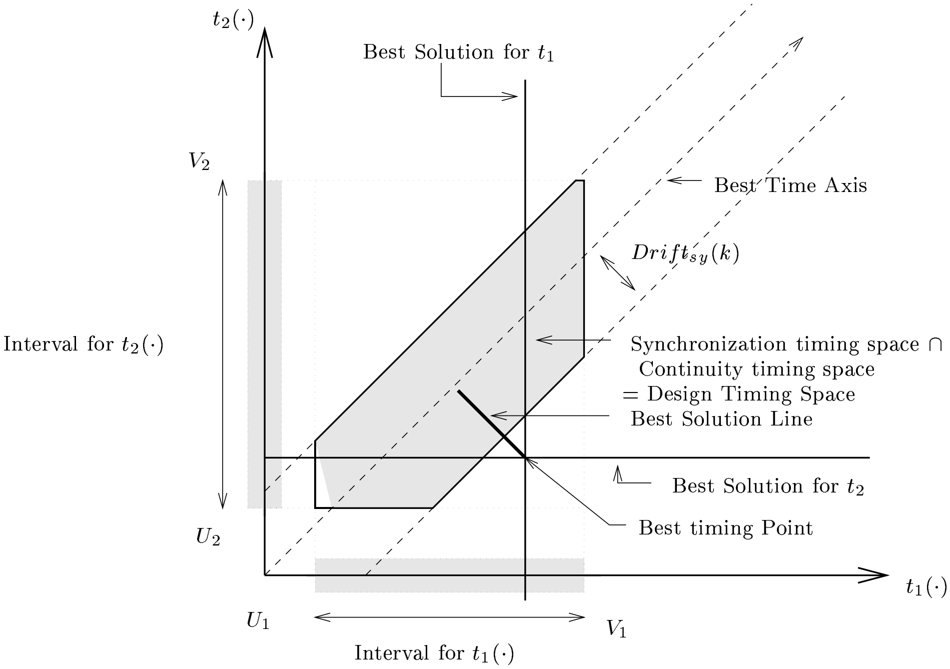

Figure 13: Design Space of Rate and Drift Profiles

5.3.1 Algorithm for Timing Selection

Analogous to the minimization of unit mixing loss in the case of our content selection algorithm, the timing selection algorithm attempts to minimize unit synchronization drifts. To state the timing selection algorithm, let `Best(k,i)` be the ideal time to display the k^{th} synchronization granule of stream $s_i(\cdot)$. Let the interval to begin displaying the k^{th} synchronization granule of stream $s_i(\cdot)$ be $[U_i(k), V_i(k)]$, as computed in (5). Let $Drift_{sy}(k)$ be the maximum allowable inter-stream drift computed by (21). For each call of the time selection algorithm, call the content selection algorithm: If the content selection algorithm finds a vector of synchronization granules, display it: otherwise increase the time. If the content selection algorithm is unable to find a vector of synchronization granules, then call the recovery algorithm. The algorithm to compute timing points follows:

Algorithm 2 Scheduling: Timing Points at Slot k

```
▷ To Compute timing points for the $k^{th}$ synchronization granule vector
$\vec{t_n} = (\sum_{i=1}^{n} Best(k,i)/n, \ldots, \sum_{i=1}^{n} Best(k,i)/n)$
Until $(\vec{t_n} \in \prod_{i=1}^{n}[U_i(k), V_i(k)])$
for each i= 1, n $t_i = t_i + 1$
    Call the content selection algorithm for a sync grannule vector
    if available, display.
end for each
if contents not displayed call the recovery algorithm.
end
```

Explanation

(a)At each time if available display a required synchronization granule.

(b) If not found advance the time.

(c) If not displayed at the end of time, call recovery algorithm.

The recovery algorithm is called only when appropriate media granules are unavailable at the display site. Policies and corresponding algorithms depend on the class of service that is to be provided, i.e. deterministic, probabilistic or best effort delivery with specified QoS metrics. If the delivery and display site management has deterministic guarantees, then the delivery mechanism has to make sure that the display site buffers are never empty, i.e. Algorithm 2 will never call the recovery algorithm. In case of services with probabilistic guarantees, the recovery algorithm has to make sure that defaults on specifications are kept to a pre-specified limit. Consequently, the delivery mechanism has to make sure that the probability of display site calling the recovery algorithm has to be kept below a certain value. This is a promising area of future work.

For best effort services, there are several options. One of them is to restart from an ideal position and hope that the delivery of media granules returns to normal. The other option is to momentarily suspend some defaulting streams until they can be displayed without violating overall specifications.

6 Summary, Conclusions and Future Work

In this paper we have defined continuity parameters for CM streams and shown how they can be beneficially used to schedule a display of CM streams. Our continuity parameters consist of three groups: sequencing, rate and drift parameters. While sequencing parameters determine what frames can be skipped or paused, rate and drift parameters limit delays and time drifts in schedules for rendition.

We have defined synchronization parameters that can be used to specify application needs. They consist of mixing, rate and drift profiles. Mixing profiles specify which combination of frames can be simultaneously displayed. Rate profiles specify rendition rates of a collection of streams, while drift profiles specify allowable timing drifts between otherwise simultaneously displayable frames from component streams.

The paper has an exhaustive categorization of what parameters are definable in terms of the others. Our results imply that rendition rate of a collection of streams can be defined in terms of rates of their components, while mixing and drift profiles cannot be defined in terms of corresponding parameters of component streams.

These results indicate that except for the perfect case, synchronization requirements cannot be specified by sufficiently stringent continuity requirements alone. Consequently, an intelligent display manager has to make some trade-offs between satisfying synchronization requirements of a collection of streams and continuity requirements of their components. Available options for implementing policies to balance between these two classes of requirements have been clearly brought forth by describing the design space that is available to a potential implementor. Finally, as a proof of applicability of our metrics, an integrated scheduling algorithm has been presented for a display site manager.

Our results on QoS at the application level can be translated to the network level, so that packets corresponding to media frames, or collections of them can be dropped by a traffic shaping algorithm, congestion control agent, or a buffer overflow manager at any switch in a network. As a sequel to current work, we are working in several directions. One of them is to investigate client-server algorithms to ensure starvation avoidance at display sites. The other is to develop local policies to handle starvation and buffer overruns at display sites. QoS based media mixing is an interesting extension to such work. Simultaneously, work proceeds in translating user supplied QoS to CM media server, communication media and operating system parameters such as network QoS, disk parameters and operating system schedulers, all of which can benefit from application supplied QoS hints.

References

[AFKN94] R.T. Aptekar, J.A. Fisher, V. Kisimov, and H. Nieshlos. Distributed Multimedia:User Perception and Dynamic QoS. In *SPIE* , volume 2188. SPIE, 1994.

[AFKN95] R.T. Aptekar, J.A. Fisher, V. Kisimov, and H. Nieshlos. Video Acceptability and Frame Rate. *IEEE Multimedia*, pages 32–40, September 1995.

[Bau92] W. Bauerfeld. RACE Project CIO (R2060): Coordination, Implementation and Operation of Multimedia Tele-Services on top of a Common Communication Platform. In *Proceedings of IWCA*, 1992.

[CCH94] A. Campbell, G. Coulson, and D. Hutchison. A Quality of Service Architecture. *ACM SIGCOM Computer Communication Review*, 1(2):6–27, April 1994.

[Chr89] Flaviu Christian. Probabilistic clock synchronization. *Distributed Computing*, 3:146–159, 1989.

[FBZ92] Dominico. Ferari, Anindo. Benarjea, and Hui. Zhang. Network Support for Multimedia; A Discussion of the Tenet Approach. Technical Report TR-92-072, University of California, Berkeley, 1992.

[Fer93] Dominoco Ferari. Delay jitter control scheme for packet-switching networks. *Computer Communication*, 15:367–373, 1993.

[FGMW92] Dominico Ferari, Amit Gupta, Mark Moran, and Bernd. Wolfinger. A Continuous Media Communication Service and Its Implementation. In *GLOBECOM '92*, December 1992.

[FSS95] K. Fujikawa, Shinjo Shimoto, and Hisashi Shimamura. Application Level QoS Modelling for a Distributed Multimedia System. In R Jain and N. C. Liu, editors, *1995 Pacific workshop on Distributed Multimedia Systems*, pages 44–52. Knowledge Systems Institute, March 1995.

[FV90] Dominoco Ferari and Dinesh Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE Journal on Selected Areas in Communication*, 8:368–379, April 1990.

[GM93] Anil Gupta and Mark Moran. Channel groups. Technical Report TR-93-015, ICSI and University of California at Berkeley, 1993.

[JS95] Kevin Jeffay and D. L. Stone. An Empirical Study of Delay Jitter Management Policies. *Multimedia Systems*, 2(6):267–279, January 1995.

[Mil95] I Miloucheva. QoS Research for Distributed Multimedia Applications. In R Jain and N. C. Liu, editors, *1995 Pacific workshop on Distributed Multimedia Systems*, pages 148–155. Knowledge Systems Institute, March 1995.

[NK82] W. E. Naylor and L. Klinerock. Stream traffic communication in packet-switched networks: destination buffering considerations. *IEEE Computer Communications*, 12:2527–2534, 1982.

[NS95] K. Nahrstedt and R. Steinmetz. Resource Management in Networked Multimedia Systems. *IEEE Computer*, pages 52–63, May 1995.

[Ran93] P. V. Rangan. Efficient Storage Techniques for Digital Continuous Multimedia. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):564–573, august 1993.

[RB93] K. Ravindran and V. Bansal. Delay Compensation Protocol for Synchronization of Multimedia Data streams. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):574 – 589, August 1993.

[RR93] S. Ramanathan and P. Venkat Rangan. Feedback Techniques for Intra-Media Continuaity and Inter-Media Synchronization in Distributed Multimedia Systems. *The Computer Journal*, 36(1):19 – 33, 1993.

[RTP94] S. V. Raghavan, Satish K. Tripathi, and B. Prabhakaran. On QoS Parameters and Services for MultiMedia Applications. Technical Report 3167, Department of Computer Science, University of Maryland, College Park, MD 20742 USA, 1994.

[RV93] P. V. Rangan and H. M. Vin. Designing a Multiuser HDTV Storage Server. *IEEE Journal on Seleted Areas in Communications*, 11(1):153–164, January 1993.

[SE93] R. Steinmetz and Clemens Engler. Human perception of media synchronization. Technical Report 43.9310, IBM European Networking Center, Heidelberg, 1993.

[SN95] J. M. Smith and K. Nahrstedt. The QoS Broker. *IEEE Multimedia*, 2(1):53–67, Spring 1995.

[SW94] Richard Staehli and Jonathan Walpole. Script-Based QOS Specification for Multimedia Presentations. Technical report, Department of Computer Science, Oregon Graduate Institute of Science and Technology, 1994.

[Tok92] H. Tokuda, editor. *Continuous media communication with dynamic control using ARTS with an FDDI network*, volume 10, 1992.

[Tow93] Don Towsley. Providing quality of service in packet switched networks. In L. Donatiello and R. Nelson, editors, *Performance Evaluation of Computer Communication Systems*, pages 560–586. Springer Verlag, 1993.

[VKvBG95] Andreas Vogel, Brigitte Kerherve, Gregor von Bochmann, and Jan Gecsei. Distributed Meltimedia and QoS: A Survey. *IEEE Multimedia Magazine*, 3:10–19, Summer 1995.

[WM91] Bernd. Wolfinger and Mark Moran. A Continuous Media Transport Service and Protocol for Real-Time Communication . In *Proceedings of the Second International Workshop on Network and Operating System Support for digital Audio and Video.* Springer Verlag, November 1991.

Index