Time-Interleaved Simulcast and Redundant Intra Picture Insertion for Reducing Tune-In Delay in DVB-H

Vinod Kumar Malamal Vadakital*, Miska M. Hannuksela†, and Moncef Gabbouj*
* Tampere University of Technology, Tampere, Finland
† Nokia Research Center, Tampere, Finland

***Abstract*— Tune-in delay is the time elapsed from the instant a program is chosen, until the rendering of the chosen program begins. This paper provides an analysis of various components that affect tune-in delay and discusses methods to reduce this delay, in the Digital Video Broadcasting – Handheld (DVB-H) systems. In DVB-H, time-slicing, and misalignment of random access units relative to the time-sliced burst boundaries are the major contributors to tune-in delay. Two methods, namely time-interleaved simulcast and redundant intra picture insertion, are proposed for reducing tune-in delay. Simulcasting is a transmission mechanism, where multiple programs of the same source material are coded in different qualities, and transmitted simultaneously. The receivers choose the most appropriate of the simulcasted programs, matching its capabilities, to be consumed. In the proposed time-interleaved simulcast method, the delay caused due to DVB-H time-slicing is reduced. It is done so, by transmitting time-sliced bursts of the simulcast programs maximally apart from each other. The proposed redundant intra picture insertion method reduces the delay related to alignment of random access pictures with time-slice boundaries. Theoretical results show that the proposed methods provide up to 60% reduction in tune-in delay.**

I. INTRODUCTION

The Digital Video Broadcasting-Handhelds (DVB-H) [1] specification enables power-constrained hand-held mobile devices to receive point-to-multipoint data. Time-slicing was introduced in the standard to reduce power consumption and it has been shown to improve power efficiency by as much as 90%. When time-slicing is used, information is transmitted in bursts. The bursts are transmitted at a bit rate substantially higher than the average bit rate of the information itself. A receiver powers on only when the information it needs, is being transmitted, and goes into standby otherwise, thus reducing power consumption.

However, the downside of time-slicing is the lengthened tune-in or channel-switch delay, compared to continuous transmission. The frequency and misalignment of random access units with respect to time-sliced burst boundaries, also contributes to increase the tune-in delay. Minimization of tune-in delay requires minimizing the probability that a receiver has to wait until the next burst of required information arrives. It also requires proper alignment of random access units with the boundaries of time-sliced bursts. A detailed analysis of the factors affecting tune-in delay is provided in this paper.

Recommended media codecs for DVB-H have been specified in [2]. Among those, H.264/AVC [3] is highly recommended for video compression in DVB-H. All discussions in this paper use H.264/AVC as basis. However, the methods proposed in this paper are general enough to support most other video coding standards. Five receiver classes, referred to as Internet Protocol Integrated Receiver-Decoder (IP-IRD) classes, are specified in [2] to address the heterogeneous processing capabilities of the transmitting and receiving devices.

Transmission to heterogeneous receivers tends to make the transmitter biased towards the lowest capability receivers. One method to reduce this bias is to organize the video stream into scalable layers [4]. Each additional layer enhances its base layer in some quality for e.g. spatial scalability. A receiver then chooses a subset of the coded layers that it can decode and present optimally with respect to its capabilities. H.264/AVC was recently extended by a scalable extension [5], known as Scalable Video Coding (SVC).

An alternative solution to heterogeneous receivers is simulcasting [6]. A number of independent bitstreams, e.g., one for each desired spatial resolution, is encoded from the same original video sequence and transmitted. A receiver chooses the stream that matches with its capabilities and consumes it.

The advantages of simulcasting compared to scalable coding include full compatibility with existing DVB-H receivers and lower computational requirements for receivers. If SVC bitstreams were used, only the base layer could be decoded by existing DVB-H receivers. Furthermore, decoding of an SVC bitstream is computationally more complex than decoding of an H.264/AVC stream resulting to the same spatial resolution and fidelity of the decoded pictures. Due to these facts, simulcasting could still be a practical solution for serving heterogeneous receivers.

The paper proposes two methods for reducing tune-in time in DVB-H broadcasts. The first method is applicable when at least two streams of the same origin are simulcasted. The method interleaves the time-sliced bursts of the simulcasted streams, such that each burst maximally apart from each other. The second method suits DVB-H systems where encoding is not done in synchronization with transmission. Here, redundant intra-coded pictures are inserted periodically during encoding, but only the first redundant intra-coded picture for each time-sliced burst is maintained for over-the-air transmission while the others are removed. Receivers can use the redundant intra-coded pictures when tuning in, while they do not affect the normal decoding process.

The rest of the paper is organized as follows. Section II introduces the DVB-H transmission system. Section III lists the different contributors for the tune-in delay and analysis the each contributor. Section IV introduces and analysis time-interleaved simulcast. Section V describes redundant intra picture insertion and analyses it. The potential reduction in tune-in delay is analyzed in section VI. Finally, conclusions are given in section VII.

II. DVB-H Transmission

In DVB-H, IP packets are encapsulated into Multi-protocol encapsulation (MPE) sections [7]. However, before encapsulation into MPE sections, the IP packets can be optionally protected by Reed-Solomon (RS) forward error correction (FEC) codes. The FEC parity bytes are called MPE-FEC.

A. *Computing MPE-FEC*

IP packets of a multimedia program **P,** fill a two-dimensional matrix $\mathbf{M}_{ADT}$. This matrix $\mathbf{M}_{ADT}$, is called the *application data table* (ADT). $\mathbf{M}_{ADT}$ has dimensions $(r \times k)$: $r \in \{256, 512, 768, 1024\}$ and $0 < k \leq 191$. Each cell in $\mathbf{M}_{ADT}$ can accommodate an information byte. IP packets are filled into $\mathbf{M}_{ADT}$ in column major order. No IP packet in $\mathbf{M}_{ADT}$ can be fragmented. For each row in $\mathbf{M}_{ADT}$, a (n, k) RS FEC code is computed. RS FEC codes are systematic codes. Hence, $\mathbf{M}_{ADT}$ is unaffected after FEC coding. The r rows of $(n\text{-}k)$ parity bytes form another matrix $\mathbf{M}_{RSDT}$. This matrix $\mathbf{M}_{RSDT}$ is called the *Reed Solomon data table* (RSDT). A horizontal concatenation of the matrices $\mathbf{M}_{MPE\text{-}FEC} = \mathbf{M}_{ADT} \mid \mathbf{M}_{RSDT}$ results in a matrix of dimensions $(r \times n)$: $r \in \{256, 512, 768, 1024\}$ and $0 < n \leq 255$. This matrix is called the MPE-FEC matrix and its structure is shown in Fig. 1.

The unfilled cells of $\mathbf{M}_{ADT}$ are filled with zeros and these cells are collectively called *padding*. Whole columns of $\mathbf{M}_{ADT}$ that are padded are called *padding columns*. Code-rate/bitrate control of the transmitted data is achieved by varying the amount of padding in the ADT, and/or varying the number of RSDT columns transmitted. The RSDT columns that are not transmitted are referred to as *punctured columns*.

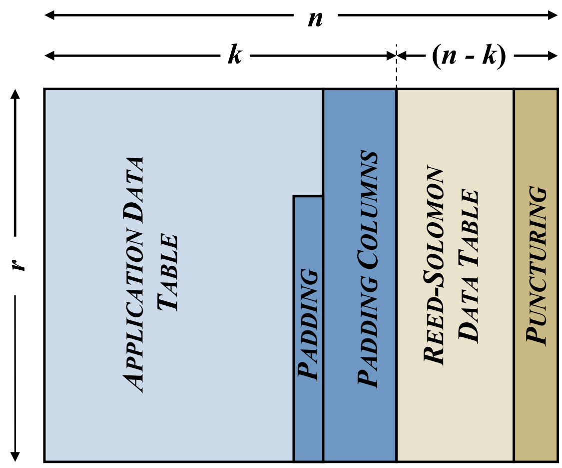

Fig. 1. Structure of an MPE-FEC matrix.

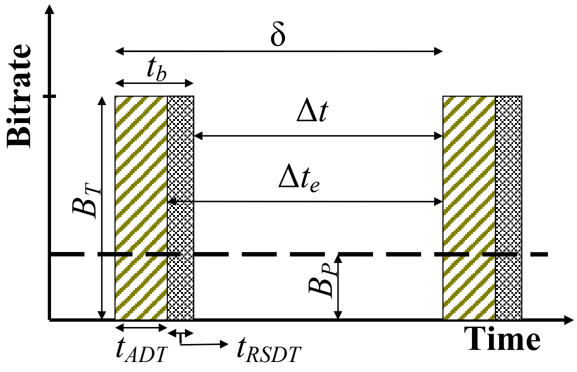

Fig. 2: Time-sliced burst parameters.

All IP packets in $\mathbf{M}_{ADT}$ are encapsulated into MPE sections. The non-punctured columns of $\mathbf{M}_{RSDT}$ are encapsulated into MPE-FEC sections. The MPE and the MPE-FEC sections are then fragmented at the transport layer into constant-size MPEG-2 transport stream (TS) packets, before time-sliced transmission. Henceforth, in this paper, the program data TS packets in $\mathbf{M}_{ADT}$ and the FEC TS packets in $\mathbf{M}_{RSDT}$ are referred to as data and FEC respectively.

B. *Time-Slicing*

The TS packets of **P** are sent in time-sliced bursts to achieve receiver power savings. Henceforth, time-sliced bursts are simply referred to as bursts and it is shown in Fig. 2. Each burst carries one $\mathbf{M}_{MPE\text{-}FEC}$. The bursts are transmitted at a bitrate B_T, which is substantially higher than the average bitrate B_P, of **P**. FEC is sent after data in the burst.

The program **P** is allocated the entire or some part (much larger than B_P) of bitrate B_T, for a known period of time. Data and FEC belonging to one $\mathbf{M}_{MPE\text{-}FEC}$ of **P** is transmitted during this period, called the *burst-time* t_b. The parts of t_b used for transmitting data and FEC, are denoted by t_{ADT} and t_{RSDT} respectively. After t_b, no data or FEC of **P** is transmitted for the *off-time* duration Δt. Similarly, no data of **P** is sent for the *effective off-time* duration Δt_e. The *cycle-time* δ, is defined as the time between the start of two consecutive bursts

$$\delta = \left(t_b + \Delta t\right) = \left(t_{ADT} + \Delta t_e\right) \quad (1)$$

During Δt other programs can use the channel. The next time **P** can be transmitted is after a duration Δt. A receiver interested in consuming **P**, switches on when **P** is being transmitted for a duration t_b, after which it switches off for a period of Δt, before switching on again. The off-time Δt and the *burst-time* t_b can be freely varied.

C. *Transmission of Multimedia Data in Bursts*

In this article, **P** contains only audio and video. They are coded separately and multiplexed together. Let a *sampling curve* be defined as the total amount of bits produced by a media encoder up to a sampling instant τ. Any compressed audio and video bitstreams can then be represented by their sampling curves, $C_a[\tau]$ and $C_v[\tau]$ respectively, as shown in Fig. 3. Both curves are monotonically non-decreasing $\forall\,\tau$. In a short time window, the curves reveal a staircase structure, with rising edges at the sampling instants.

The audio and video components of **P** can be collectively represented by a single sampling curve $\varphi[\tau]$. This curve $\varphi[\tau]$ is called the *program sampling curve*. It is defined as the cumulative sum of all component multimedia data until a sampling instant τ and here computed as

$$\varphi[\tau] = C_a[\tau] + C_v[\tau] \tag{2}$$

where $C_a[\tau]$ and $C_v[\tau]$ are the sampling curves for audio and video of **P** respectively. A burst of **P** contains the multiplexed multimedia data in an interval $[\tau_s, \tau_e]$, as shown in Fig. 3. If $\mathbf{D} \in \mathbf{P}$ is the part of **P** transmitted in a burst, then the total data bits $|\mathbf{D}|$ in the burst is given by

$$|\mathbf{D}| = \left(C_a[\tau_e] - C_a[\tau_s]\right) + \left(C_v[\tau_e] - C_v[\tau_s]\right) \tag{3}$$

III. TUNE-IN-DELAY IN DVB-H

Let *tune-in initiation time* τ_t be defined as the instant when the user decides to consume **P** and initiates action to receive data from the channel. *Tune-in delay* $\Delta_{(T\text{-}IN)}$, also known as *start-up delay*, is defined as the amount of time elapsed after τ_t to the moment when the rendering of **P** starts. *Channel zapping delay* or *channel-switch delay* refers to changing from one program to another and is equivalent to $\Delta_{(T\text{-}IN)}$. $\Delta_{(T\text{-}IN)}$ can be considered as a cumulative sum of the following component delays:

A1. Time-slice synchronization delay $\Delta_{(T\text{-}SYNC)}$.

A2. Delay to compensate potentially incomplete reception of the first burst $\Delta_{(COMP)}$.

B. Reception duration of the first time-sliced burst $\Delta_{(RCPT)}$.

C. Delay to compensate the size variation of FEC $\Delta_{(FEC)}$.

D. Delay to compensate for the synchronization time between associated media streams (e.g. audio and video) $\Delta_{(M\text{-}SYNC)}$.

E. Delay until media decoders are refreshed to produce correct output samples $\Delta_{(REFRSH)}$.

F. Delay to compensate the varying bitrate of a media bitstream $\Delta_{(VBR\text{-}COMP)}$.

G. Processing delays of the receiver and player implementations $\Delta_{(PROC)}$.

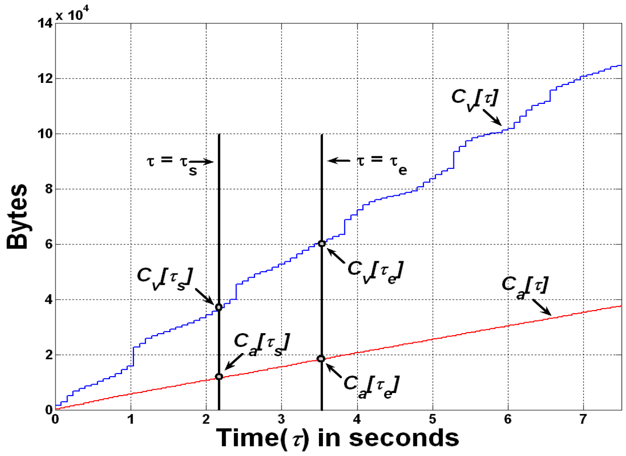

Fig. 3: Sampling curves $C_a[\tau]$ and $C_v[\tau]$ and time-slicing interval $[\tau_s, \tau_e]$

Thus, $\Delta_{(T\text{-}IN)}$ can be given as

$$\begin{aligned}\Delta_{(T-IN)} =& \Delta_{(T-SYNC)} + \Delta_{(COMP)} + \Delta_{(RCPT)} \\ &+ \Delta_{(FEC)} + \Delta_{(M-SYNC)} + \Delta_{(REFRSH)} \\ &+ \Delta_{(VBR-COMP)} + \Delta_{(PROC)}\end{aligned} \tag{4}$$

Equation (4) is a simplification, as the delay to acquire the required transport-level signaling, such as Program Specific Information/Service Information (PSI/SI) and Entitlement Control Messages (ECM) for conditional access (CA), are not considered. Furthermore, it is assumed that no content protection is used and hence related delays e.g. for acquiring the content protection keys are omitted from the discussion. Finally, the delay jitter of burst intervals (Delta-t Jitter) is not handled either, as its implication is regarded as obvious for implementations.

Each component of $\Delta_{(T\text{-}IN)}$ is discussed in more detail below.

A. *Time-slicing synchronization delay $\Delta_{(T\text{-}SYNC)}$ and delay to compensate potentially incomplete reception of the first time-sliced burst $\Delta_{(COMP)}$.*

There are two possibilities for τ_t, relative to the transmission of **P**:

- Tune-in τ_t, is during the data transmission time t_{ADT}. It is marked as τ_{t1} in Fig. 4a. In a special case, τ_t is exactly at the beginning of a burst carrying **P.** This case is indicated as τ_{t2} in Fig. 4a.
- The tune-in τ_t, is during Δt_e of **P**, as indicated by τ_{t3} in Fig. 4b.

Before analysis of these scenarios, two delays are defined. The first, called *time-slice synchronization* delay $\Delta_{(T\text{-}SYNC)}$, is defined as the time elapsed from τ_t to the moment when the receiver starts getting data of **P**. The second, called the *incomplete ADT compensation* delay $\Delta_{(COMP)}$, is incurred to compensate for the playback duration of data that was not received before τ_t in the burst. This delay is applicable only when τ_t occurs during t_{ADT}.

When τ_t is during t_b, the decoding and/or playback has to be delayed by $\Delta_{(COMP)}$. This time $\Delta_{(COMP)}$, is equal to the playback duration of those coded data units that came in the burst prior to the tune-in time τ_t. Delaying the decoding and/or playback by $\Delta_{(COMP)}$ guarantees that the media is presented without any pause. In the special case, when τ_t is exactly at the beginning of a burst, all data need for decoding is available and hence $\Delta_{(COMP)} = 0$. Sometimes it may not be possible to apply FEC decoding of an incompletely received burst. This is because the amount of data columns that were not received may outnumber the correction capability of the FEC code. To simplify further delay analysis of $\Delta_{(T\text{-}SYNC)}$ and $\Delta_{(COMP)}$, the following are assumed.

- Data is transmitted in decoding order.
- Audio and video frames are interleaved in ascending order of decoding times.
- The decoding order is identical to the output order.
- **P** is CBR.

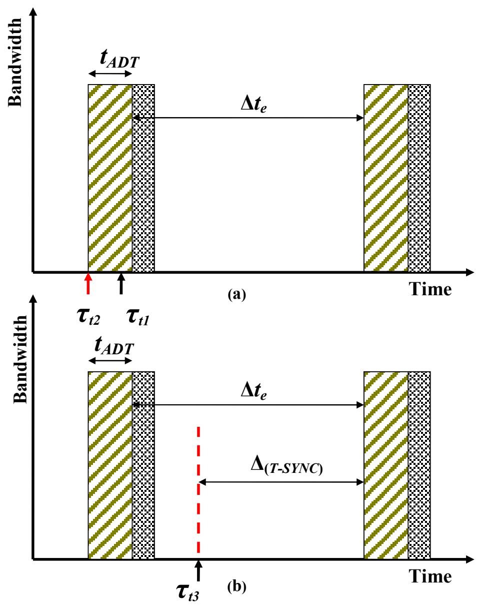

Fig. 4: Three possible instances of tune-in initiation time τ_t

Assuming τ_t to be a uniformly distributed random variable, $\Delta_{(COMP)}$ ranges from 0 to the cycle time δ, and the expected $\Delta_{(COMP)}$ is

$$E\left[\Delta_{(COMP)}\right] = \delta / 2 \tag{5}$$

The expected $\Delta_{(T\text{-}SYNC)}$ is

$$E\left[\Delta_{(T-SYNC)}\right] = \Delta t_e / 2 \tag{6}$$

If E_b is defined as an event when τ_t is during t_{ADT}, then the probability of this event occurring is given by

$$P(E_b) = t_{ADT} / \delta \tag{7}$$

When τ_t is during Δt_e, it has to wait until the next burst to start decoding **P**. If E_o is defined as an event that τ_t is during Δt_e, then the probability of E_o occurring $P(E_o)$, is given by

$$P(E_o) = \Delta t_e / \delta \tag{8}$$

Receivers tuning in at τ_{t1} will experience $\Delta_{(COMP)}$, and those tuning in at τ_{t3} will experience $\Delta_{(T\text{-}SYNC)}$. Receivers tuning in at τ_{t2} will experience neither $\Delta_{(COMP)}$ nor $\Delta_{(T\text{-}SYNC)}$.

A method to reduce $\Delta_{(T\text{-}IN)}$ when τ_t is during t_{ADT} is described in [8]. The method enables decoding and rendering some parts of data in the incompletely received burst. This partly rendered data is perceived as a reduced quality representation of **P**. This transmission arrangement was shown to provide tune-in delay reduction in continuously transmitted services. However, its impact is not as pronounced in DVB-H, because of short burst transmission time t_b.

A technique known as *parallel elementary streams* [9] reduces $\Delta_{(T\text{-}SYNC)}$ to zero, when channel-switching. Here, more than one program is encapsulated into the same burst. A switch between the programs within the same burst therefore causes no delay to synchronize with burst for the new desired program. However, this method reduces the cycle-time and consequently increases the power usage required for reception.

B. *Reception duration of the time-sliced burst* ($\Delta_{(RCPT)}$)

The reception duration of a burst depends on the size of the first $\mathbf{M}_{MPE\text{-}FEC}$ carrying **P** and the transmission bitrate B_T. DVB-H allows the service provider to select the size of $\mathbf{M}_{MPE\text{-}FEC}$ within certain bounds. The amount of data in $\mathbf{M}_{MPE\text{-}FEC}$ can be further controlled by using padding and puncturing. The transmission bitrate B_T depends largely on the modulation scheme used in the physical layer for radio transmission. Higher order modulations like 64-QAM provide higher B_T, than say QPSK. However, using higher order modulation also increases the data's susceptibility to errors. Furthermore, potential non-time-sliced services reduce the transmission bitrate of the time-sliced bursts accordingly.

If receivers started media decoding immediately after receiving the first IP datagram of **P**, i.e., during t_{ADT}, an erroneous IP datagram might not be correctable by FEC decoding before its rendering time. Hence, receivers should buffer an entire MPE-FEC frame and apply FEC decoding, if necessary, before decoding of the media streams.

If τ_t is a uniformly distributed random variable, then the expected reception time, when τ_t is during Δt is given by

$$E\left[\Delta_{(RCPT)}\right] = t_b \tag{9}$$

When τ_t is during t_b the expected reception time is given by

$$E\left[\Delta_{(RCPT)}\right] = t_b / 2 \tag{10}$$

If parallel elementary streams are used and programs are switched within the same burst, the $\Delta_{(RCPT)}$ is not counted towards $\Delta_{(T\text{-}IN)}$.

C. *Delay to compensate the size variation of FEC blocks* ($\Delta_{(FEC)}$)

Program playback without a pause may require additional initial buffering beyond the reception of the first burst. This is due to instantaneous bitrate variations in **P** and consequently size variation of bursts. Let $t(i)$ be the reception time of the last bit of the i^{th} burst in transmission order. Furthermore, let $b(i)$ be the number of bits in the IP payloads of a media stream within the burst i, and $r(i)$ be the average bitrate of the media stream. If, for $\forall$ i,

$$b(i)/r(i) = t(i+1) - t(i) \tag{11}$$

the initial buffering duration to compensate for instantaneous bitrate variations of bursts would always be zero. However, this is might not be the case. For e.g.

1. The FEC code rate k/n and amount of padding may vary.

2. The scheduling of bursts may not follow accurately Equation(11), but rather follow the average bitrates of **P** and the δ derived from average bitrate.

3. If parallel elementary streams are used, meeting the bit budget accurately for every stream within a burst is a challenging task for VBR media like video.

In order to allow DVB-H receivers to compensate for the instantaneous bitrate variation of bursts, a minimum buffering time for this delay component is signaled for each program as a parameter in the Session Description Protocol (SDP) [10]. The minimum buffering time is specified relative to the hypothetical receiver buffering model in [10].

D. Delay to compensate the synchronization between associated streams ($\Delta_{(M\text{-}SYNC)}$)

Given that a burst contains audio-visual data of $\mathbf{P}$ in the interval $[\tau_s, \tau_e]$, it is advantageous to transmit associated audio and video data in the same burst. Furthermore, it is also preferable that the rendering start time and rendering end time of the audio frames in a burst, match as closely as possible to the rendering start time and rendering end time of the video pictures in the same burst. If this is not the case, then the playback of either audio or video has to be delayed, for synchronized playback.

Rendering times of the data is conveyed in the timestamp field of the RTP header of each packet. However, the RTP timestamps are initialized to a random value for audio and video independently. The timestamps are indicated in terms of clock ticks of a base clock frequency. The base clock frequency for audio and video are typically different. RTCP sender reports are transmitted periodically for each RTP stream to relate the RTP timestamp of the associated RTP stream to the wall clock time of the sender. To synchronize audio and video playback, the player has to map the RTP timestamps of both the streams to the same timeline. It is advantageous to send at least one RTCP sender report for each stream in every burst, to achieve correct audio-video synchronization. More information on audio-video synchronization can be obtained from [2].

E. Delay until media decoders are refreshed to produce correct output samples ($\Delta_{(REFRSH)}$)

An audio/video decoder cannot start immediate decoding after τ_t. Due to the use of prediction in audio/ video compression, a decoder has to wait until the next random access point before commencing decoding. Intra coded pictures of a video sequence, and all audio frames can be independently decoded. These independently decodable entities can be used as random access points. The delay $\Delta_{(REFRSH)}$, is the time elapsed after τ_t, until the next random access point. Since the play-out duration of an audio frame is typically smaller than a video random access point interval, the discussion here is focused on $\Delta_{(REFRSH)}$ for video only.

Any intra-coded picture is a random access point when reference pictures for inter prediction cannot be selected. However, when multiple reference pictures are used, and reference picture selection is allowed, random access intra pictures have to be specifically indicated. H.264/AVC provides two types of intra pictures that enable random access. The first is an Instantaneous Decoding Refresh (IDR) picture. The other is an intra picture at the beginning of an open group of pictures (GOP). All pictures following an IDR picture in decoding order can be correctly decoded. An open GOP starts with an intra picture having the property that no picture following it in output order refers (in it's inter prediction process) to a picture prior to it in decoding order. This means that, when decoding is initiated from an intra picture starting an open GOP, all pictures following the intra picture in output order can be correctly decoded but some picture preceding the intra picture may not be decodable.

To minimize $\Delta_{(T\text{-}IN)}$ for video playback, every burst should have a random access intra picture as the first picture in output order. Assuming one random access picture per burst the expected $\Delta_{(REFRSH)}$ is given by

$$E\left[\Delta_{REFRSH}\right] = \left(\delta/2\right) + (1/f) \qquad (12)$$

where f is the picture rate of the video. Methods to start a burst with an intra picture can be classified into adaptive FEC blocking and adaptive picture grouping.

In adaptive FEC blocking, the size of $\mathbf{M}_{MPE\text{-}FEC}$ is selected so that a random access intra picture is the first picture, in output order, within $\mathbf{M}_{MPE\text{-}FEC}$. This implies that the amount of data, FEC, and Δt are likely to vary between bursts. Composition of $\mathbf{M}_{MPE\text{-}FEC}$ using adaptive FEC blocking requires look-ahead and buffering of $\mathbf{P}$. The look-ahead and buffering requirements extend beyond the size of a single $\mathbf{M}_{MPE\text{-}FEC}$ to avoid receiver buffer violations.

Adaptive picture grouping refers to encoding and bitstream adaptation methods that aim to place a random access intra picture as early as possible (in output order) within a burst. In DVB-H systems where the encoding of programs is live and real-time, a feedback from the $\mathbf{M}_{MPE\text{-}FEC}$ composition process to the video encoder is used to place the random access intra picture where needed. In systems that transmit pre-encoded programs, it is difficult to know the exact location of random access pictures and other burst parameters, beforehand, without pre-processing the program. Two methods that aim to reduce $\Delta_{(REFRSH)}$ are now reviewed.

Tourapis and Boyce [11] proposed an auxiliary bitstream transmitted along with the primary bitstream. The auxiliary bitstream was coded with frequent random access pictures coded at lower quality. A receiver wanting to tune-in received and decoded the first random access picture arriving after τ_t in the auxiliary bitstream. Then further decoding is continued from next picture, following the random access picture in decoding order, in the primary bitstream. The method of Tourapis and Boyce [11] assumed continuous transmission. Therefore, the auxiliary bitstream was not synchronized to burst boundaries making it is sub-optimal in terms of bitrate for time-sliced transmission. Furthermore, this method might not be compatible with the existing video coding standards. The method requires that the decoded intra picture be buffered in the decoded picture buffer, after which the decoding would continue seamlessly from the primary bitstream. This kind of a decoder stream switching is not described in the video standards.

In [12], an auxiliary bitstream containing intra-coded pictures is transmitted along with the primary bitstream to the IP encapsulator. For each burst, the IP encapsulator inserts the first possible intra-picture from the auxiliary bitstream instead of the corresponding picture in the primary bitstream. There is an inevitable mismatch in the reconstructed picture of the modified primary bitstream, and this mismatch propagates in reconstructed video due to predictive coding. It was experimented in [12] that the propagated error was subjectively minor. The mismatch could be avoided altogether by using of SI and SP pictures, proposed in [13], which are, however, not included in the profiles included in the IP-IRD classes for DVB-H.

F. Delay to compensate the varying bitrate of a media bitstream ($\Delta_{(VBR\text{-}COMP)}$)

Audio streams coded for multimedia transmission are typically CBR. However, video streams require initial buffering to smoothen out short-term bitrate variations. The amount of initial buffering required, is specified relative to the operation of a hypothetical reference decoder (HRD) model.

The H.264/AVC HRD model contains a coded picture buffer (CPB), an instantaneous decoding process, a decoded picture buffer (DPB), and an output picture cropping block. The CPB and the instantaneous decoding process are specified similar to any other video coding standard. The output picture cropping block, simply crops samples that are outside the signaled output picture extents, from the decoded picture. In H.264/AVC, the DPB was introduced to control memory requirements for decoding conformant bitstreams.

The input bitrate to the CPB is typically constant and usually corresponds to the average bitrate of the stream. Every random access intra picture has an initial CPB buffering delay. This delay is conveyed in the bitstream and guarantees that decoded picture output occurs at the correct pace without overflowing the CPB or DPB. The size of the CPB is limited by the Level definitions (Annex A) of H.264/AVC. When the maximum bitrate allowed by any particular level is used, the CPB size is approximately in the range of 1 second to 2.5 seconds of coded data, depending on the level in use.

In H.264/AVC, the decoding and output order of pictures need not be identical. Decoded pictures are buffered for two reasons. First, they are used as reference pictures in inter prediction, and second, for reordering decoded pictures to the output order. The DPB includes a unified decoded picture buffering process for reference pictures and output reordering. A decoded picture is removed from the DPB when it is no longer used as reference and needed for output. The maximum size of the DPB that bitstreams are allowed to use is specified in the Level definitions of H.264/AVC. The DPB initial buffering can be indicated in the video bitstream.

A DVB-H receiver should delay the video playback as indicated by the initial CPB and DPB buffering delays. The start of audio playback should be delayed and synchronized with video accordingly.

G. Processing delays of the receiver and player implementations ($\Delta_{(PROC)}$)

Processing delay $\Delta_{(PROC)}$ associated with FEC and media decoding are inevitable, but the hypothetical buffering models assumes it to be zero. Furthermore, the protocol stack implementations of the receiver may cause additional initial delays. However, since these delays are implementation dependent, they are excluded from the consideration of this paper.

The tune-in $\Delta_{(T\text{-}IN)}$ delay can be significantly reduced

- If the time spent by a receiver to wait before data of the next burst arrives is reduced.
- The random access points are synchronized with the time-slicing burst structure.

The next two sections propose methods to reduce each of the above mentioned components of tune-in delay.

IV. Time-Interleaved Simulcast

Let a simulcast session $\mathbf{S}$ transmitting the same source signal be defined as $\mathbf{S} = \{\mathbf{P}_i: 1 \le i \le k;\ i, k \in \mathbf{N}\}$, where $\mathbf{S}$ consist of k independently coded programs $\mathbf{P}_i$, targeted at k different classes of receivers. The programs $\mathbf{P}_i$ are coded at different qualities, picture sizes, and/or picture rates. As a convention, let $\mathbf{P}_1$, targeted at the lowest class of receivers, be coded at the lowest quality and $\mathbf{P}_k$, targeted at the highest class of receivers, be coded at the best quality. It is assumed that a receiver in class M: $1 \le M \le k$, can decode all streams of a lower class than itself $\mathbf{P}_i$: $1 \le i \le M$, but no streams targeted to receivers in a higher class $\mathbf{P}_i$: $M < i \le k$. For each program $\mathbf{P}_i \in \mathbf{S}$ the program sampling curve $\varphi_i[\tau]$ as shown in Fig. 5 can be drawn.

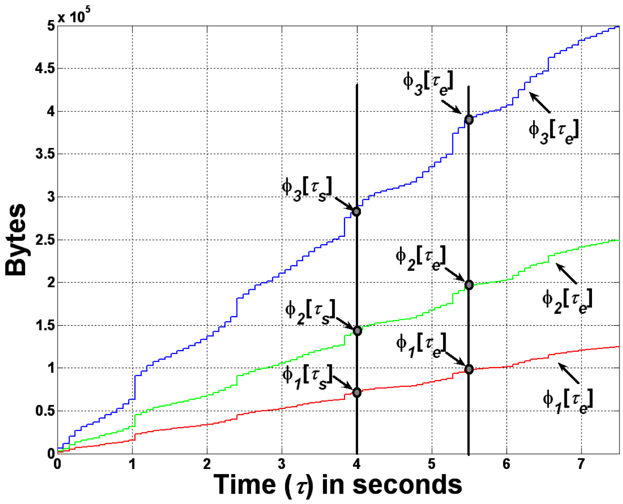

Fig. 5: Program sampling curves $\varphi_1[\tau]$, $\varphi_2[\tau]$, $\varphi_3[\tau]$ for three $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ and a burst time interval $[\tau_s, \tau_e]$.

A. Non-Interleaved Simulcasting (NIS)

This method is a straightforward realization of simulcasting. Let $\mathbf{D}_{(i,j)}$: $1 \le i \le k;\ j, i, k \in \mathbf{N}$ be the data of $\mathbf{P}_i$ during the j^{th} time interval $[\tau_{(i,j)}, \tau_{(i,j+1)}]$. It is assumed here that $\mathbf{D}_{(i,j)}$ for any i and j, fits into one $\mathbf{M}_{MPE\text{-}FEC}$. The time intervals $[\tau_{(i,j)}, \tau_{(i,j+1)}]$ are the same across all $\mathbf{P}_i$ and $\forall j$. Hence, $[\tau_{(i,j)}, \tau_{(i,j+1)}]$ is simply denoted as $[\tau_{(j)}, \tau_{(j+1)}]$. The data $\mathbf{D}_{(i,j)}$ $\forall i$, in the time interval $[\tau_{(i,j)}, \tau_{(i,j+1)}]$ is sent as consecutive bursts, herein called a burst-cluster $\mathbf{B}_j$. Every burst in $\mathbf{B}_j$ contains data of one and only one program. Within $\mathbf{B}_j$, the bursts are so arranged that the programs $\mathbf{P}_i$ are transmitted in ascending order. This means that the burst containing $\mathbf{P}_k$, which has the best quality, is transmitted first in $\mathbf{B}_j$, and the burst containing $\mathbf{P}_1$, which has the worst quality, is transmitted last in $\mathbf{B}_j$.

Fig. 6 shows the realization of NIS for $k = 3$ programs. Assume that there are k bursts, one from each $\mathbf{P}_i$, in any $\mathbf{B}_j$. Let t_b be redefined as the transmission time of $\mathbf{B}_j$ (rather than a single burst in $\mathbf{B}_j$) and Δt be redefined as the off-time between two consecutive $\mathbf{B}_j$ (rather than two consecutive bursts). It is easily seen that, NIS is similar to a time-slice transmission of a single program with the only difference being that, instead of transmitting a burst in the single program case, a burst cluster $\mathbf{B}_j$ is transmitted in the simulcasting case. The value of t_b is small compared to Δt. Therefore, it is very probable that the biggest contributing component to $\Delta_{(T\text{-}IN)}$, $\Delta_{(T\text{-}SYNC)}$ is still comparatively large.

However, a small reduction of $\Delta_{(T\text{-}IN)}$ can still be achieved. Assume a receiver R_M, M: $1 \le M \le k$, wants to tune into $\mathbf{S}$. The program $\mathbf{P}_M$ is what it aims to consume. Let τ_t be during the transmission of $\mathbf{B}_j$. If τ_t be the instant before the arrival of $\mathbf{P}_M$ in the burst, then R_M has $\Delta_{(T\text{-}SYNC)} = 0$. If τ_t be the instant after the arrival of $\mathbf{P}_M$ in the burst, then it is possible that R_M can still receive the program $\mathbf{P}_N$, $1 \le N \le M$, which comes after $\mathbf{P}_M$ in $\mathbf{B}_j$. The probability that R_M can decode a program $\mathbf{P}_N$ in $\mathbf{B}_j$, increases with M. This tuning in is possible, only because of the arrangement of $\mathbf{P}_i$ in $\mathbf{B}_j$. But, due to the relatively low probability τ_t occurring during t_b, the expected $\Delta_{(T\text{-}IN)}$ reduction is small.

If $\Delta t_{e(i)}$ denote the effective off-time for the transmission program $\mathbf{P}_i$, then the expected value of $\Delta_{(T\text{-}SYNC)}$ is given by

$$E\left[\Delta_{(T-SYNC)}\right] = \Delta t_{e(i)}/2 \tag{13}$$

and the expected value of $\Delta_{(T\text{-}COMP)}$,

$$E\left[\Delta_{(T-COMP)}\right] = \left(\tau_{(j+1)} - \tau_{(j+1)}\right)/2 \tag{14}$$

B. *Time-interleaved Simulcasting (TIS)*

The drawback of NIS is that $\Delta_{(T\text{-}SYNC)}$ is still comparatively large. Time-interleaved simulcasting (TIS) solves this shortcoming. TIS, like NIS, has the same time interval $[\tau_{(i,j)}, \tau_{(i,j+1)}]$ across all $\mathbf{P}_i$ and so again $[\tau_{(i,j)}, \tau_{(i,j+1)}]$ is simply denoted as $[\tau_{(j)}, \tau_{(j+1)}]$. The information $\mathbf{D}_{(i,\ j)}$ corresponding to the time interval $[\tau_{(j)}, \tau_{(j+i)}]$ is also arranged in ascending order of $\mathbf{P}_i$, as in NIS. However, the data in a time interval is not transmitted as a burst cluster, rather as individual bursts. The Δt between two bursts carrying $\mathbf{D}_{(k,\ j)}$ is divided into k partitions Δt_i: $1 \le i \le k$, as shown in Fig. 7(b). After each Δt_i data $\mathbf{D}_{(k\text{-}i,\ j)}$ is transmitted in a burst. Each burst, similar to NIS, carries data of one and only one program in the session.

To achieve the most gain from TIS, Δt_i should be chosen so that the burst are placed maximally apart. The bursts are maximally apart when $\Delta t_i = \Delta t_j$: $1 \le i \le k$, $1 \le j \le k$, $\forall$ i, j. When receiver R_M where M: $1 \le M \le k$, attempts to tune into $\mathbf{S}$, the average expected $\Delta_{(T\text{-}SYNC)}$ is reduced by a factor of k. This $\Delta_{(T\text{-}SYNC)}$ reduction k is always true for the highest class of receiver R_k. Furthermore, a receiver of any class can often decode at least part of the data meant for other receivers, such as a subset of a temporally scalable video stream. Hence, the reduction factor k can also be true for lower classes of receivers. If R_M tunes into a cycle after a burst of $\mathbf{D}_{(M,\ j)}$, it can still tune into a burst of $\mathbf{D}_{(N,\ j)}$ where $1 \le N \le M$ which is data of lower quality. It decodes and renders the data of $\mathbf{D}_{(N,\ j)}$ until it $\mathbf{D}_{(M,\ j+1)}$.

However this method has a disadvantage. If $\Delta_{(INIT)}$ is the initial start up delay for first time-interval $[\tau_{(0)}, \tau_{(1)}]$ to be available for transmission, then all programs $\mathbf{P}_i$: $i \ne k$ experience additional delays before the arrival of the first burst. This means that receivers of different classes render different information at any time instant.

C. *Overlapped Time-Interleaved Simulcasting (OTIS)*

TIS introduce unequal initial delay to different classes of receivers. Overlapped time-interleaved Simulcasting (OTIS) solves this limitation. Here, similar to TIS, the programs $\mathbf{P}_i$ are time-interleaved such that all bursts are spaced maximally apart from each other. The first time-slices for all $\mathbf{P}_i$ are transmitted back-to-back as one burst. Let the j^{th} cycle-time $\delta_{(i,\ j)}$, for a program be a constant C. Furthermore, let $\delta_{(i,\ j)}$ be the same $\forall$ $\mathbf{P}_i$ after the burst $j = 1$. The first cycle-time $\delta_{(i,\ 1)}$, is computed as

$$\delta_{(i,1)} = C\left(k - i + 1\right)/k : 1 \le i \le k, i \in \mathrm{N}, C \in \mathrm{R} \tag{15}$$

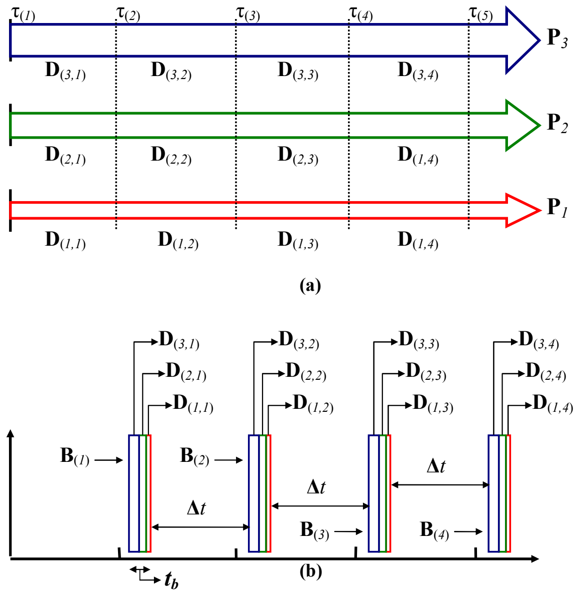

Fig. 6: Non-Interleaved simulcasting (a) three programs and their respective information time-intervals (b) burst transmission and program arrangement within each burst.

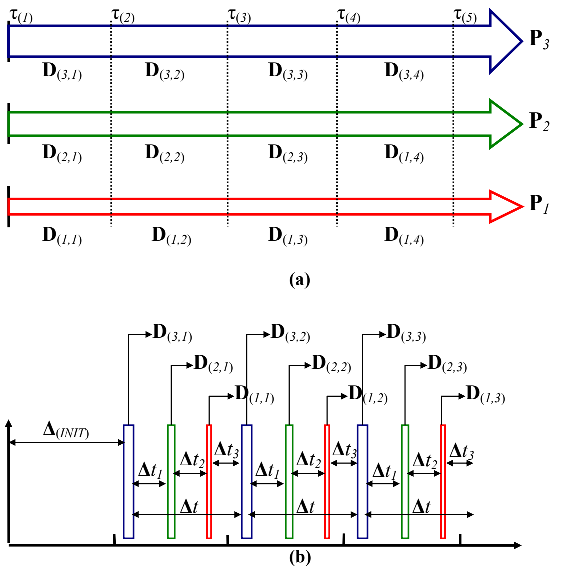

Fig. 7: Time-Interleaved Simulcasting (a) Three programs and their respective time intervals (b) burst transmission and program arrangement within each burst.

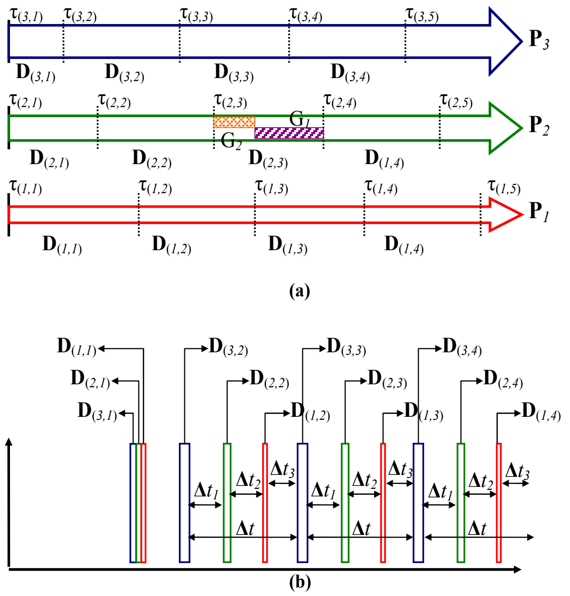

Fig. 8: Time-Interleaved Simulcasting (a) Three programs and their respective time intervals (b) burst transmission and program arrangement within each burst.

As was the case with both NIS and TIS, $\mathbf{D}_{(i,\ j)}$ corresponding to the time interval $[\tau_{(j)}, \tau_{(j+i)}]$ is transmitted in ascending order of $\mathbf{P}_i$. Fig. 8 shows a simulcast of three k = 3 programs. The time-intervals $[\tau_{(i,j)}, \tau_{(\mathrm{i},j+1)}]$ and the data $\mathbf{D}_{(i,j)}$ within each program $\mathbf{P}_i$ are presented in Fig. 8 (a). Fig. 8(b) shows the burst transmission order, the constant off-times between bursts.

As a first step to using this method, the data $\mathbf{D}_{(i,j)}$ for $i > 1$ are split to two groups G_1 and G_2 according to their sampling times. The first group G_1 contains data in the time interval $[\tau_{(i\text{-}1,j)}, \tau_{(i,j+1)}]$ and the second group G_2 consisting of data in the time interval $[\tau_{(i,j)}, \tau_{(i\text{-}1,j)}]$. For e.g. consider $\mathbf{D}_{(2,\ 3)}$ which is the data in the 3rd burst of $\mathbf{P}_2$. The data within $\mathbf{D}_{(2,\ 3)}$ is split into G_1 and G_2 where G_1 contains information of $\mathbf{D}_{(2,\ 3)}$ in the time interval that corresponds to $[\tau_{(1,3)}, \tau_{(2,4)}]$ and G_2 contains data of $\mathbf{D}_{(2,\ 3)}$ the corresponds to $[\tau_{(2,3)}, \tau_{(1,3)}]$. The two groups are marked in figure Fig. 8(b). Data of the first group are sent earlier than the data of the second group. Similarly, $\mathbf{D}_{(1,j)}$ are transmitted as two groups containing data within $[\tau_{(k,j)}, \tau_{(1,j+1)}]$ and $[\tau_{(1,j)}, \tau_{(k,j)}]$, and packet of the first of these groups is sent prior to the packets of the second group.

A receiver R_M that desires to consume $\mathbf{P}_M$ receives the first time-sliced burst of any $\mathbf{P}_\mathrm{i}$ in $\mathbf{S}$ that carries data either commensurate with or inferior to its capabilities. In practice, it is also possible that a receiver can also tune in to $\mathbf{D}_{(i,j)}$, $i > M$, and decode some contents of $\mathbf{D}_{(i,j)}$. These contents could be say the audio track or intra pictures only. However, to simplify analysis, it is assumed here that $M = k$. Hence, R_M is able to tune into any burst $\mathbf{D}_{(i,j)}$. The expected probability $P(E_{SIM})$ that τ_t is during any of the bursts $\mathbf{D}_{(i,j)}$ becomes

$$P\left(E_{SIM}\right) = k \times t_{ADT} / \delta \tag{16}$$

In order to analyze the reduction of $\Delta_{(T\text{-}IN)}$ if τ_t is during $\mathbf{D}_{(i,j)}$, two cases are considered separately. If data reception starts during $[\tau_{(i\text{-}1,j)}, \tau_{(i,j+1)}]$: $i > 1$, then all data in the time-interval $[\tau_{(i,j)}, \tau_{(i\text{-}1,j)}]$ is received provided that no transmission errors occur. Similarly, if data reception starts during $[\tau_{(k,j)}, \tau_{(1,j+1)}]$: $i = 1$, then all data in the time-interval $[\tau_{(1,j)}, \tau_{(k,j)}]$ is received. The next received burst contains the immediate subsequent period of data. Therefore, playback is continuous. This tune-in time range is referred as the second playback portion of the burst and denoted with a subscript *bs* in the delay components. The subscript *bf* is used to denote that the reception started during the first playback portion of the burst, i.e., the period of $[\tau_{(i,j)}, \tau_{(i\text{-}1,j)}]$: $i > 1$ or $[\tau_{(1,j)}, \tau_{(k,j)}]$: $i = 1$.

It can been seen that, $\Delta_{(T\text{-}SYNC),bs} = 0$ and $\Delta_{(COMP),bs} = 0$. The expected reception duration of the first burst and the expected probability of tuning in during the second playback portion of the burst are inversely proportional to the number of simulcast programs k in the service and is given by

$$E\left[\Delta_{(RCPT,bs)}\right] = t_{ADT}\left(k+1\right)/2k \tag{17}$$

$$P\left(E_{bs}\right) = P\left(E_{SIM}\right)\left(1-\left(1/k\right)\right) \tag{18}$$

When the tune-in occurs during the first playback portion of a burst, some data of the burst would be useful and hence $\Delta_{(T\text{-}SYNC,bf)} = 0$. However, as τ_t within the first playback portion is arbitrary, the expected delay to achieve playback with any pause is

$$E\left[\Delta_{(COMP,bf)}\right] = \left(\delta/k\right)/2 \tag{19}$$

The expected reception duration of the first burst and the expected probability of tuning in during the first playback portion are given by

$$E\left[\Delta_{(RCPT,bf)}\right] = \left(t_{ADT}/k\right)/2 \tag{20}$$

$$P\left(E_{bf}\right) = P\left(E_{SIM}\right)/k = t_{ADT}/\delta \tag{21}$$

The value $E\left[\Delta_{(T-SYNC)}\right]$ is computed by

$$E\left[\Delta_{(T-SYNC)}\right] = \left(\Delta t_e - t_{ADT}(k-1)\right)/2k \tag{22}$$

V. Redundant Intra Picture Insertion (RIPI)

In H.264/AVC, a coded picture can be either a primary coded picture or a redundant coded picture. A primary coded picture is used to decode valid bitstreams. A redundant coded picture is decoded only when the decoding of the primary coded picture fails. A redundant coded picture may be the entire picture or only a subset of macroblocks in a picture. Furthermore, redundant pictures can also have different selections of macroblock modes and/or different reference pictures compared to the primary coded picture. The use of redundant coded pictures for improving error robustness has been studied e.g. in [14][15][16].

Redundant intra picture insertion (RIPI) can be used to reduce the decoder refresh delay component $\Delta_{(REFRSH)}$. A regular bitstream is encoded with a restriction that every s^{th} picture (here called S picture) is coded such that no picture succeeding the S picture, is inter-predicted from any picture preceding the S picture, all in decoding order. When temporal scalability is used, a more general scheme, where the constraint for S pictures applies only to the lowest level in the temporal scalability hierarchy can be used to implement RIPI. This constraint of S pictures causes a negligible drop in compression efficiency. Depending on the frequency of S pictures, this drop in compression efficiency is typically in the range of 0.05 to 0.1 dB in terms of mean luma PSNR, when compared to an unconstrained bitstream [12]. The interval s defines the granularity at which random access points can be provided in the bitstream. An S picture is a regular reference picture at the lowest temporal level and can be of any coding type. For every S picture, an intra-coded redundant picture is also encoded. The coded redundant picture can be of lower quality (higher quantization step size) compared to the S picture. The encoded bitstream is transmitted to an IP encapsulator that also composes the $\mathbf{M}_{MPE\text{-}FEC}$, among other things. For every $\mathbf{M}_{MPE\text{-}FEC}$ composed, only the first intra-coded redundant picture, in decoding order, is placed in the ADT. All other intra-coded redundant pictures in the bitstream are removed. Hence, the bitrate increase due to the redundant intra-coded picture remains moderate. This increase is however, significantly lower compared to the bitrate increase that would result with periodical coding of primary intra pictures at the frequency of S pictures.

At τ_t, receivers can either start decoding from the beginning of the first GOP, or the first redundant intra-coded picture, whichever is comes earlier in decoding order. If the decoding is started from a redundant intra-coded picture, there is a mismatch in the reconstructed pixel values, compared to the case where the decoding started from an IDR picture preceding the current burst. However, the drift only propagates until the next primary IDR picture occurs in the bitstream and is therefore not likely to be annoying. The mismatch could be avoided altogether by using the SP coding type for S pictures and SI coding type for the redundant intra-coded pictures.

Assuming that the IP encapsulator makes no effort to align S pictures with the boundaries of bursts, the expected value of $\Delta_{(REFRSH)}$ is derived as

$$E\left[\Delta_{(REFRSH)}\right]=s/2f \qquad (23)$$

where f is the picture rate of coded pictures, assuming a constant picture rate.

VI. Analysis of Tune-In Delay Reduction

This section analyses and compares the practical expected tune-in delays expected with overlapped time-interleaved simulcast (OTIS), redundant intra picture insertion (RIPI), combined OTIS and RIPI, and conventional time-sliced transmission over DVB-H. The algorithms CS and OTIS with and without RIPI are now theoretically analyzed.

The average bitrate B_p for all programs $\mathbf{P}_i$ in the simulcast session was fixed at 352 kbps. The video frame rate was fixed at 25 frames per second. $\Delta_{(T\text{-}IN)}$ in all cases is then computed using Equation (4). The delay $\Delta_{(FEC)}$ was set to zero, because $\mathbf{P}_i \,\forall\, i$ was assumed to be of constant bitrate = B_p and the code rate was fixed for all burst. Therefore, there is variation in the amount of data that is transmitted in a burst. The delay $\Delta_{(M\text{-}SYNC)}$ was also zero because it was assumed that all bursts contained all media

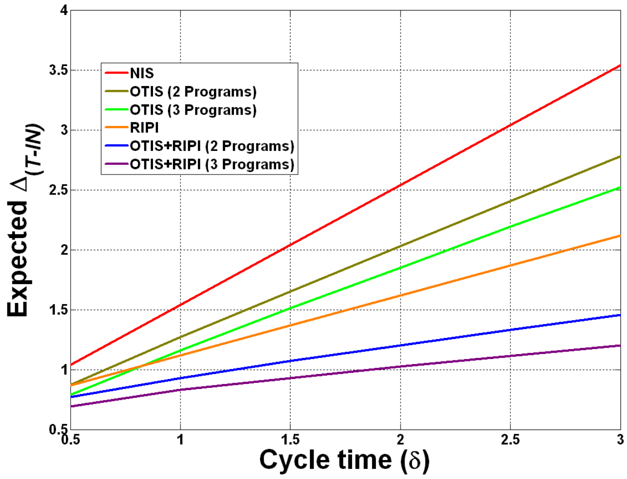

Fig. 9: The theoretical expected tune in delay plotted as a function of the cycle time

data for the burst playback period. The processing delays of the receiver and player implementations $\Delta_{(PROC)}$ was also set to zero. After the above assumptions and Equation (4) the expected $\Delta_{(T\text{-}IN)}$ is

$$\begin{aligned}E\left[\Delta_{(T-IN)}\right]&=E\left[\Delta_{(T-SYNC)}\right]+E\left[\Delta_{(COMP)}\right]\\&+E\left[\Delta_{(RCPT)}\right]+E\left[\Delta_{(REFRSH)}\right]+E\left[\Delta_{(VBR-COMP)}\right]\end{aligned} \qquad (24)$$

The value of E[$\Delta_{(VBR\text{-}COMP)}$] was set to 0.4, which is a typical value obtained from [17]. The expected values E[$\Delta_{(T\text{-}IN)}$] for NIS, OTIS and OTIS+ RIPI are computed and plotted against the different δ and the curves in Fig. 9 is obtained.

Table 1: Tune-in delay reduction percentages compared to NIS

	0.5	1	1.5	2	2.5	3
OTIS (2Prg)	15.90	17.50	19.00	20.10	20.80	21.40
OTIS (3Prg)	23.70	24.20	25.80	27.00	27.90	28.60
RIPI	16.30	27.30	32.80	36.20	38.50	40.10
OTIS+ RIPI (2Prg)	25.70	39.40	47.50	52.70	56.20	58.80
OTIS+ RIPI (3Prg)	33.50	46.10	54.30	59.60	63.30	66.10

From the curves in Fig. 9 it can be seen that among the methods discussed, OTIS with RIPI performs the best compared to NIS. As the number of programs k in the session increases, the tune-in delay reduces. However this is at the cost of addition bitrate overhead.

The curves plotted in Fig. 9 and the Table 1 are theoretical results after simplification. Further experimental research is necessary to verify these theoretical results.

VII. Conclusions

A receiver that either tunes into a DVB-H program or changes from one program to another, experiences a delay before the desired program can be rendered. This delay is called the tune-in delay or channel-switching delay. The components of tune-in delay were identified and mathematically analyzed in this paper. To maximize the perceived media quality in heterogeneous receivers, simulcasting was used. Receivers are grouped into classes and each class of receivers consumes media that best matches its capabilities. In order to reduce the tune-in delay, the time-sliced bursts of the simulcast programs of the same content were transmitted maximally apart from each other. Moreover, the redundant intra picture insertion (RIPI) method was proposed for DVB-H systems in which encoding and time-slicing are not synchronized. In RIPI, the encoder provides a redundant intra picture stream, from which only the first one in each time-sliced burst is selected for transmission over-the-air. The proposed methods were theoretically proved to reduce the tune-in delay by as much as 60%. Further experimentation is required to verify these results in a practical context.

REFERENCES

[1] ETSI Technical Specification 302 304, "Digital Video Broadcasting (DVB): Transmission systems for handheld terminals," V1.1.1, 2004.

[2] ETSI Technical Specification 102 005, "Digital Video Broadcasting (DVB); specification for the use of video and audio coding in DVB services delivered directly over IP protocols".

[3] ISO/IEC 14496–10:2003, "Coding of audiovisual objects—part 10: advanced video coding," 2003, also ITU-T Recommendation H.264 "Advanced video coding for generic audiovisual services."

[4] Y. Wang, J. Ostermann, and Y.-Q. Zhang, "Video Processing and Communications," Englewood Cliffs, NJ: Prentice-Hall, Sep. 2001.

[5] ITU-T Draft Recommendation H.264 Amendment 3, "Advanced video coding for generic audiovisual services: scalable video coding," 2007.

[6] S. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," Proc. of IEEE INFOCOM'96, pp. 553–560, Mar. 1996.

[7] ETSI Technical Recommendation 101 202, "Digital Video Broadcasting (DVB): Implementation guidelines for data broadcasting," V1.2.1, 2003.

[8] D. Tian, V.K. Malamal Vadakital, M.M. Hannuksela, S. Wenger, and M. Gabbouj, "Improved H.264/AVC video broadcast/multicast," Proceedings of SPIE, vol. 5960 (Proc. of Visual Communications and Image Processing), pp. 71-82, Jul. 2005.

[9] ETSI Technical Recommendation 102 377, "Digital Video Broadcasting (DVB); DVB-H implementation guidelines".

[10] ETSI Technical Specification 102 472, "Digital Video Broadcasting (DVB); IP datacast over DVB-H; content delivery protocols".

[11] J. M. Boyce and A. M. Tourapis, "Fast efficient channel change," Proc. of IEEE Int. Conf. on Consumer Electronics (ICCE), pp. 1-2, Jan. 2005.

[12] M. Rezaei, M.M. Hannuksela, and M. Gabbouj, "Tune-in time reduction in video streaming over DVB-H," IEEE Transactions on Broadcasting, vol. 53, no. 1, part 2, pp. 320-328, Mar. 2007.

[13] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 637-644, Jul. 2003.

[14] C. Zhu, Y.-K. Wang, M.M. Hannuksela, and H. Li, "Error resilient video coding using redundant pictures," Proc. of IEEE Int. Conf. on Image Processing (ICIP), pp. 801-804, Oct. 2006.

[15] Y.-K. Wang, C. Zhu, and H. Li, "Error resilient video coding using flexible reference frames," Proceedings of SPIE, vol. 5960 (Proc. of Visual Communications and Image Processing), pp. 691-702, Jul. 2005.

[16] Y.-K. Wang, M. M. Hannuksela, and M. Gabbouj, "Error resilient video coding using unequally protected key pictures," Proc. of Int. Workshop VLBV03, Lecture Notes in Computer Science, vol. 2849/2003, pp. 290-297, Sep. 2003.

[17] M. Rezaei, I. Bouazizi, V.K. Malamal Vadakital, M. Gabbouj, "Optimal Channel Changing Delay for Mobile TV over DVB-H," Proc. of IEEE Int. Conf. on Portable Information Devices (PORTABLE), pp. 1-5, March 2007.