Prediction and Transmission Optimization of Video Guaranteeing a Bounded Zapping-Delay in DVB-H

Vinod Kumar Malamal Vadakital and Moncef Gabbouj, *Fellow, IEEE*

***Abstract*—Zapping-delay is an important factor contributing to the quality-of-experience (QoE) of any multimedia service; the longer the zapping-delay, the worse is the QoE. Therefore, for a good QoE, it is necessary to bind zapping-delay within acceptable bounds. Coded video uses a motion compensated prediction structure that contributes to the zapping-delay. In DVB-H, zapping-delay is further increased because of the time-sliced burst transmission architecture. Hence, any attempt to bind zapping-delay must consider both the prediction structure of coded video, as well as the time-sliced transmission architecture of DVB-H. This paper analyzes prediction structure of video as binary relations generating a prediction graph. Reachability concepts of graph theory are then applied to construct binary valued indicator functions. These indicator functions inform about the reception, decoding, and play-out states of each access-unit in the video sequences. zapping-delay is studied by dissecting the components that it is composed off. Conditions for minimizing each of them individually, while the final zapping-delay does not exceed a known bound, is found. Emphasis is also laid on gradual quality enhancement after the display of the new program has started.**

***Index Terms*—Broadcast channels, DVB-H, H.264/AVC, video processing, zapping-delay.**

I. Introduction

MOBILE television broadcasting to hand-held devices is enabled by DVB-H [4]. DVB-H improves upon its predecessor, DVB-T [3], to enable data broadcasting to hand-held devices. Time-slicing, and Reed-Solomon (RS) based link-layer forward error correction (FEC) codes, are some new features added by DVB-H to DVB-T. These features were essential to improve power utilization, and error resiliency.

The link-layer FEC used by DVB-H, is called Multi-Protocol Encapsulation-Forward Error Correction (MPE-FEC). RS based FEC is calculated over some number of source IP packets, called the source-block. Data of each source block is interleaved in a matrix, called the MPE-FEC matrix, before FEC is computed. The source IP packets, along with the computed FEC data, are collectively called the FEC-block. The FEC-block is then transmitted as a burst. Each burst is called a time-slice, and the process is called time-slicing.

Time-slicing was introduced in the standard to reduce power consumption. It has been shown to improve power efficiency by as much as 90% [6]. The bursts are transmitted at a much higher bit-rate than the average bit-rate of the data stream. A receiver powers-on only when the time-slice it needs, is on the air. Otherwise, it goes into a stand-by mode. The frequent switching-on and switching-off of the receiver reduces terminal power consumption.

Mobile-television is considered as the primary use-case for DVB-H. A DVB-H broadcast can carry many individual (television) programs. With many choices of programs, users tend to surf across programs. This phenomenon is called *zapping*. A *zapping-event* is an event initiated by the user, when the display of the currently consumed program is replaced by another. In literature, zapping is also called *channel-surfing*, *channel-changing*, *channel-switching* and many more. This paper uses the nomenclature zapping.

In a digital broadcast, zapping is not immediate. More often than not, there is some finite delay, called the zapping delay. Zapping-delay is the time elapsed after a zapping-event is triggered, until a picture of the chosen program is displayed without errors. The predominant cause for zapping-delay is due to the way video is compressed, coded, and transmitted.

The recommended media codecs for DVB-H have been specified in [5]. Among them, H.264/AVC [9] is a recommended video codec. H.264/AVC uses block based transform coding, along with *motion compensated prediction* (MCP), for compression. While the use of MCP provides very good coding efficiency, the prediction structure it uses, destroys the independent decoding capability of access-units. An access-unit, in this paper, refers to the coded representation of one full picture of a video sequence. Therefore, there is a constraint on the random-access decoding capability of coded video, which contributes to zapping-delay.

When block based FEC is used, the entire FEC block must be received, before decoding can commence. This is because the whole FEC block is needed to correct any reception error of the block. This delay is called the FEC decoding delay. MPE-FEC, being a block based FEC, also has a finite FEC decoding delay, which also contributes to the zapping-delay.

Furthermore, long inter-burst times between time-slices also contributes to zapping-delay. Long inter-burst times is needed for power conservation. However, when a zapping-event occurs during the interval between two bursts, no relevant data is available for decoding. A receiver has to wait until the next burst is

Manuscript received September 13, 2010; revised February 02, 2011; accepted February 21, 2011. Date of publication April 19, 2011; date of current version May 25, 2011. This work was supported by the Academy of Finland under Project 129657 (Finish Centre of Excellence program 2006–2011).

The authors are with the Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland (e-mail: vinod.malamalvadakital@tut.fi; moncef.gabbouj@tut.fi).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2011.2127570

received, decoded and a random-access (Intra Decoder Refresh) access-units is found in the decoded time-slice.

Zapping-delay has a direct affect on *Quality-of-Experience* (QoE) [10]. Hence, there is a need to limit zapping-delay within some bound. For a bounded zapping delay, finding optimal prediction structures and transmission order is necessary. This paper only deals with the scenario when a zapping-event occurs during the reception of a burst. The main delay components contributing to zapping-delay are listed and analyzed. Theoretical solutions are provided to minimize each of these components, such that the final zapping-delay is bounded. These theoretical solutions, apart from providing a bounded zapping-delay, do so by gradually enhancing quality.

The process of optimization starts with an analysis of video prediction structures, and their reception in time-sliced bursts. Graph theoretic principles are used to analyze prediction structures. A video sequence is formulated as a binary relation over the set of access-units it contains. This relation generates a prediction graph, which is a *directed acyclic graph* (DAG). Reception, decoding, and play-out of access-units in a video sequence are expressed in terms of binary valued indicator functions. Given a zapping-event time, connectivity concepts of graph theory are used to construct these indicator functions. Zapping-delay is studied in terms of its component contributors. Conditions for minimizing each of them individually, while the over-all zapping-delay does not exceed a known bound, are discussed.

While optimizing predictions structures, emphasis is also laid on optimizing QoE. A gradual enhancement of presentation quality is assumed to improve QoE. For this purpose, a criterion for *gradual frame-rate enhancement* (GFRE), after the display of the new program starts, is defined. GFRE is first formulated as an in-equality of access-unit play-out durations. Among all prediction structures that satisfy this inequality, an optimization measure that selects the best among them, is found.

The paper is organized as follows. Section II introduces the concept of MCP and time-slicing using simple relational and graph-theoretic notations. The problem is formulated in Section III. Prior research in this area is presented in Section IV. Methodology used for optimization is presented in Section V. System implementation considering two different transmission scenarios are presented in Section VI. The simulations for optimization, and bound estimation are detailed in Section VII. Finally, the paper ends with the conclusions and future work in Section VIII.

II. Preliminaries

A. Prediction Structures Formulated as a Relation

The core paradigm of video compression is minimizing redundancies and irrelevancies. MCP is a temporal redundancy reducing technique, used extensively by hybrid video coders. In this paper, the smallest unit of a prediction structure is assumed to be an access-unit. Then, due to motion compensated prediction, a coded video sequence is a binary relation on the set of access-units; The relation is named "*Predicted From*".

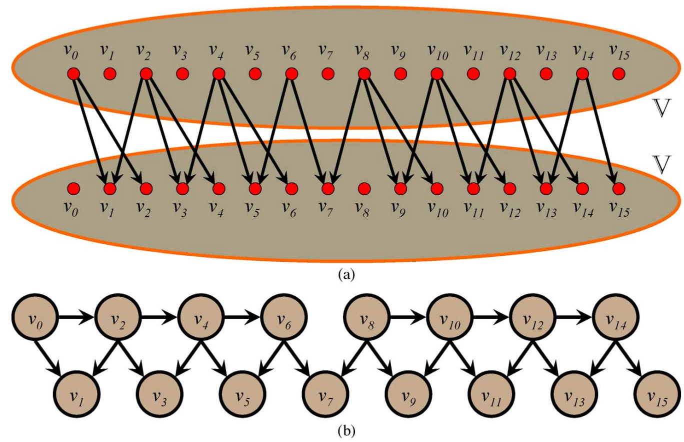

Fig. 1. A video sequence, $R_G^{\mathbb{V}^2}$, with 16 access-units depicted as a map, and a graph: (a) $R_G^{\mathbb{V}^2}$ as ordered pairs in $\mathbb{V}^2$; (b) graph G of the relation $R_G^{\mathbb{V}^2}$.

Let $\mathbb{V} = \{v_i : i \in \mathbb{Z}^+\}$ be the set of access-units, and $\mathbb{V}^2 = \{(v_i, v_j) : \{v_i, v_j\} \in \mathbb{V}; i, j \in \mathbb{Z}^+\}$ be the ordered pairs in the Cartesian product $\mathbb{V} \times \mathbb{V}$.

Definition 1 (Prediction Formulated as a Relation): The relation "Predicted From" is a three tuple, $R_G^{\mathbb{V}^2} = (\mathbb{V}, \mathbb{V}, G)$, where $G = (\mathbb{V}, \mathbb{A})$ is a directed acyclic graph (DAG) generated by the relation. The set, $\mathbb{A}$, is called the arc-set. It is a sub-set of $\mathbb{V}^2$. The graph G does not contain loops or cycles.

Example 1: Consider a video sequence with 16 access-units. This video sequence can be expressed as a binary relation, $R_G^{\mathbb{V}^2}$. An example mapping between access-units of the sequence is shown in Fig. 1(a). The relation generates a directed graph $G = (\mathbb{V}, \mathbb{A})$, where $\mathbb{A}$ is the arc-set. The graph G, with access-units in $\mathbb{V}$, and arcs in $\mathbb{A}$, is shown in Fig. 1(b). In the figure, access-units are depicted as nodes, and arcs as directed lines between nodes.

Henceforth, in this paper, a video sequence implies the relation $R_G^{\mathbb{V}^2}$, and vice-versa. The total number of access-units in $\mathbb{V}$ is denoted as $|\mathbb{V}|$.

B. Classes of Access Units

In a video sequence, every access-unit can be assigned into one of two mutually exclusive classes. Class membership is decided based on the access-unit type. The first, is the "*Intra access-unit*"; An Intra access-unit is not predicted from any other access-unit. The second is the "*Predicted access-unit*"; A predicted access-unit uses for prediction, other access-units, either of the Intra type or the predicted type. Thus, $\mathbb{V}$ can be partitioned such that,

$$\mathbb{V} = \left\{\mathbb{V}^{\mathbf{I}} \cup \mathbb{V}^{\mathbf{P}} : |\mathbb{V}| = |\mathbb{V}^{\mathbf{I}}| + |\mathbb{V}^{\mathbf{P}}|\right\}, \tag{1}$$

where, $\mathbb{V}^{\mathbf{I}}$ is the set of Intra access-unit, and $\mathbb{V}^{\mathbf{P}}$ the set of predicted access-unit. Given $R_G^{\mathbb{V}^2}$, the access-units are classified by applying the following rules:

- $v_k \in \mathbb{V}^{\mathbf{I}}$, only if there are no arcs directed to v_k, in G.
- $v_k \in \mathbb{V}^{\mathbf{P}}$, if there is at least one directed arc to v_k, in G.

Example 2: In the graph shown in Fig. 1(b), $\mathbb{V} = \{\mathbb{V}^{\mathbf{I}}, \mathbb{V}^{\mathbf{P}}\}$, where,

$$\begin{aligned}\mathbb{V}^{\mathbf{I}} &= \{v_0, v_8\}\\ \mathbb{V}^{\mathbf{P}} &= \mathbb{V} - \mathbb{V}^{\mathbf{I}}\\ &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}\end{aligned}$$

C. Condition for Decoding an Access-Unit Without Errors

An access-unit can be decoded without errors, only if all other access-units that it is predicted from (either directly or indirectly), can be decoded without errors. To formulate this more precisely, concepts of "*path*", "*full-path*", and "*all-path*" are needed. These concepts are related to the "connectivity" notion of graph theory [2], and their definitions follow.

Definition 2 (Path): Given $R_G^{\mathbb{V}^2} = (\mathbb{V}, \mathbb{V}, G)$ a path between two access-units v_t and v_h is a sub-graph, $G^{\mathcal{P}}(v_t, v_h) = (\mathbb{V}^{\mathcal{P}}, \mathbb{A}^{\mathcal{P}})$, of G, such that

- $\mathbb{V}^{\mathcal{P}} \subseteq \mathbb{V}$,
- $|\mathbb{V}^{\mathcal{P}}| \leq |\mathbb{V}|$,
- $\mathbb{A}^{\mathcal{P}} \subseteq \mathbb{A}$,
- $\{v_t, v_h\} \subseteq \mathbb{V}^{\mathcal{P}}$,
- There exist a sequence $X = (x_0, \ldots, x_{(n-1)})$ with a bijection $\mathbb{V}^{\mathcal{P}} \rightarrow X$ such that,
  — $\mathbb{A}^{\mathcal{P}} = \{(x_0 x_1), (x_1 x_2), \ldots, (x_{(n-2)} x_{(n-1)})\}$,
  — $x_i \in X : 0 \leq i \leq (n-1)$,
  — $x_0 = v_t$, and $x_{(n-1)} = v_h$.
  For a given v_t and v_h, there can be many paths, $\{G^{\mathcal{P}_i}(v_t, v_h) : i \in \mathbb{Z}^+\}$ from v_t to v_h.

Example 3: In the graph shown in Fig. 1(b), for a path $G^{\mathcal{P}}(v_2, v_5)$

- $\mathbb{V}^{\mathcal{P}} = \{v_2, v_4, v_6, v_5\}$
- $\mathbb{A}^{\mathcal{P}} = \{(v_2, v_4), (v_4, v_6), (v_6, v_5)\}$
- $\mathbb{V}^{\mathcal{P}} \subseteq \mathbb{V}$
- $\mathbb{A}^{\mathcal{P}} \subseteq \mathbb{A}$
- $X = \{x_0, x_1, x_2, x_3\}$, such that$v_2 = x_0, v_4 = x_1, v_6 = x_2, v_5 = x_3$

There are two paths from v_2 to v_5 (see Fig. 2).

- $G^{\mathcal{P}_0}(v_2, v_5) = (\mathbb{V}^{\mathcal{P}_0}, \mathbb{A}^{\mathcal{P}_0})$
  — $\mathbb{V}^{\mathcal{P}_0} = \{v_2, v_4, v_6, v_5\}$
  — $\mathbb{A}^{\mathcal{P}_0} = \{(v_2, v_4), (v_4, v_6), (v_6, v_5)\}$
- $G^{\mathcal{P}_1}(v_2, v_5) = (\mathbb{V}^{\mathcal{P}_1}, \mathbb{A}^{\mathcal{P}_1})$
  — $\mathbb{V}^{\mathcal{P}_1} = \{v_2, v_4, v_5\}$
  — $\mathbb{A}^{\mathcal{P}_1} = \{(v_2, v_4), (v_4, v_5)\}$

Definition 3 (Full-Path): Given $R_G^{\mathbb{V}^2} = (\mathbb{V}, \mathbb{V}, G)$, a full-path, $G^{\mathcal{F}}(v_k) = (\mathbb{V}^{\mathcal{F}}(v_k), \mathbb{A}^{\mathcal{F}}(v_k))$, of any $v_k \in \mathbb{V}$ is a sub-graph of G that satisfies

- $\mathbb{V}^{\mathcal{F}}(v_k) \subseteq \mathbb{V}$,
- $\mathbb{A}^{\mathcal{F}}(v_k) \subseteq \mathbb{A}$, and
- $G^{\mathcal{F}}(v_k) = G^{P}(v_t, v_k)$ where,$v_t \in \mathbb{V}^{\mathbf{I}}$.

In other words, a full-path to v_k, is a path from an Intra access-unit to v_k. If $v_k \in \mathbb{V}^{\mathbf{I}}$, then the full-path for v_k is a trivial graph. There can more than one full-paths, $\{G^{\mathcal{F}_i}(v_k) : i \in \mathbb{Z}^+\}$, to any given v_k.

Example 4: In the graph shown in Fig. 1(b), there are two full-paths to v_5 [see Fig. 3(a-b)]

- $G^{\mathcal{F}_0}(v_5) = (\mathbb{V}^{\mathcal{F}_0}(v_5), \mathbb{A}^{\mathcal{F}_0}(v_5))$
  — $\mathbb{V}^{\mathcal{F}_0}(v_5) = \{v_0, v_2, v_4, v_5\}$
  — $\mathbb{A}^{\mathcal{F}_0}(v_5) = \{(v_0, v_2), (v_2, v_4), (v_4, v_5)\}$
- $G^{\mathcal{F}_1}(v_5) = (\mathbb{V}^{\mathcal{F}_1}(v_5), \mathbb{A}^{\mathcal{F}_1}(v_5))$
  — $\mathbb{V}^{\mathcal{F}_1}(v_5) = \{v_0, v_2, v_4, v_6, v_5\}$
  — $\mathbb{A}^{\mathcal{F}_1}(v_5) = \{(v_0, v_2), (v_2, v_4), (v_4, v_6), (v_6, v_5)\}$

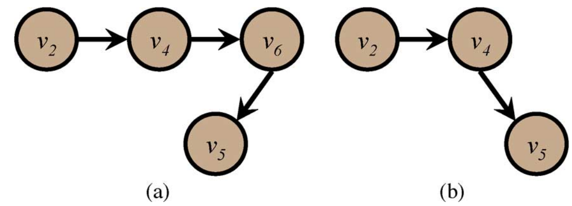

Fig. 2. Two paths from v_2 to v_5 for $R_G^{\mathbb{V}^2}$ shown in Fig. 1(b): (a) $G^{\mathcal{P}_0}(v_2, v_5)$; (b) $G^{\mathcal{P}_1}(v_2, v_5)$.

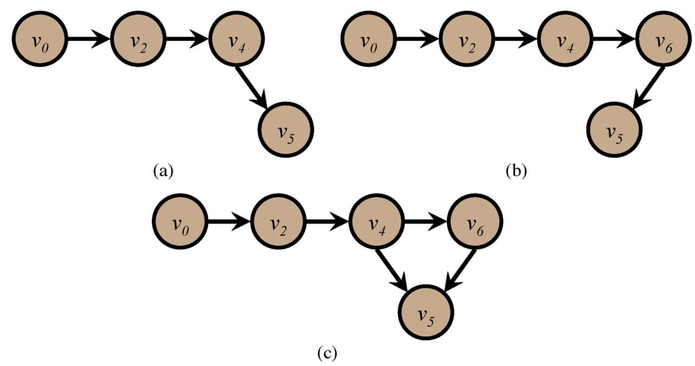

Fig. 3. Two full-paths sub-graphs, and an all-path to v_5, for $R_G^{\mathbb{V}^2}$, shown in Fig. 1(b): (a) $G^{\mathcal{F}_0}(v_5)$; (b) $G^{\mathcal{F}_1}(v_5)$; (c) $G^{\mathcal{A}}(v_5)$.

Definition 4 (All-Path): Given $R_G^{\mathbb{V}^2} = (\mathbb{V}, \mathbb{V}, G)$, an all-path, $G^{\mathcal{A}}(v_k) = (\mathbb{V}^{\mathcal{A}}(v_k), \mathbb{A}^{\mathcal{A}}(v_k))$, of any $v_k \in \mathbb{V}$ is a sub-graph of G that satisfies

- $\mathbb{V}^{\mathcal{A}}(v_k) \subseteq \mathbb{V}$,
- $\mathbb{A}^{\mathcal{A}}(v_k) \subseteq \mathbb{A}$,
- $\mathbb{V}^{\mathcal{A}}(v_k) = \{\mathbb{V}^{\mathcal{F}_0}(v_k) \cup \mathbb{V}^{\mathcal{F}_1}(v_k) \cup \cdots \cup \mathbb{V}^{\mathcal{F}_{n-1}}(v_k)\}$,
- $\mathbb{A}^{\mathcal{A}}(v_k) = \{\mathbb{A}^{\mathcal{F}_0}(v_k) \cup \mathbb{A}^{\mathcal{F}_1}(v_k) \cup \cdots \cup \mathbb{V}^{\mathcal{F}_{n-1}}(v_k)\}$
  where,
  —$\{G^{\mathcal{F}_0}(v_k), \ldots, G^{\mathcal{F}_{n-1}}(v_k)\}$ is the set of all full-path sub-graphs to v_k.

The value of n, which is the total number of full-paths to v_k, is obtained by applying the depth-first search algorithm for directed graphs.

Example 5: In the graph shown in Fig. 1(b) $G^{\mathcal{A}}(v_5) = (\mathbb{V}^{\mathcal{A}}(v_5), \mathbb{A}^{\mathcal{A}}(v_5))$ where,

- $\mathbb{V}^{\mathcal{F}_0}(v_5)$, $\mathbb{V}^{\mathcal{F}_0}(v_5)$, $\mathbb{A}^{\mathcal{F}_0}(v_5)$ and $\mathbb{A}^{\mathcal{F}_1}(v_5)$ are listed in the previous example.
- $\mathbb{V}^{\mathcal{A}}(v_5) = \{\mathbb{V}^{\mathcal{F}_0}(v_k) \cup \mathbb{V}^{\mathcal{F}_1}(v_k)\} = \{v_0, v_2, v_4, v_6, v_5\}$
- $\mathbb{A}^{\mathcal{A}}(v_5) = \{\mathbb{A}^{\mathcal{F}_0}(v_k) \cup \mathbb{A}^{\mathcal{F}_1}(v_k)\} = \{(v_0, v_2), (v_2, v_4), (v_4, v_6), (v_4, v_6), (v_6, v_5)\}$

For any given v_k with its all-path sub-graph $G^{\mathcal{A}}(v_k) = (\mathbb{V}^{\mathcal{A}}(v_k), \mathbb{A}^{\mathcal{A}}(v_k))$, the access-units belonging to $\mathbb{V}^{\mathcal{A}}(v_k)$ are the minimum necessary access-units that must be available at the decoder for proper decoding of v_k.

Let there be an indicator function $I^{\mathbf{R}}(v_i) : \mathbb{V} \rightarrow \{0, 1\}$, called the receive indicator (RI), for all $v_i \in \mathbb{V}$ of $R_G^{\mathbb{V}^2}$. $I^{\mathbf{R}}(v_i)$ informs if v_i is received/available for decoding. Then, another

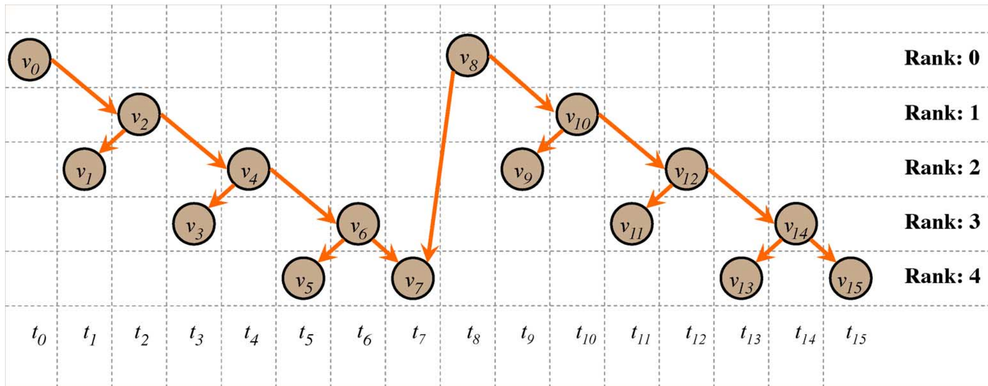

Fig. 4. Hasse diagram describing the poset of access-units, ordered based on decoding precedence, for the graph shown in Fig. 1(b).

indicator function, $I^{\mathbf{D}}(v_i) : \mathbb{V} \rightarrow \{0, 1\}$, called the decode indicator (DI), can be computed such that

$$I^{\mathbf{D}}(v_i) := \begin{cases} 1, & \text{iff } I^{\mathbf{R}}(v_k) = 1, \forall v_k \in \mathbb{V}^{\mathcal{A}}(v_i) \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

As a convention, both $I^{\mathbf{R}}(v_i)$ and $I^{\mathbf{D}}(v_i)$ are written in play-out order. In (2), the value of the indicator function $I^{\mathbf{D}}(v_i)$ is set to one, if and only if all access-units in its all-path graph are received without errors. Under all other conditions, v_i is decoded with errors.

Example 6: Consider the graph in Fig. 1(b). Let the RI sequence be

$$I^{\mathbf{R}} = (0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

which indicates that access-units v_0 to v_4 are not received/available and access-units v_5 to v_{15} are received/available. Then the DI sequence is

$$I^{\mathbf{D}} = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)$$

because, for all $v_k : 0 \leq k \leq 7$, $G^{\mathcal{A}}(v_k)$ contains v_0 in its all-path sub-graph, which is not received/available.

D. *Transitive Reduction of Prediction Structures*

A partially ordered set (poset) [17] is a set of pairs in which, for some pairs, a binary relation can be defined, indicating that one element in the pair precedes the other in the same pair. These relations are called partial order to emphasize the fact that not all pairs of elements in the set adhere to the relation criteria. Let the order relation on the set $\mathbb{V}$ be called "*decoding precedence*". This order relation forms a poset, $o(\mathbb{V}, \prec)$. More specifically, this order relation is a "strict partial order", because the elements of the poset $o(\mathbb{V}, \prec)$ follow irreflexivity rather than reflexivity. Henceforth, the $\prec$ is dropped from the poset notation.

The poset, $o(\mathbb{V})$, is visualized using Hasse diagrams. When constructing Hasse diagrams, the elements of the poset are represented as nodes. For a poset $o(\mathbb{V})$, if a node $v_j \prec v_i : v_j, v_i \in \mathbb{V}$, then the node v_j is placed lower than the node v_i. If there exist no v_k such that $v_j \prec v_k$ and $v_k \prec v_i$ then the nodes v_j and v_i are connected by a directed line from v_i to v_j. The other pairs in $o(\mathbb{V})$ correspond to directed paths.

The vertical ordering of access-units in a Hasse diagram is given by its rank. The rank for v_i in a poset $o(\mathbb{V})$ is obtained by assigning rank 0 to all minimal elements in $o(\mathbb{V})$, then removing these elements, assigning rank 1 to the minimal elements in that smaller poset, and so on:

$$\text{rank}(v_i) := \begin{cases} 0, & v_i \text{ min in } o(\mathbb{V}) \\ n, & v_i \text{ min in } o(\mathbb{V}) \setminus \{q : \text{rank}(q) < n\} \end{cases} \quad (3)$$

For the horizontal ordering, the nodes are placed such that for all $\{v_i, v_j\} \in \mathbb{V}$, v_i is placed to the left of v_j if $i < j$. Therefore, the horizontal ordering of access-units is nothing but the presentation order. The Hasse diagram for the predictions graph shown in Fig. 1(b) is presented in Fig. 4.

The following observations can be made from a Hasse diagram for any $R_G^{\mathbb{V}^2}$:

1) Hasse diagram is a graph, henceforth denoted by G'.
2) Every directed path of the graph G is a directed path G'.
3) For any path, $G^{\mathcal{P}}(v_t, v_h)$, all access-units $\{v_k : v_k \in \mathbb{V}^{\mathcal{P}} \setminus \{v_h\}\}$ satisfies $\text{rank}(v_k) < \text{rank}(v_h)$

Then, with the understanding that a full-path is a special case of a path, and an all-path is a union of full-paths, the following conclusion can be drawn:

1) For all $v_k \in \mathbb{V} : \text{rank}(v_k) > 0$, $v_j \in \{\mathbb{V}^{A}(v_k) \setminus \{v_k\}\}$ have $\text{rank}(v_j) < \text{rank}(v_k)$.
2) For all $v_k \in \mathbb{V} : \text{rank}(v_k) = 0$, $v_k \in \mathbb{V}^{\mathbf{I}}$.
3) For all $v_k \in \mathbb{V} : \text{rank}(v_k) > 0$, $I^{\mathbf{D}}(v_k) = 1$, if for all $v_j \in \mathbb{V} : \text{rank}(v_j) < \text{rank}(v_k)$, $I^{\mathbf{R}}(v_j) = 1$.

Given $R_G^{\mathbb{V}^2}$, a Hasse diagram G' is a visual representation of the transitive reduction of G. Since, G is an acyclic DAG, G' is unique for every $R_G^{\mathbb{V}^2}$. Therefore, the relation $R_G^{\mathbb{V}^2}$ can be written as $R_{G'}^{\mathbb{V}^2}$ without any loss of generality, to denote a video sequence with access units in $\mathbb{V}$ and generating a Hasse diagram G'.

E. *Coding and Structural Delay*

Most modern video encoders that use MCP allow the use of both past and future pictures as references for prediction. Prediction from a past picture is called forward prediction, while prediction from a future picture is called backward prediction. If a video coder uses only forward prediction for MCP, then it can start encoding the current picture without any delay. However, for backward prediction, the encoder has to wait until the future picture is available for encoding, encode the picture, and use the reconstructed version of the encoded future picture for MCP coding the current picture. This delay in encoding the current picture when backward prediction is used, is called the structural delay. Given $R_{G'}^{\mathbb{V}^2}$, the structural delay denoted by Δ^s, is given by

$$\Delta^s := \max_{(v_i,v_j)\in\mathbb{A}} (|j-i|). \quad (4)$$

F. Zapping-Event, Reception, and Decoding

When $R_{G'}^{\mathbb{V}^2}$ is transmitted through a channel, at the receiver, a non-overlapping time interval, $\delta^{\mathbf{R}}(v_i)$, which gives the time interval during which bits of v_i are received, exists for every $v_i \in \mathbb{V}$. This time interval is given by

$$\delta_{v_i}^{\mathbf{R}} := \left[t_s^{\mathbf{R}}(v_i), t_e^{\mathbf{R}}(v_i)\right), \quad (5)$$

where,

- $t_s^{\mathbf{R}}(v_i)$ and $t_e^{\mathbf{R}}(v_i) \in \mathbb{R}^+$ and,
- $t_s^{\mathbf{R}}(v_i) < t_e^{\mathbf{R}}(v_i)$

$t_s^{\mathbf{R}}(v_i)$ and $t_e^{\mathbf{R}}(v_i)$ denotes the start time and the end time of the interval $\delta^{\mathbf{R}}(v_i)$ for v_i respectively. Then the duration $\tau^{\mathbf{R}}(v_i)$ is given by

$$\tau^{\mathbf{R}}(v_i) := t_e^{\mathbf{R}}(v_i) - t_s^{\mathbf{R}}(v_i). \quad (6)$$

Since, $t_s^{\mathbf{R}}(v_i) < t_e^{\mathbf{R}}(v_i)$, $\tau^{\mathbf{R}}(v_i) \neq 0$ and $\tau^{\mathbf{R}}(v_i) \in \mathbb{R}^+$.

At the receiver, all data (assuming no channel errors) after a zapping-event, is considered received/available. Let $t^\bullet$ be the time when a zapping-event is triggered at the receiver. Then, for a video sequence $R_{G'}^{\mathbb{V}^2}$, $I^{\mathbf{R}}(v_i)$ is given by

$$I^{\mathbf{R}}(v_i) := \begin{cases} 1, & \text{iff } t^\bullet \leq t_s^{\mathbf{R}}(v_i) \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Knowing the RI function $I^{\mathbf{R}}(v_i)$, the DI function $I^{\mathbf{D}}(v_i)$ can be calculated using (2).

G. Zapping-Event and Play-Out

Every $v_i \in \mathbb{V}$ of the sequence $R_{G'}^{\mathbb{V}^2}$, is associated with a play-out time-interval $\delta^{\mathbf{P}}(v_i)$ given by,

$$\delta_{v_i}^{\mathbf{P}} := \left[t_s^{\mathbf{P}}(v_i), t_e^{\mathbf{P}}(v_i)\right), \quad (8)$$

and the duration, $\tau^{\mathbf{P}}(v_i)$, is given by

$$\tau^{\mathbf{P}}(v_i) := t_e^{\mathbf{P}}(v_i) - t_s^{\mathbf{P}}(v_i). \quad (9)$$

At the receiver, the RI function, $I^{\mathbf{R}}(v_i)$, and the DI function, $I^{\mathbf{D}}(v_i)$, is known using (7) and (2) respectively. Then, the play-out indicator (PI) function, $I^{\mathbf{P}}(v_i)$, is calculated as

$$I^{\mathbf{P}}(v_i) := \begin{cases} 1, & \text{iff } I^{\mathbf{D}}(v_i) = 1 \text{ and } t^\bullet \leq t_s^{\mathbf{P}}(v_i) \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

H. Zapping-Delay

After calculation of the PI function using (10), the first picture of the new program is presented at the time instant, $t^\times$, given by

$$t^\times := \mathop{\arg\min}_{\substack{v_i : v_i \in \mathbb{V}, \\ I^{\mathbf{P}}(v_i)=1}} \left(t_s^{\mathbf{P}}(v_i)\right). \quad (11)$$

Then, assuming $t^\bullet$ as the time when a zapping-event occurs, zapping-delay which is denoted by $\Delta^\bullet$, is computed as

$$\Delta^\bullet := t^\times - t^\bullet. \quad (12)$$

The following holds true for any $v_i \in \mathbb{V}$ that minimizes zapping-delay.

- $t^\bullet \leq t_s^{\mathbf{R}}(v_i)$
- $t^\bullet \leq t_s^{\mathbf{R}}(v_k),\ \forall v_k \in \mathbb{V}^{\mathcal{A}}(v_i)$.

I. Quality Aspects of Zapping

Quality-of-Experience (QoE) is a subjective measure of the over-all acceptability of a media program as perceived by the user. From previous studies [10], it has been established that a zapping-delay beyond some bound adversely affects QoE. Hence, there is a need to keep zapping-delay within acceptable bounds. In [12] the NorDig consortium has defined the zapping-delay to be bound to 1.5 seconds.

Zapping, as identified in DVB CM-IPTV, can be performed in two different ways. In the first case, after the time instant $t^\times$, the video is presented to the user at full quality (final quality of the coded video). In the second case, there is a transition period, called the zapping-transition, after $t^\times$ until some time instant $t^* : t^* \geq t^\times$, when the video is presented at reduced quality. After the time t^*, the video is presented at full quality.

Given the set of time instants, $\{t^\bullet, t^\times, t^*\}$, related to each other with the inequality $t^\bullet \leq t^\times \leq t^*$, the zapping-transition, Δ^* is defined by the equation

$$\Delta^* := t^* - t^\times. \quad (13)$$

A contribution of this paper, apart from the minimization of zapping-delay, also introduces the concept of gradual quality improvement during zapping-transition. In gradual quality improvement, at time instant $t^\times$, the presentation of the video starts with a reduced-quality, and gradually improves until the time instant t^*, when the presentation resumes at full-quality. The three different ways of performing zapping is conceptually presented in Fig. 5. In this figure, full-quality is denoted by Q_f, and reduced-quality is denoted by Q_r.

J. Gradual Frame-Rate Enhancement

In this paper, video frame-rate is the quality measure that is gradually improved during the zapping-transition period. This gradual improvement of frame-rate is termed as Gradual Frame-rate Enhancement (GFRE).

Let $I^{\mathbf{P}}(v_i)$ be the play-out indicator function for $R_{G'}^{\mathbb{V}^2}$, calculated after reception and decoding at the client. If f is the frame-rate at which $R_{G'}^{\mathbb{V}^2}$ was encoded, then ideally, play-out duration $\tau^{\mathbf{P}}(v_i)$ should be $1/f$ for all v_i with $I^{\mathbf{P}}(v_i) = 1$. This requires consecutive access-units, after zapping and in play-out order, to have their play-out indicator set. However, due to the prediction structure and a possible reception order modification of the transmitted video, this is not always true. A remedial action taken by the player is to account for the unplayable access-unit by continuing to present the previously displayed picture. This concealment method is called the freeze-frame concealment.

Let $\tau^{\mathbf{P}} = (\tau^{\mathbf{P}}(v_0), \tau^{\mathbf{P}}(v_1), \ldots, \tau^{\mathbf{P}}(v_{n-1}))$ be the play-out duration sequence, after freeze-frame concealment. The

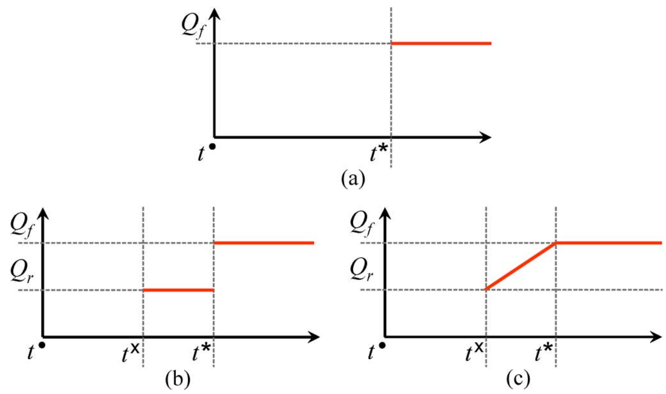

Fig. 5. Three ways of zapping: (a) zapping without transition period; (b) zapping with transition period; (c) gradual zapping in transition period.

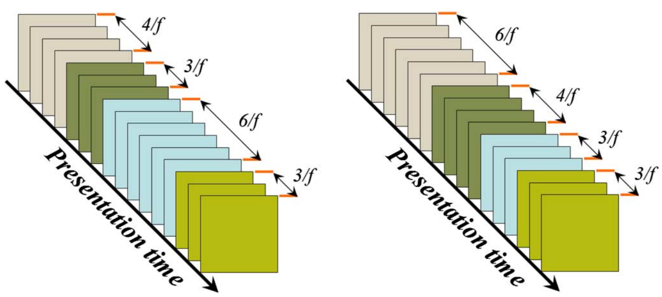

Fig. 6. Frame-rate variations for a presented video sequence: (a) presentation pattern that does not conform to Definition 5; (b) presentation pattern that follows Definition 5.

play-out duration sequence is ordered in play-out order. From this sequence $\tau^{\mathbf{P}}$, let those v_i which have their play-out duration as zero be removed. That is

$$\tau^{\mathbf{P}>\mathbf{0}} = \left(\tau^{\mathbf{P}} \setminus \left(\tau^{\mathbf{P}}(v_i) : \tau^{\mathbf{P}}(v_i) = 0\right)\right). \tag{14}$$

This forms a sub-sequence of $\tau^{\mathbf{P}}$ giving

$$\tau^{\mathbf{P}>\mathbf{0}} = \left(\tau^{\mathbf{P}}(v_{m_0}), \tau^{\mathbf{P}}(v_{m_1}), \ldots, \tau^{\mathbf{P}}(v_{m_{k-1}})\right),$$

where, the indices $m_0 < m_1 < \cdots < m_{k-1}$ are ordered according to the play-out order. Then the condition for GFRE is given below.

Definition 5: For any $R_{G'}^{\mathbb{V}^2}$, a GFRE is achieved, only if the access-units of the sequence are received such that for any $I^{\mathbf{R}}(v_i) : v_i \in \mathbb{V}$, the sequence $\tau^{\mathbf{P}>\mathbf{0}}$ follows the order condition

$$\tau^{\mathbf{P}>\mathbf{0}} = \left(\tau^{\mathbf{P}}(v_{m_0}) \le \tau^{\mathbf{P}}(v_{m_1}) \le \cdots \le \tau^{\mathbf{P}}(v_{m_{k-1}})\right) \tag{15}$$

where, $\tau^{\mathbf{P}>\mathbf{0}} = (\tau^{\mathbf{P}} \setminus (\tau^{\mathbf{P}}(v_i) : \tau^{\mathbf{P}}(v_i) = 0))$.

Example 7: Consider the two presentations pattern shown in Fig. 6. Both show a presentation of pattern with four presentable access-units $\{v_{m_0}, v_{m_1}, v_{m_2}, v_{m_3}\}$. However, in Fig. 6(a), $\tau^{\mathbf{P}>\mathbf{0}} = \{4/f, 3/f, 6/f, 3/f\}$ which does not conform to Definition 5. Where as in Fig. 6(b), $\tau^{\mathbf{P}>\mathbf{0}} = \{6/f, 4/f, 3/f, 3/f\}$ conforms to Definition 5. In the figures f is the play-out frame-rate of the sequence.

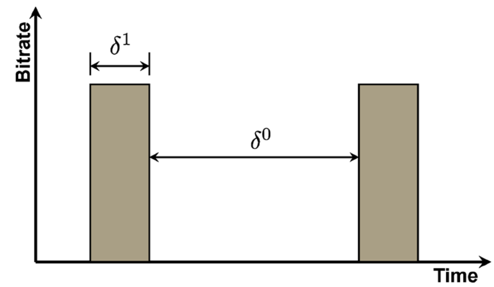

Fig. 7. Time-slicing in DVB-H.

K. Time-Sliced Transmission Design

Mobile hand-held terminals are battery operated and hence have limited power at their disposal. Receiving, de-modulating and decoding a data stream dissipates power. Therefore, the solution to reduce power dissipation lies in reducing the amount of time the terminal front-end is switched on. Time-slicing, along with time division multiplexing does exactly that. Time-slicing chops a continuous data stream into time-slices, with data of each time-slice being transmitted as bursts, at a much higher rate than the average data stream rate. Time-slice design also allows other programs to be time-division multiplexed in the broadcast. The terminal, upon choosing a program, the program-of-interest, needs to switch on its front-end only during the transmission of the burst carrying data of the program-of-interest, and switching it off otherwise.

Fig. 7 shows the time-sliced transmission design of a single program transmission. Two regions in time can be identified: The on-time $\delta^{\mathbf{1}}$ and the off-time $\delta^{\mathbf{0}}$. During $\delta^{\mathbf{1}}$, the receiver front-end is switched on and data is received. During $\delta^{\mathbf{0}}$, the receiver front-end is switched off, and the data received during the previous $\delta^{\mathbf{1}}$ is decoded and presented.

Let $R_{G'}^{\mathbb{V}^2}$ be the video sequence that is broadcasted using DVB-H. This video sequence is time-sliced with each time-sliced burst containing data from one MPE-FEC matrix. Time-slicing is then equivalent to segmenting the sequence $R_{G'}^{\mathbb{V}^2}$ into sub-sequences

$$R_{G'}^{\mathbb{V}^2} = \left({}^0R_{G'}^{\mathbb{V}^2} \bullet {}^1 R_{G'}^{\mathbb{V}^2} \bullet \cdots\right)$$

where, ${}^iR_{G'}^{\mathbb{V}^2} : i \in \mathbb{Z}^+$, is a sub-sequence transmitted in the ith time-slice burst. The symbol $\bullet$ denotes concatenation of sub-sequences to obtain the full sequence. The set of access-units transmitted in the kth burst is denoted by $\mathbb{V}_k$. Therefore, $\mathbb{V} = \{\mathbb{V}_0 \cup \mathbb{V}_1 \cup \cdots\}$. The kth sub-sequence generates the graph $G_k = (\mathbb{V}_k, \mathbb{A}_k)$. The set $\mathbb{V}_k$ can be partitioned into two disjoint sets: the set of Intra access-units $\mathbb{V}_k^{\mathbf{I}}$, and the set of predicted access-units $\mathbb{V}_k^{\mathbf{P}}$. Hence, $\mathbb{V}_k = \mathbb{V}_k^{\mathbf{I}} \cup \mathbb{V}_k^{\mathbf{P}}$. Transitive reduction of the graph G_k gives the Hasse diagram G'_k. Then, all definitions previously defined for $R_{G'}^{\mathbb{V}^2}$, is valid for ${}^kR_{G'}^{\mathbb{V}^2}$ also.

A zapping-event for a time-sliced transmission of $R_{G'}^{\mathbb{V}^2}$ can be separated into two cases: (i) zapping-event occurs during the transmission of a time-sliced burst, and (ii) zapping-event occurs during the off-period in between transmission of two time-sliced burst. The two cases can be separated by choosing intervals in the finite sequence

$$\left(\underbrace{[t_0,t_1)}_{\delta_0^{1^*}},\underbrace{[t_1,t_2)}_{\delta_0^{0}},\ldots,\underbrace{[t_{2i-2},t_{2i-1})}_{\delta_{i-1}^{1^*}},\underbrace{[t_{2i-1},t_{2i})}_{\delta_{i-1}^{0}},\ldots\right),$$

such that,

- $t_0 < t_1 \cdots < t_{2i} < \cdots$,
- $t_0 = t_s^{\mathbf{R}}(v_0)$,

and separating them as on-period intervals $\delta_k^{1^*}$ and off-period intervals δ_k^0 for a time-sliced burst k. A zapping-event is considered to have occurred during the reception of a burst, only if at least one access-unit of the burst is completely received. The notation $\delta_k^{1^*}$ is used instead of δ_k^1 to stress this fact. Then, the separation of the on-period from the off-period is done such that $\delta_i^{1^*} = \{[t_{2i}, t_{2i+1}) : i \in \mathbb{Z}^+\}$, and $\delta_i^0 = \{[t_{2i+1}, t_{2i+2}) : i \in \mathbb{Z}^+\}$.

III. Problem Formulation

Time-slicing combined with the prediction structure of video, causes zapping-delay. Consider a transmitted time-slice burst ${}^kR_{G'}^{\mathbb{V}^2}$. This burst is transmitted during the time-interval $[t_{2k}, t_{2k+1}) : k \in \mathbb{Z}^+$. Let the zapping-event occur at some time $t_{2k} \le t^{\bullet} < t_{2k+1}$.

Play-out of any access-unit of ${}^kR_{G'}^{\mathbb{V}^2}$ can start only after the sub-sequence has been fully received. This is because, if errors are encountered during the transmission of ${}^kR_{G'}^{\mathbb{V}^2}$, MPE-FEC error correction must be triggered. After, MPE-FEC error correction, access-units that are correctly decoded and are available for presentation can be displayed. The play-out time duration of the access-units transmitted in a burst should ideally be enough to cover the off-duration after the transmission of the burst.

In DVB-H, where a time-sliced burst encapsulates an FEC block, if the zapping-event occurs during the reception of a burst, there is a delay for data in the time-sliced burst to be received before decoding and play-out can commence. This delay is called "*Time-slice Reception Delay*", which is denoted by Δ^{α}. Fig. 8 pictorially shows the component delays that collectively adds up to form the zapping delay.

After, the burst is received, the RI, DI, and PI functions are computed. Depending on the sub-sequence received, there can be a further delay after reception to synchronize to the first access-unit that can be displayed. This delay is called the "*Play-out Synchronization Delay*" and denoted by Δ^{β}.

Apart from Δ^{α} and Δ^{β} contributing to the zapping-delay, there are several other factors that contributes to it. Program Specific Information (PSI)/System Information (SI) tables are required by DVB-H receivers to correctly zap into programs in a multiplex. In DVB-H, PSI/SI information is not time-sliced, and delivered at regular intervals (100 ms). Apart from PSI/SI tables conditional access (CA) and Digital Rights Management (DRM) information adds to the zapping-delay. In this paper only Time-slice Reception Delay, and Play-out Synchronization Delay are considered. All other factors are ignored due to their minimum contribution to zapping-delay.

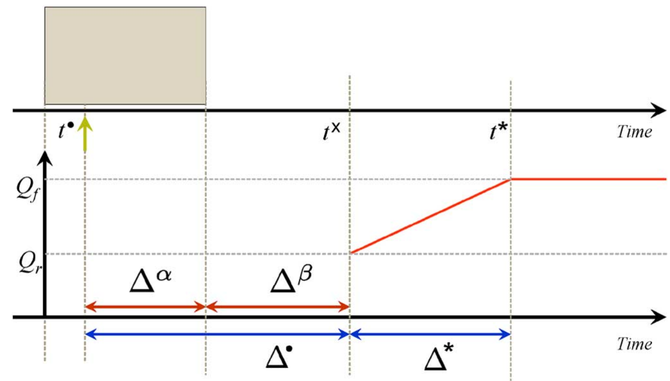

Fig. 8. Zapping delay along with its component delays.

IV. Related Work

A fast zapping method for set-top boxes is presented in [1]. Here, for any coded (primary) sequence, an auxiliary coded sequence is generated. This auxiliary sequence is a low-quality representation of the primary sequence, with frequent random access points (RAP). A RAP, in H.264/AVC, is an Intra Decoder Refresh (IDR) picture. In the primary sequence RAPs are coded far apart. When a zapping-event is triggered, the decoder synchronizes to the next RAP in the auxiliary sequence, to start decoding.

With the auxiliary stream approach, zapping-delay can definitely be reduced. However, this solution is not optimal in at least two ways. First, let the auxiliary bit stream contain only RAPs. The decoder, after a zapping-event, will synchronize with the next RAP, decode and place the decoded picture in the decoder picture buffer. The decoded picture is a low-quality representation of the corresponding picture in the primary sequence. The decoder then proceeds by decoding access-units from the primary sequence. Due to MCP, pictures that are decoded (from the primary sequence) after the decoded RAP, will contain errors. This error propagates until the next RAP is decoded from the primary sequence.

Next, let the auxiliary sequence also use predicted access-units for coding, apart from RAP. In this scenario, the best way a receiver can make use of the auxiliary sequence, is to synchronize to the next RAP in the auxiliary sequence, after a zapping-event, and continue decoding the access-units from the auxiliary stream, until a RAP in the primary stream is received. After a RAP in the primary stream is received, the decoder can continue decoding the primary stream. This is equivalent to zapping with a zapping-transition period. The QoE is then dependent on the duration of zapping-transition period, and the quality of the auxiliary stream presented during the zapping-transition.

An improved zapping method, specifically tailored for DVB-H, is presented in [14]. Here, during encoding, two encoders are employed to compress the same video sequence. One encoder generates a "*Splicable Bit-stream*" (SBS), while the other generates a "*Decoder Refresh Bit-stream*" (DRBS). The SBS encodes frequent splicable access-units. A spliceable access-unit, in [14], is defined as an access-unit which is encoded such that, no access-unit that follows it in presentation order, refers to an access-unit preceding it. The DRBS contains only RAPs corresponding to a spliceable access-unit. Access-units in DRBS are coded with an approximately similar quality to the corresponding spliceable access-unit in SBS. This is in contrast to the auxiliary stream method of [1], where RAPs in the auxiliary stream are coded at lower quality. Both DRBS and SBS are transmitted to the IP encapsulator. At the IP encapsulator the MPE-FEC matrix is constructed such that the first access-unit is presentation order is picked from the DRBS while the other access-units are from the SBS. Buffering requirements for this method of transmission is described in [15]. Verification of the method with experimental results, is established in [16]. This method requires no changes to the receiver/decoder.

Indeed, by ensuring that the first access-unit (in presentation order) of a sub-sequence transmitted in a burst is a RAP, this algorithm guarantees that if the entire burst is received, then zapping-delay is minimized. However, this is not true if the burst is partially received. An intelligent transmission/reception order of access-unit, as described in this paper, can also guarantee that zapping-delay is minimized when a burst is partially received.

Furthermore, to guarantee that the first access-unit transmitted is a RAP, the DRBS should encode all pictures in the sequence. Since the IP-encapsulator does not use all encoded access-units from the DRBS, this is an in-efficient use of system resources. The transcoding solution proposed in this paper reduces this in-efficiency.

Another approach, primarily devised for seem less handovers, but also useful in minimizing zapping-delay is presented in [13]. Here, a mobile terminal, apart from a DVB-H broadcast connection, also maintains a unicast bi-directional connection using a UMTS network. The DVB-H broadcast stream is also carried over the UMTS network. However, this in in-efficient use of the spectrum and does not scale well for mass multimedia broadcast.

Apart from the above mentioned works, the zapping-delay problem has also been researched in the context of IPTV. In [20] the authors analyze aspects of zapping-delay in the end-to-end transmission chain of IPTV. They also provide and discuss various solutions relevant to IPTV. Some introductory concepts presented in their work are also relevant to the method explored in this paper. Another recent work [11], also in the context of IPTV, uses a predictive tuning method to reduce zapping-delay. The authors propose a method that computes the most efficient number of pre-joining channels by estimating expected channel zapping time and expected bandwidth usage with a semi-Markov Process. The proposed method in [11] is complementary to the work presented in this paper. It can be used to minimize zapping-delay during the off periods of a time-sliced transmission.

V. Methodology

Let the maximum tolerable zapping-delay $\Delta^\bullet$ be fixed to some $\Delta^\Xi : \Delta^\Xi \in \mathbb{R}^+$. Then, the objective is to design a prediction structure and a corresponding optimal reception order, which adheres to the inequality,

$$\Delta^\bullet \leq \Delta^\Xi. \tag{16}$$

It has been already established that $\Delta^\bullet$ is the sum of Time-slice Reception Delay, Δ^α, and Play-out Synchronization Delay, Δ^β. Since, Δ^α and Δ^β are independent of each other, it is sufficient to minimize each of these component delays individually, such that the bound condition,

$$\left(\min_{\substack{\Delta^\alpha > 0,\\ \Delta^\alpha \in \mathbb{R}^+}} (\Delta^\alpha) + \min_{\Delta^\beta \in \mathbb{R}^+} (\Delta^\beta) \right) \leq \Delta^\Xi, \tag{17}$$

is met. Furthermore, after $\Delta^\bullet$, there is a zapping-transition Δ^*, where the objective is to maximize QoE using GFRE. The rest of this section is devoted to establishing the rules for minimizing zapping-delay.

Proposition: At least one Intra access-unit should be carried in a time-slice burst, for a successful zapping-event to occur after the reception of the burst, and before the reception of the next burst.

Proof: For a zapping-event to occur, at least one access-unit in the time-sliced burst must be displayed without errors. Consider the nth time-slice burst carrying the sub-sequence ${}^nR_{G'}^{\mathbb{V}^2}$. From (1), it is known that $\mathbb{V}_n$ can be partitioned into two sets, $\mathbb{V}_n^{\mathbf{I}}$, and $\mathbb{V}_n^{\mathbf{P}}$. Consider the situation where $\mathbb{V}_n^{\mathbf{I}} = \{\phi\}$. This implies, $\mathbb{V}_n = \mathbb{V}_n^{\mathbf{P}}$.

From (10), the minimum condition for any $v_k \in \mathbb{V}_n$ to be displayed without errors is that all $\mathbb{V}_n^{\mathcal{A}}(v_k)$ has $I_n^{\mathbf{D}}(v_k) = 1$. By definition, $\mathbb{V}_n^{\mathcal{A}}(v_k)$ contains at least one access-unit that is of the Intra type. If $\mathbb{V}_n^{\mathbf{I}} = \{\phi\}$, then, in the time-sliced burst, for all $v_k \in \mathbb{V}_n$, $I_n^{\mathbf{D}}(v_k) = 0$. Hence, a successful zapping-event cannot occur.

On the contrary, if $\mathbb{V}_n^{\mathbf{I}} \neq \{\phi\}$, then there exists at least one access-unit belonging to the set $\mathbb{V}_n^{\mathbf{I}}$, which if received and decoded, can be displayed without errors, regardless of the reception status of any other access-units carried in the burst. ■

Proposition 2: If f is the play-out frame-rate of ${}^nR_{G'}^{\mathbb{V}^2}$ carried in a time-sliced burst, the position in play-out order, $p^{\mathbf{I}_0}$, of the first Intra access-unit, is upper bounded by,

$$p^{\mathbf{I}_0} \leq \lfloor (\Delta^\Xi - \Delta^\alpha) f \rfloor : p^{\mathbf{I}_0} \in \mathbb{Z}^+. \tag{18}$$

Proof: A burst carrying ${}^nR_{G'}^{\mathbb{V}^2}$ contains $|\mathbb{V}_n|$ access-units. Each access-unit has a play-out duration of $1/f$. Let the position, in play-out order, of the first Intra in the time-sliced burst be $p^{\mathbf{I}_0}$. If the whole burst is received and decoded correctly, then Δ^β is given by,

$$\Delta^\beta = \frac{p^{\mathbf{I}_0}}{f}. \tag{19}$$

Zapping-delay $\Delta^\bullet$ is the sum of Δ^α, and Δ^β. Hence,

$$\Delta^\bullet = \Delta^\alpha + \frac{p^{\mathbf{I}_0}}{f}. \tag{20}$$

From (17), the maximum bound for $\Delta^\bullet$ is Δ^Ξ. Therefore, for any value of $p^{\mathbf{I}_0}$,

$$\Delta^\alpha + \frac{p^{\mathbf{I}_0}}{f} \leq \Delta^\Xi, \tag{21}$$

must be true. Evaluating the above inequality for $p^{\mathbf{I}_0}$,

$$p^{\mathbf{I}_0} \leq \lfloor (\Delta^\Xi - \Delta^\alpha) f \rfloor : p^{\mathbf{I}_0} \in \mathbb{Z}^+,$$

is obtained. ■

Proposition 3: For an uniformly distributed zapping-event time in the interval δ^{1^*}, $\Delta^{\beta} = 0$, only if the first access-unit in play-out order is of Intra type, and received last.

Proof: The proof is divided into two parts. This first part of this proof ignores prediction dependencies; that is all access-units are considered Intra coded. The second part of this proof, takes into account prediction dependencies.

Consider the sub-sequence ${}^{n}R_{G'}^{\mathbb{V}^2}$. It consist of $|\mathbb{V}_n| = k$ access-units. The set of access-units in the sub-sequence is written as $\mathbb{V}_n = \{v_0, v_1, \ldots, v_{k-1}\}$. Further, let elements of the set $\mathbb{V}_n$ be ordered according to their play-out times. That is, they form a sequence

$$\left(t_s^{\mathbf{P}}(v_0) < t_s^{\mathbf{P}}(v_1) < \cdots < t_s^{\mathbf{P}}(v_{k-1})\right).$$

For play-out synchronization delay to be zero, $I^{\mathbf{P}}(v_0) = 1$. From (2), (7), and (10), the minimum condition for $I^{\mathbf{P}}(v_0) = 1$, is that v_0 be received, i.e. $I^{\mathbf{R}}(v_0) = 1$.

Given any zapping event time $t^{\bullet} : t_{2i} \leq t^{\bullet} < t_{2i+1}$, the reception interval $\delta_{v_0}^{\mathbf{R}}$ which maximizes the probability of $I^{\mathbf{P}}(v_0) = 1$ is to be found. In other words, the best transmission time for v_0 is such that

$$t_s^{\mathbf{R}}(v_0) = \underset{t \in [t_{2n}, t_{2n+1})}{\arg\max} \left(\Pr\left(I^{\mathbf{P}}(v_0) = 1 | t^{\bullet}\right)\right). \quad (22)$$

Divide the time-interval δ_n^1 into k intervals, where each interval has one of the k access-units received in the burst. The division of the time-interval, δ_n^1, is as shown below.

$$\delta_n^1 = \underbrace{[t_{2n}, t_{m_0})}_{0} \cup \underbrace{[t_{m_0}, t_{m_1})}_{1} \cup \cdots \cup \underbrace{[t_{2n+1}, t_{m_{k-1}})}_{k-1} \quad (23)$$

The under braces indicate time-slots. As mentioned before, at least one access-unit is assumed to be received in the time-sliced burst. Therefore, the last access-unit of the sub-sequence is received during the interval $[t_{2n+1}, t_{m_{k-1}})$ where, $t_{m_{k-1}} = t_{2n+1} + \tau^{\mathbf{R}}(k-1)$. The problem then is to find which among the above time-slots should be $\delta^{\mathbf{R}}(v_0)$ such that (22) is true.

Let $E_{[t_s,t_e)}^{D} = 1$ be the event that an access-unit received in the time-interval $[t_s, t_e)$ is decodable. The zapping event time, $t^{\bullet} \in \mathbb{R}^{+}$, and is assumed to be uniformly distributed. Since, the distribution of $\Pr(E_{[t_s,t_e)}^{D} = 1)$ changes only at start-times of each time-slot in (23), by assuming that $t^{\bullet}$ can only take values from the set $\{t_{2n}, t_{m_0}, \ldots, t_{2n+1}\}$, makes the distribution discrete. Then, $\Pr(E_{[t_s,t_e)}^{D} = 1 | t^{\bullet})$ for the time-slots in (23) is the sequence,

$$\left(\frac{1}{n}, \frac{2}{n}, \ldots, 1\right).$$

Hence, when access-units are Intra coded, the best time-slot for reception is the last time-slot. Now, it remains to be proved that the probability of decoding any access-units that is predicted is lesser than when it is Intra coded.

For a predicted access-unit, say $v_i \in \mathbb{V}_n$, to be fully decodable, it requires all access-units belonging to its all-path subgraph $\mathbb{V}_n^{\mathcal{A}}(v_i)$, to be decodable. This means that v_i requires $|\mathbb{V}_n^{\mathcal{A}}(v_i)|$ access-units to be received and decoded without errors. This is equivalent to choosing $|\mathbb{V}_n^{\mathcal{A}}(v_i)|$ time-slots from among the n time-slots, and since the choice of choosing time-slots is independent, multiplying their probabilities that they are all decodable. This probability is always smaller than one, which is the probability of the access-unit being decodable, if it were Intra coded, and received last. ■

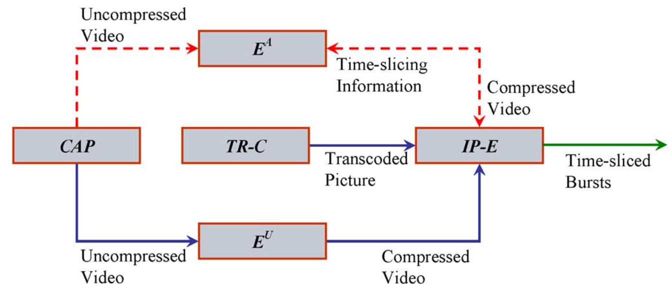

Fig. 9. System implementation that considers both time-slice aware encoding (TAE) and time-slice unaware encoding (TUE). The red arrows indicate the TAE pathway, and the blue arrows indicate the TUE pathway.

VI. System Implementation

Two transmission scenarios are considered when designing the system. In the first, the video encoder is aware of the time-slicing boundaries, because of a duplex communication link between it, and the IP-Encapsulator. The video encoder, with this knowledge, encodes the uncompressed picture in the most optimal way for zapping. This scenario is called "*Time-slice Aware Encoding*" (TAE).

In the second scenario, the video encoder is unaware of the time-slicing boundaries. It therefore, encodes the uncompressed pictures the best way it can, using some optimization criterion. The IP-Encapsulator takes as many access-units from the encoder it needs, to transmit in the next time-slice. This scenario is termed "*Time-slice Unaware Encoding*" (TUE).

The two scenarios are implemented by the system shown in Fig. 9. The red dashed arrows show the data pathway for TAE, while the blue solid lines indicate the data pathway for TUE. The overall system is a combination of the two sub-systems: the TAE, and the TUE. Using this combination, the system presented in Fig. 9 is well placed to guarantee a bounded zapping-delay for all practical encoding, and DVB-H transmission scenarios.

A. Time-Slicing Aware Encoding (TAE)

Consider only those blocks connected by red dashed arrows in Fig. 9. The processing chain starts with the block named CAP. This block is a video sensing, and capturing device. It outputs uncompressed pictures of a video sequence. The uncompressed pictures are then handed over to the video encoder E^A. The encoder E^A has a duplex communication link with the IP-encapsulator, $IP{-}E$. The forward link $(E^A \rightarrow IP{-}E)$ of this duplex connection, delivers the coded access-units, while the backward link $(IP{-}E \rightarrow E^A)$ carries information about time-slicing boundaries. The function of the IP-Encapsulator, $IP{-}E$, is to packetize the incoming coded video data into IP datagrams, compute MPE-FEC, and ready them for transport, as per the protocol standardized in [4]. The output of the $IP{-}E$ is a steady stream of transport stream packet bursts.

The duplex communication between E^A and $IP{-}E$ enables zapping-delay optimization. The $IP{-}E$ knows exactly how many coded access-units can be carried in a burst, and communicates this information to E^A. This information helps E^A design the optimal prediction structure for minimal zapping-delay, while ensuring GFRE after zapping.

B. Time-Slicing Unaware Encoding (TUE)

In some transmission scenarios, the $IP\!-\!E$ might not be able to transmit fixed number of access-units in a burst always. This could either be because of the variable bit rate of coded video, or the variable burst sizes needed for multiplexing with other services. In most cases, it is the combination of both. Therefore, it is difficult to guarantee that the first access-unit, in play-out order, of every burst is an Intra, and always received last (as suggested by Proposition 3). For such scenarios the TUE subsystem is used.

In Fig. 9, the blocks connected by the blue solid arrows, form the TUE sub-system. In TUE, the encoder, E^U, is unaware of time-slicing parameters used by $IP\!-\!E$. In other words, E^U is unaware of how many access-units are to be carried in the next time-sliced burst. The only difference between E^A and E^U is that, E^A is aware of the operations of the $IP\!-\!E$, while E^U is not. However, a parameter that E^U is aware of, is the structural delay, Δ^s.

From Proposition 1, it is known that at least one Intra access-unit is needed for zapping in a burst. However, in TUE, the encoder E^U is not aware of time-slicing boundaries. Furthermore, E^U can encode access-units with a large Intra interval to increases coding efficiency. Then, the only option is to somehow transform "*splicable*" access-units to Intra access-units. In this context, splicable access-units are similar to the one defined in [16]. It is re-defined here to align with the notations used in this paper.

Definition 6: In a coded video sequence $R_{G'}^{\mathbb{V}^2}$, where $\mathbb{V}$ is the set of access-units, and $\mathbb{A}$ is the set of pairs, called the arc set, an access-unit v_k is splicable, only if there exist no pair $(v_t, v_h) \in \mathbb{A}$, such that $t < k$ and $h > k$.

Then, a transformation from splicable access-unit to an Intra access-unit is achieved by either transcoding, or replacing the splicable access-unit, with the corresponding Intra representations.

Among all splicable access-units carried in a burst, only the first splicable picture needs transcoding/replacement with an Intra access-unit. Therefore, it is wasteful (both in terms of system efficiency, and bandwidth consideration) to generate corresponding Intra access-units for all splicable pictures, and transmit it to $IP\!-\!E$. On the contrary, transcoding is a more complex operation than replacing. This paper considers bandwidth to be a more critical resource that computational power. For this reason, a transcoding solution is proposed. The transcoding operation is performed by the block labeled $TR\!-\!C$, and is triggered only once per time-sliced burst.

VII. Simulations and Results

Optimization and establishing relevant bounds are done using experimental simulations. The two cases: TAE and TUE are considered separately. The simulations strictly follows an optimization policy that works within the gamut of current open-source standards and recommendations. First the TAE case is simulated and analyzed, followed by the TUE case.

A. Time-Slicing Aware Encoding (TAE)

The encoder, E^A, is given a maximum structural delay, $\Delta^s > 1$, and the total number of access-units, n, that can be transmitted in the next burst. The unit for counting Δ^s is by the number of pictures in play-out order, and it is constrained such that $\Delta^s \leq n-1$. The structural delay, Δ^s, is chosen to be greater than one because, DVB-H is not a suitable medium for low-delay video dissemination. The burst transmission architecture of DVB-H, always introduces delay, not suitable for low-delay applications.

Finding the optimal prediction structure and transmission order in the TAE case, proceeds in three steps. In the first step, access-units that are to be transmitted in the next burst are assigned ranks. In the second step, prediction arcs are assigned. This is followed by the step where the optimal reception order is deduced. These steps are now described in detail in the following sub-sections.

1) Rank Assignment: The procedure starts with an $n \times n$ empty matrix. Let this matrix be called the Hasse matrix $\mathbf{H}$. The rows of $\mathbf{H}$, top to bottom, denote increasing ranks. The columns, left to right, denote increasing play-out order of access-units in the current burst. Therefore, $\mathbf{H}_{(0,0)}$ is the top-left cell, and $\mathbf{H}_{(n-1,n-1)}$ is the bottom-right cell. Each cell in the Hasse matrix can be assigned one access-unit.

The n access-units form list called the unordered list, $\mathbf{L_U}$. The assignment in $\mathbf{L_U}$ is simply the access-units in their play-out order. Another list $\mathbf{L_O}$ is constructed for the ordered access-units. The initial assignment for this list is empty.

Start assigning every $v_{j\Delta^s} : j \in \mathbb{Z}^+; j\Delta^s \leq n-1$ to $\mathbf{H}_{(j,j\Delta^s)}$. Remove the access-unit $v_{j\Delta^s}$ from $\mathbf{L_U}$, and insert it into the ordered-list $\mathbf{L_O}$. If $\bmod\,(n-1,\Delta^s) \neq 0$, assign the access-unit v_{n-1} to $\mathbf{H}_{(\lfloor (n-1)/\Delta^s \rfloor + 1, n)}$. Again, remove v_{n-1} from $\mathbf{L_U}$, and insert into $\mathbf{L_O}$. The set of access-units

$$\lambda_{\Delta^s}^n = \left\{ v_{j\Delta^s} : j \in \mathbb{Z}^+; j\Delta^s \leq (n-1) \right\} \cup \{v_{n-1}\}, \quad (24)$$

are all now considered ordered. The set $\lambda_{\Delta^s}^n$ is called the *spine* of the prediction structure. The number of access-units in $\lambda_{\Delta^s}^n$, denoted by $|\lambda_{\Delta^s}^n|$, is given by,

$$|\lambda_{\Delta^s}^n| = \begin{cases} \frac{(n-1)}{\Delta^s} + 1, & \text{if } \bmod(n-1,\Delta^s) = 0 \\ \left\lfloor \frac{(n-1)}{\Delta^s} \right\rfloor + 2, & \text{otherwise.} \end{cases}$$

For the remaining access-units in $\lambda_{\Delta^s}^n$, the following procedure is done iteratively, assigning one access-unit to each row r of the Hasse matrix.

Find all candidate access-units that satisfy Definition 5. This is done by considering each access-unit belonging to the list $\mathbf{L_U}$ individually, adding that access-unit in $\mathbf{L}_O$ and evaluating if the inequality in (15) is true. If true, then the access-unit under consideration is a candidate. If false, then it is not. Finally, a candidate list, $\mathbf{L_C}$, is obtained.

After the list $\mathbf{L_C}$ is constructed, the best among the candidate access-units, $v_c \in \mathbf{L_C}$, needs to be found using an optimization measure. The optimization criterion proposed in this paper is called the mean square play-out duration deviation from the optimal play-out duration. The most optimal play-out duration is $1/f$, where f is the frame-rate of $R_{G'}^{\mathbb{V}^2}$. This measure is denoted by $\pi(v_i)$ and given by

$$\pi(v_i) = \left(\frac{1}{f}\right) - \left(\frac{1}{k}\sum_{i=0}^{k-1}\left(\tau^{\mathbf{P}}(v_i)\right)^2\right) : v_i \in \mathbf{L_C} \qquad (25)$$

The best candidate is then,

$$v_c = \underset{v_i \in \mathbf{L_C}}{\arg\min}\left(\pi(v_i)\right) \qquad (26)$$

The best candidate v_c is then assigned to $\mathbf{H}_{(r,c)}$.

The simulation of rank assignment procedure is demonstrated by three example assignments shown in Fig. 10. All the shown prediction structures is assigned with $\Delta^s = 4$. With this structural delay, the access-units $\{v_{j\Delta^s} : j \in \mathbb{Z}^+\}$ are $\{v_0, v_4, v_8, v_{12}, \ldots\}$. Therefore, by using (24), the spine set for $n = 7$, $n = 9$, and $n = 11$ are,

$$\lambda_4^7 = \{v_0, v_4, v_6\}$$
$$\lambda_4^9 = \{v_0, v_4, v_8\}$$
$$\lambda_4^{11} = \{v_0, v_4, v_8, v_{10}\}$$

The remaining access-units, are allocated to the cells of $\mathbf{H}$, by first identifying candidate access-units satisfying Definition 5, then computing the measure given by (25) for all candidate access-units, and then finding the minimum using (26).

2) Arc Assignments: After rank assignment, the matrix $\mathbf{H}$ is filled with access-units to be carried in a burst. Each access-unit has a unique rank in $\mathbf{H}$. The next step is to assign arcs between access-units in $\mathbb{V}$ to form the set of arcs, $\mathbb{A}$. However, when assigning arcs there is one constraint imposed. The arcs are assigned such that,

$$\mathbb{A} = \{(v_t, v_h) : v_t, v_h \in \mathbb{V}; \mathrm{rank}(v_h) > \mathrm{rank}(v_t)\}. \qquad (27)$$

Now the reason for the rank assignments can be elaborated. It is known from [7], [8] that multi-hypothesis prediction improves coding efficiency. By placing the spine the way it has been described before, all access-units not in the spine, i.e. access-units belonging to the set $\mathbb{V} - \lambda_s^n$, will have at least one access-units for backward and forward prediction. Bi-directional prediction is useful for predicting access-units at scene change boundaries, improving coding efficiency.

Although there can be many ways of assigning arcs that adhere to (27), this paper only considers the dyadic structure. The dyadic coding structure is used in the scalable video extension to H.264/AVC [19]. The choice of a dyadic structure is because, it has already been established that the dyadic structure improve coding efficiency in [18].

For the examples shown in Fig. 10, arc assignment using dyadic prediction structures are illustrated in Fig. 11. It can be noticed that there are no arcs that are directed up-wards in the Hasse diagram. This conforms to (27). Comparing Fig. 11(a) and (b), it can also be noticed that, while all temporally scalable coding using the dyadic prediction structure satisfies (25) and (26), all dyadic prediction structure that satisfies these equations, need not be scalable. This is because of the unequal arc lengths produced by (25) and (26), when $n \neq (2^{i+1} + 1) : i \in \mathbb{Z}^+$.

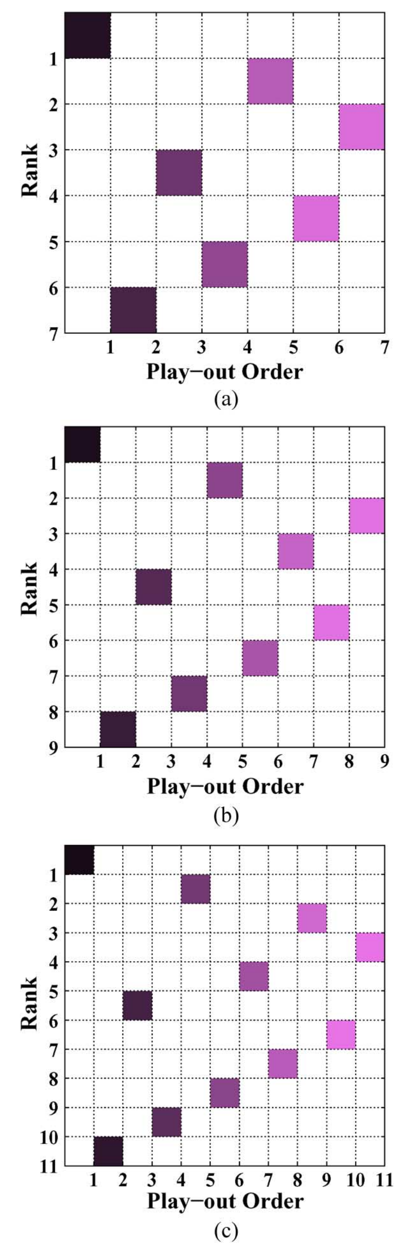

Fig. 10. Rank assignments for three unique (Δ^s, n) pairs: (a) $\Delta^s = 4$ and $n = 7$; (b) $\Delta^s = 4$ and $n = 9$; (c) $\Delta^s = 4$ and $n = 11$.

3) Optimization of Reception Order: The constraint in (27) ensures that there exist no (v_t, v_h) directed-arc pairs such that $\mathrm{rank}(v_h) < \mathrm{rank}(v_t)$. Therefore, for any access-unit $v_k \in \mathbb{V}$, the access-units in its all-path graph, $\mathbb{V}^{\mathcal{A}}(v_k)$, all have a ranks less than or equal to $\mathrm{rank}(v_k)$. Then, for any v_k, using (7), (2), and (10), $I^P(v_k) = 1$ for any zapping-event time $t^\bullet$ with the highest probability, if the reception order of access-units is in the descending order of ranks.

Fig. 12 shows the reception order, (on the left) and play-out duration sequence, τ^P (on the right), for the three examples that have been considered. In Fig. 12(a) and (b), the non-increasing curves indicate GFRE for the different zapping-event time instants. However, in Fig. 12(c), there is a case when $t^\bullet = t9$, where GFRE is not available.

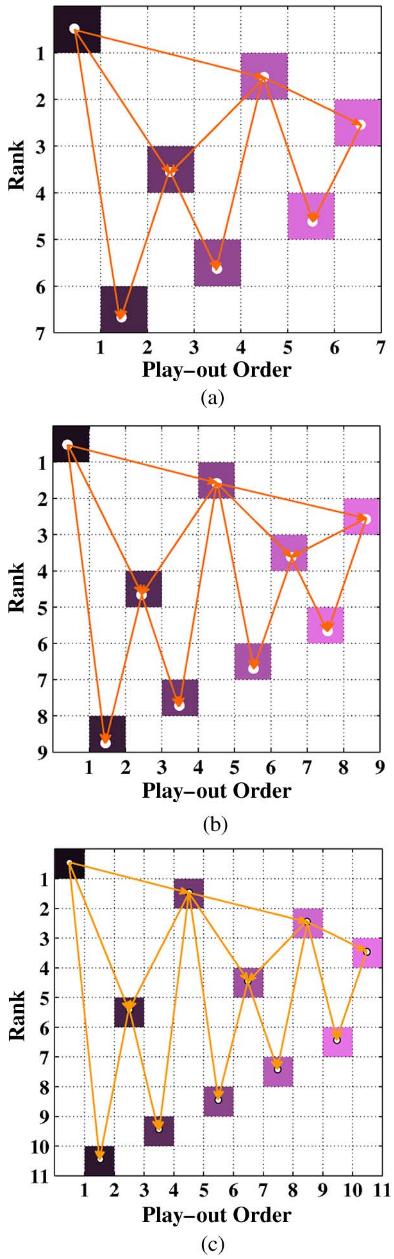

Fig. 11. Arc assignment for the rank assigned cases of (Δ^s, n) pairs, shown in Fig. 10: (a) $\Delta^s = 4$ and $n = 7$; (b) $\Delta^s = 4$ and $n = 9$; (c) $\Delta^s = 4$ and $n = 11$.

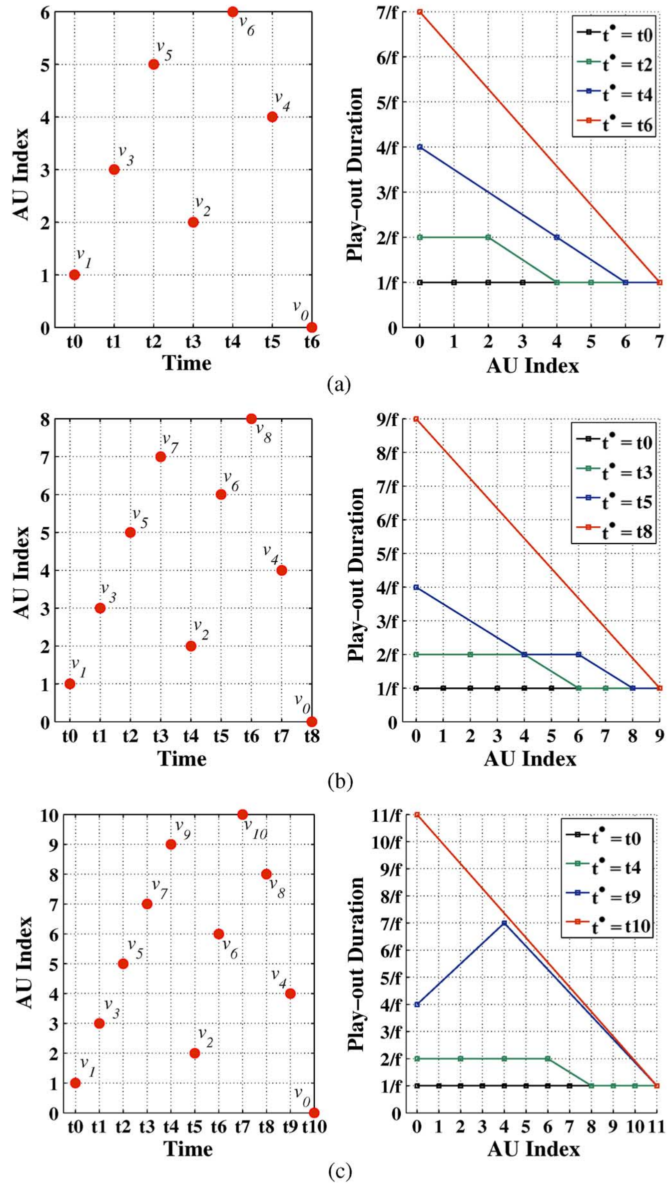

Fig. 12. (Left) Reception order and (right) play-out duration of access-units that are assigned ranks and arcs shown in Figs. 10 and 11 respectively: (a) $\Delta^s = 4$ and $n = 7$; (b) $\Delta^s = 4$ and $n = 9$; (c) $\Delta^s = 4$ and $n = 11$.

Consider the case when $s = 4$ and $n = 11$. After allocating ranks and arcs for access-units, the optimal reception order is

$$(v_0, v_3, v_5, v_7, v_9, v_2, v_6, v_{10}, v_8, v_4, v_0)$$

The spine set is, $\lambda_4^{11} = \{v_{10}, v_8, v_4, v_0\}$. The access-units in the spine are received last. For the case when a zapping-even occurs at $t9$, the only access-units that can be decoded and presented are v_4, v_0. This causes the loss of GFRE.

There are two ways to tackle this problem. In the first, the solution lies in presenting only v_0 when any access unit of the spine is not received. This means that, there could be some access-units of the spine that are received and decode perfectly, but are not displayed. This could be considered as a waste of computation resources. The second solution is to constraint the number of access-units in the spine of a burst to a maximum of three. This constraint in-turn fixes the maximum number of access-units in a burst. The optimal number of access-units n, with a structural delay Δ^s to be carried in a time-sliced burst should follow $n \leq 2\Delta^s$ if GFRE is to be achieved for any zapping-event time $t^\bullet$.

B. Time-Slicing Unaware Encoding (TUE)

The TAE simulations and analysis considered the dyadic prediction structure for arc assignment. The justification for doing so was also given. It was also found that a temporally scalable video, with a dyadic prediction structure, is a special case of the prediction structure satisfying the minimization condition given by (22). Hence, a temporally scalable video, using dyadic prediction structure, is also considered for the TUE case.

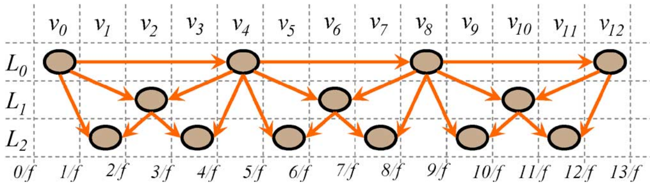

Fig. 13. Temporal scalable coding, using dyadic prediction, with a structural delay, $s = 4$.

Let the TUE encoder, E^U, code a video sequence using a dyadic prediction structure. The base-layer access-units, denoted by the set λ_{Δ^s}, where Δ^s is the structural delay, then forms the spine of the prediction structure. The enhancement layer access-units are elements of the set $\mathbb{V} - \lambda_{\Delta^s}$.

Fig. 13 shows a temporal scalable video sequences that uses dyadic prediction structure. The structural delay s is set to four. The access-units constituting the spine of this sequences are $\lambda_4 = \{v_0, v_4, v_8, v_{12}\}$. These access-units belong to the base layer of the temporal scalable video. The remaining set of access-units, $\{v_1, v_2, v_3, v_5, v_6, v_7, v_9, v_{10}, v_{11}\}$, constitute the enhancement layer.

A pre-requisite to minimize zapping delay, established from Proposition 1, is that at least one Intra access-unit is available in a burst. The E^U cannot always guarantee this because, unlike the E^A, it is not aware of the time-slicing boundaries generated at the $IP{-}E$. In fact, optimally, the E^U encodes access-units with a large Intra interval, which increases coding efficiency. Then, the only option is to transcode splicable access-units to Intra access-units. Splicable access-unit form the base-layer when dyadic prediction structure is used. The following corollary from Proposition 1 can be deduced.

Corollary 1 (To Proposition 1): Every time-sliced burst should carry at least one base-layer access-unit for a successful zapping-event.

The operation of a transcoder is then a base-layer decoder combined with an Intra encoder, where the Intra encoding is triggered only once per time-slice. The base-layer decoding with an occasional Intra encoding increase computational complexity modestly, and faster than real-time implementations are practically available for both these procedures.

Equation (18) in Proposition 2, gives the bound for the position of the first Intra access-unit in a burst. Extending this to the case where splicable access-unit can be replaced with Intra access-unit, in the worst case, the start of a time-slice is just after a splicable picture in play-out order. Then, the next splicable access-unit is displayed after $\Delta^s - 1$ access-units in play-out order. Substituting $p^{\mathbf{I}_0} = \Delta^s - 1$, the time-slice synchronization delay Δ^α is bounded by

$$\Delta^\alpha \leq \Delta^\Xi - \frac{\Delta^s - 1}{f}. \tag{28}$$

The time-slice synchronization delay, Δ^α, must be always greater than zero. Hence,

$$\Delta^s < \Delta^\Xi f + 1. \tag{29}$$

Simulation to evaluate parameters of zapping-delay was conducted for 10000 time-sliced bursts. The number of access-units, called the AU-count, carried in each burst, was assumed to be drawn from a log-normal distribution. It must be stressed that the results to be established are distribution independent. Therefore, the results established here is true for any distribution. The log-normal distribution was chosen only for illustration sake. Three mean-variance pairs, (μ, σ), for the log-normal distribution were again arbitrarily selected. The simulated pairs were, {(10,20),(20,20),(30,20)}. Fig. 14 shows the generated AU-count pattern for the pair (10,20). The probability mass function of the generated pattern is also shown in the figure.

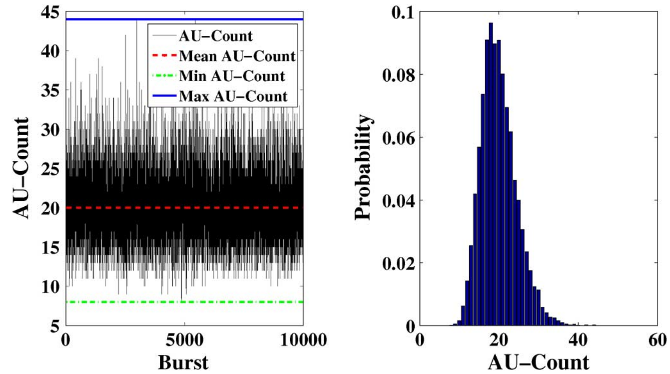

Fig. 14. (Left) AU-count per burst {Mean 20, Max 44, Min 8}. (Right) Log-normal probability mass function, $\{\mu, \sigma\} = \{20, 20\}$, used to generate the AU-count.

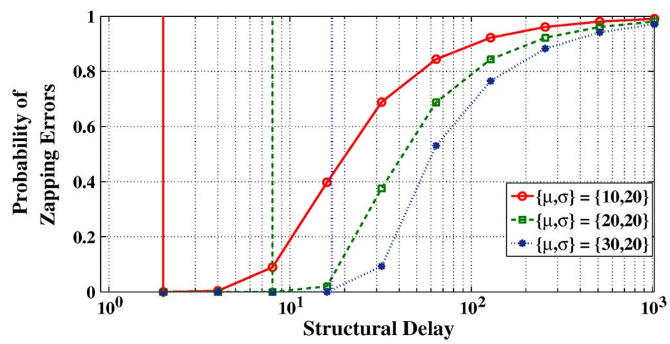

Fig. 15. Probability of zapping error for ten unique values of structural delays and the three generated AU-count sequence shown in Fig. 14.

Fig. 15 plots the probability of an unsuccessful zapping-events considering ten different values of structural delay Δ^s, for each AU-count pattern generated. These values were {2,4,8,16,32,64,128,256,512,1024}. Zapping-event times were assumed to be uniformly distributed. The figure also indicates the minimum number of access-units per burst. The plots indicate that for a zapping-event with a guaranteed zero failure probability, the structural delay Δ^s must be less than the minimum AU-count among all time-sliced bursts. This is an experimental proof of the fact that at least one base-layer access-unit should be carried in a burst for a successful zapping.

The maximum play-out synchronization delay, Δ^β, for the ten different structural delays simulated are plotted in Fig. 16. An example threshold value of 0.5 for Δ^Ξ is also plotted. From the figure, it is noticed that, for Δ^β to be less than Δ^Ξ, the value of Δ^s follows the inequality in (29). Secondly, given some value of Δ^s such that $\Delta^\Xi > \Delta^\beta$, the remaining time duration,

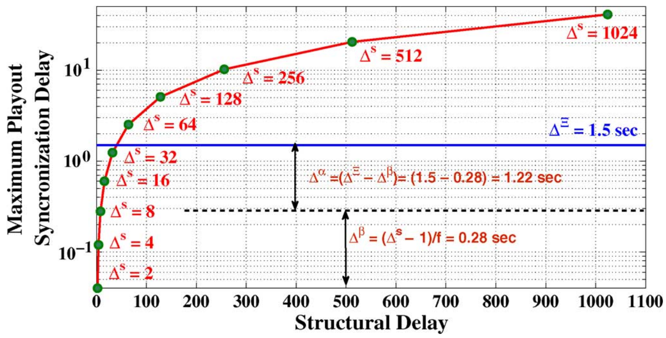

Fig. 16. Maximum play-out synchronization delay plotted against the ten unique structural delay simulated.

$\Delta^\Xi - \Delta^\beta$, gives the maximum time for the transmission synchronization delay Δ^α. The value of Δ^α has an impact on the bandwidth usage in DVB-H.

VIII. Conclusion

Zapping-delay is a contributing factor that negatively effects Quality-of-Experience (QoE). When broadcasting video over DVB-H, the prediction structure of coded video, and the time-sliced transmission architecture, contribute towards zapping-delay. This paper started with a theoretical analyses of video prediction structures. A coded video sequence was treated as a binary relation between access-units. The transitive reduction of the graph generated by the binary relation was analyzed using Hasse diagrams. Reachability concepts of graph theory was then used to establish conditions for reception, decoding and play-out of access-units after a zapping-event. The importance of zapping followed by a gradual improvement in quality was stated. For this cause, Gradual frame-rate enhancement (GFRE) was defined. The need for both prediction structure optimization and reception order modification, for zapping with gradual QoE enhancement, were illustrated.

Optimization strategies considered two scenarios. In the first, called the time-slice aware encoding (TUE), the video encoder was aware of time-slicing boundaries. In the TAE scenario, the encoder was fully aware of how many access-units were to be carried in a time-slice burst, and it could optimize its prediction structure and deduce the most optimal reception order of access-units. Algorithm for construction of the optimal prediction structure using a dyadic prediction architecture was given. It was proved that the most optimal prediction structure contained as its first access-unit in presentation order and Intra coded picture. It was also proved that this Intra access unit needed to be transmitted last in the burst for the best chance of zapping with minimal delay. Furthermore, it was also found that the ideal reception order for GFRE is when access-units were received in their decreasing order of their ranks.

In the second scenario, the encoder did not have any knowledge of where the time-slicing boundaries occurred. This scenario was called time-slice unaware encoding (TUE). In the TUE scenario, where information about access-units are unknown, a temporal scalable video using a dyadic prediction structure was considered. A transcoding solution is proposed, with the assumption that network bandwidth is a more critical resource than transmitter side computing power. The bounds for the maximum structural delay, given the maximum bound for zapping delay was established.

By using the system, methods, and constraints presented in this paper, a zapping delay below some maximum threshold can be guaranteed. Quality of the video after zapping, is also gradually enhanced. Furthermore, the coding method has not deviated from the previously researched dyadic architecture shown to provide good coding efficiency. Therefore, the coding efficiency along with bounded zapping-delay and gradual frame enhancement improves QoE.

However, the proposal in this paper only partially solves the problem of bounded zapping-delay. This paper has assumed that the zapping-event occurs during the transmission of a burst of the desired program. The case of bounded zapping-delay when a zapping-event occurs in between two bursts carrying the desired program is the future for this research.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments. Thanks go to Dr. M. Hannuksela for his involvement in this work. Thanks also go to Mr. L. Kondrad, and Dr. R. Bregovic for their support during the writing of this paper.

References

[1] J. M. Boyce and A. M. Tourapis, "Fast efficient channel change," in *Proc. ICCE Consum. Electron. Digest Tech. Papers. Int. Conf.*, 2005, pp. 1–2.

[2] R. Diestel, *Graph Theory*, ser. Graduate Texts in Mathematics, 3rd ed. New York: Springer-Verlag, 2005, vol. 173.

[3] *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television-EN 300 744*, DVB Std. V1.5.1, ETSI, Jun. 2004 [Online]. Available: http://www.etsi.org, European Standard (Telecommunications Series)

[4] *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)-EN 302 304*, DVB Std. V1.1.1, ETSI, Nov. 2004 [Online]. Available: http://www.etsi.org, European Standard (Telecommunications Series)

[5] *Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols-TS102 005, Technical Specifications*, DVB Std. V1.3.1, ETSI, Jul. 2007 [Online]. Available: http://www.etsi.org, Technical Specifications

[6] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital broadcast services to handheld devices," *Proc. IEEE*, vol. 94, no. 1, pp. 194–209, 2006.

[7] B. Girod, "Efficiency analysis of multihypothesis motion—Compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, 2000.

[8] B. Girod, "Why B—Pictures work: A theory of multi-hypothesis motion-compensated prediction," in *Proc. Int. Conf. Image Process. ICIP*, 1998, vol. 2, pp. 213–217.

[9] *Information Technology Coding of Audio-Visual Objects Part 10: Advanced Video Coding*, ISO/IEC Std. Fifth, ISO/IEC, May 2009.

[10] R. Kooij, K. Ahmed, and K. Brunnstrom, "Perceived quality of channel zapping," in *Fifth IAESTED Int. Conf. Commun. Syst. Netw. (CSN)*, Aug. 28–30, 2006, pp. 155–158.

[11] C. Y. Lee, C. K. Hong, and K. Y. Lee, "Reducing channel zapping time in IPTV based on user's channel selection behaviors," *IEEE Trans. Broadcast.*, vol. 56, no. 3, pp. 321–330, Sep. 2010.

[12] NorDig, "Unified IRD specification," Tech. Rep., Jul. 1, 2010.

[13] V. Ollikainen and C. Peng, "A handover approach to DVB-H services," in *Proc. IEEE Int. Multimedia Expo Conf.*, 2006, pp. 629–632.

[14] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Video encoding and splicing for tune-in time reduction in IP datacasting (IPDC) Over DVB_H," in *Proc. IEEE Int. Multimedia Expo Conf.*, 2006, pp. 601–604.

[15] M. Rezaei, M. M. Hannuksela, V. K. M. Vadakital, and M. Gabbouj, "Spliced video and buffering considerations for tune-in time minimization in DVB_H for mobile TV," in *Proc. IEEE 17th Int. Personal, Indoor Mobile Radio Commun. Symp.*, 2006, pp. 1–5.

[16] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Tune-in time reduction in video streaming over DVB-H," *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 320–328, Mar. 2007.

[17] B. Schroeder, *Ordered Sets: An Introduction*, 1st ed. Cambridge, MA: Birkhauser Boston, 2002.

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF," in *Proc. IEEE Int Multimedia and Expo Conf.*, 2006, pp. 1929–1932.

[19] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.

[20] P. Siebert, T. N. M. Van Caenegem, and M. Wagner, "Analysis and improvements of zapping times in IPTV systems," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 407–418, 2009.

Vinod Kumar Malamal Vadakital received the B.Tech. degree in computer science and engineering from Bangalore University, Bangalore, India, and the M.S. degree in information technology from Tampere University of Technology, Tampere, Finland, in 1998 and 2005, respectively.

From 1999 to 2001, he worked as a Project Assistant at the Indian Institute of Science, Bangalore, India. From 2001 to 2003 he was a Research Engineer at Fraunhofer Institute of Integrated Circuits (IIS-B), Erlangen, Germany. From 2003 to 2005, he worked as a Research Assistant at Tampere University of Technology. Presently, he is a Researcher at the Tampere University of Technology and he is working towards the doctoral degree. His research interests are in the areas of video coding algorithms, video quality analysis, and mobile multimedia communications.

Moncef Gabbouj (S'86–M'91–SM'95–F'11) received the B.S. degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, Indiana, in 1986 and 1989, respectively.

He is an Academy Professor with the Academy of Finland since January 2011. He was Professor at the Department of Signal Processing at Tampere University of Technology, Tampere, Finland. He served as Head of the Department during 2002–2007. Dr. Gabbouj was a visiting professor at the American University of Sharjah, UAE, in 2007–2008 and Senior Research Fellow of the Academy of Finland in 1997–1998 and 2007–2008. His research interests include multimedia content-based analysis, indexing and retrieval, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding.Dr. Gabbouj served as Distinguished Lecturer for the IEEE Circuits and Systems Society in 2004–2005. He served as Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and was guest editor of Multimedia Tools and Applications, the European journal Applied Signal Processing. He is the past chairman of the IEEE Finland Section, the IEEE CAS Society, Technical Committee on DSP, and the IEEE SP/CAS Finland Chapter.

Dr. Gabbouj was the recipient of the 2005 Nokia Foundation Recognition Award and co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award from the 1994 Nordic Signal Processing Symposium. He is a Fellow of IEEE and member of IEEE SP and CAS societies.