Optimal Frame Transmission for Scalable Video with Hierarchical Prediction Structure

Saied Mehdian and Ben Liang*

Abstract—An optimal frame transmission scheme is presented for streaming scalable video over a link with limited capacity. The objective is to select a transmission sequence of frames and their transmission schedule such that the overall video quality is maximized. The problem is solved for two general classes of hierarchical prediction structures, which include as a special case the popular dyadic structure. Based on a new characterization of the interdependence among frames in terms of trees, structural properties of an optimal transmission schedule are derived. These properties lead to the development of a jointly optimal frame selection and scheduling algorithm, which has computational complexity that is quadratic in the number of frames. Simulation results show that the optimal scheme substantially outperforms two existing alternatives.

I. INTRODUCTION

Video streaming is contributing to an ever increasing portion of the Internet traffic going through service providers. Beside stringent demands on bandwidth and delay, it requires adaptation to heterogeneous access networks and devices, in order to achieve satisfactory viewing experience. The efficient and adaptive transmission of video is paramount to service providers and users alike.

Toward this goal, one thrust of engineering effort is on efficient video coding. The H.264 Advanced Video Coding (AVC) standard, including its Scalable Video Coding (SVC) enhancements [1][2], is the most widely adopted coding scheme, used in a wide array of applications including Blu-ray and Youtube. It provides a means to encode a video stream into sub-streams of different quality, which may be selectively transmitted and decoded based on the available communication hardware and bandwidth. However, the H.264 standard does not specify how to select or schedule video data for transmission.

The other engineering thrust is on adaptive transmission schemes for scalable video streaming. It concerns the joint optimization of two procedures, first, the selection of a subset of the video frames for transmission, and second, scheduling the transmission of those frames before their display deadlines expire. This is challenging mainly due to the Hierarchical Prediction Structure (HPS) of the video codec [2]. Prediction is necessary to improve coding efficiency, but it creates complicated dependencies between frames in a video sequence. Most existing studies on efficient transmission of scalable video [3][10] address dependence by a simplified flow-based model, in which a video layer can be decoded only if all lower layers are already decoded. In the more realistic frame-based approach, dynamic frame dropping and retransmission have been studied in [11]-[17], but these proposals are mostly based on heuristics that are challenging to analyze mathematically.

In this work, we present an analytical approach to optimize the frame transmission schedule of videos with hierarchical prediction, with joint consideration for the optimal frame selection given a limited link capacity. The optimization objective is to minimize the loss of playback quality in the transmitted video sequence. To the best of our knowledge, this is the first work to provide a provably optimal solution to frame-by-frame lossy video transmission which accounts for hierarchical prediction.

Our main contributions are as follows. First, we formally characterize the inherent properties of an optimal frame transmission schedule in terms of the dependency structure arising from the HPS. This leads to the categorization of two general classes of videos, termed Sequential Isomorphically Ordered and Quasi Sequential Isomorphically Ordered, which include as special cases some commonly employed prediction schemes such as the hierarchical dyadic structure [2]. Second, we develop efficient algorithms to compute the frame ordering rules in an optimal transmission schedule. We show that they hold regardless of the subset of frames selected for transmission and, in particular, they govern a unique universal transmission sequence of which an optimal schedule is a subsequence. Third, based on the above structural observations, we propose dynamic programming solutions for jointly optimal frame selection and scheduling, which is shown to have only quadratic complexity. Simulation with common test video traces show that the proposed method can substantially improve the quality of lossy video streaming over a link with limited capacity.

The rest of this paper is organized as follows. In Section II the related works are presented. In Section III we explain the system model and problem statement. In Section IV we discuss the special properties of an optimal schedule. In Sections V and VI optimal algorithms for scheduling in the two general classes of HPS are presented. In Section VII simulation results with video traces are shown to demonstrate the performance gain by the proposed optimal solution. Finally, Section VIII concludes the paper.

II. RELATED WORK

Previous works related to this paper can be categorized into two groups: classical scheduling and scalable video streaming.
A. Classical Deterministic Scheduling

Classical scheduling problems related to our work are addressed under the theory of deterministic scheduling with delay constrained jobs [18]-[20]. The schedule may need to satisfy some precedence relation between jobs. However, if a job is discarded, it has no effect on the other jobs [21]. This is different from the dependency relation between video frames. To the best of our knowledge, there is no known general theory on scheduling with dependency.

B. Scheduling for Scalable Video Streaming

Many prior works do not consider the dependency relation in video transmission [22]-[30]. In all theses works the video content is transmitted in an earliest-deadline-first (EDF) fashion. In [24], a dropping scheme is proposed which does not take into account deadline constraints. In particular, for each regular time interval, a constant size set of consecutive frames is considered for transmission and some of them are dropped to comply with channel rate limit, regardless of deadline constraints. In [22], [23], a dropping scheme utilizing dynamic programming is presented which dynamically adjusts the transmission power and therefore the rate to ensure selected packets meet their deadlines. Studies that do consider the dependency relation can be divided into two main groups: flow based and frame based. In the flow-based approach [3]-[10], the video is modelled as a set of inter-dependent data flows, each providing basic or enhanced playback quality. Flow-based dependency structure often is relevant only within the same display frame. In this work, we consider the more complicated frame-based dependency structures.

In the frame-based approach, the video sequence is modelled as a set of data units, roughly corresponding to the display frames. Each data unit may depend on one or more data units for decoding. In [11]-[14], heuristics are proposed to drop data units under bad link conditions according to some predefined priority. No analytical result is presented on how to set priority levels, and evaluation is performed through simulation only. In comparison, our proposed scheduling algorithm is provably optimal.

More sophisticated online frame-based retransmission schemes have been studied for the lossy environment in [15]-[17]. In [15], a heuristic retransmission windowing approach is used to improve the rate-distortion performance of video delivery over an erasure channel. In addition, [15] intends to minimize a cost function including as a linear term the amount of transmitted data during an entire video streaming session, whereas in this thesis, a channel rate limit is considered which not only restricts the amount of transmission data during the entire session but also at each time instant, which is a much more realistic contraint. In [16], a content-aware hybrid ARQ method for video streaming over wireless links is proposed and analyzed using Markov chains, where frames are transmitted in EDF fashion and retransmission are used for I-frames. In [17], an approximate analytical model for multi-user video transmission over wireless links is developed, for videos that do not use B-frames. These works target more ambitious problems than ours, obtaining weaker suboptimal solutions. In this work, we target a simpler offline decision problem with an optimal scheduling algorithm.

III. Hierarchical Prediction Structure and Problem Statement

We consider a time-slotted point-to-point communication scenario where the transmitter sends a pre-recorded video sequence to the receiver through a link with fixed capacity C bits/timeslot. We assume that a lower-layer protocol ensures the correct reception of any data transmitted at or below the link capacity. Without loss of generality, we omit the propagation delay on the link, since otherwise we only need to shift the display time of all frames by a constant offset to accommodate it.

A. Hierarchical Prediction Structure

The video consists of a sequence of frames indexed in their display order by 1, 2, . . . , N. The frames are classified into three groups: I-frames, P-frames, and B-frames. I-frames are intra-coded and do not depend on other frames, P-frames are inter-coded based on a preceding frame in the display order, and B-frames are inter-coded based on a preceding frame and a successive frame. The set of P-frames and B-frames between two consecutive I-frames, plus the leading I-frame, is called a group of pictures (GOP). The P-frames and B-frames of different GOPs are isolated from each other, but those within a GOP adhere to a specific dependency structure governed by the adopted prediction coding. This corresponds to the general HPS of H.264 AVC, which enables temporal scalability [2]. In this work, we focus on common videos with a single spatial layer. The problem of jointly optimizing multiple spatial layers is left for future research.

As an example, the most widely adopted HPS is the hierarchical dyadic structure. Following the notations of [31], a hierarchical dyadic structure is denoted by Gnm where n is the size of each GOP and m is the number of B-frames between consecutive I-frames or P-frames, with \( m = 2^m - 1 \) for some \( w \in \mathbb{N} \). Figure 1 illustrates a GOP with G16B3.

Fig. 1: Frames \((0,1,\ldots,15)\) form a GOP with hierarchical dyadic structure G16B3.
this paper, we use $G$ to denote the DAG corresponding to the video sequence under consideration, and the terms “frame” and “node” are used interchangeably depending on context.

We denote the sequence of GOPs as $\{GOP_i\}$ in their display order. We also denote the sequence of I-frames as $\{I_i\}$ in their display order, so that $I_i$ corresponds to GOP$_i$. The descendants of $I_{i+1}$ can only be in GOP$_i$ and GOP$_{i+1}$, and except $I_{i+1}$, the frames in GOP$_i$ cannot depend on any frame in other GOPs. Let $\mathcal{N}_i$ be the set of non-I-frames in GOP$_i$, and $\mathcal{D}_i$ be the set of frames in GOP$_i$ that are descendants of both $I_i$ and $I_{i+1}$.

**B. Quality Optimization under Limited Link Capacity**

Each video frame $l$ is associated with three parameters $(S_l, d_l, q_l)$, where $S_l$ is its size in bits, $d_l$ is its display deadline offset against some given transmission start time (i.e., time 0), and $q_l$ is its quality increment, the expected lost of video quality if the frame is not displayed [15]. For example, one possible measure of $q_l$ is the peak signal-to-noise ratio. Denote the deadline of the last frame as $T$.

A transmission schedule is a vector containing the transmission starting times of a sequence of video frames, sorted in ascending order. It indicates both the selected frames for transmission and the order of transmission. In a transmission schedule, a frame is decodable if and only if its parents are already decoded. A frame is successful if it becomes decodable and arrives at the receiver prior to its display deadline.

The reward function of a transmission schedule, $S$, is defined as the sum of the quality increment of its successful frames, i.e.,

$$ z(S) = \sum_{l \in S, l \text{ is successful}} q_l \quad (1) $$

Our objective is to find a transmission schedule that maximizes the reward given a link capacity limit $C$. This problem is called the Scheduling Problem throughout the rest of this paper. The resultant transmission schedule is called an optimal schedule, and the resultant reward function is indicated with $z^*$.

**Transmission of Unsuccessful Frames.** An optimal schedule may permit unsuccessful frames to be transmitted. Although these frames are not displayed, their importance arises from the fact that due to the dependency structure, they can help decode other frames. However, the transmission of an unsuccessful frame which has no successful descendant is useless. Therefore, in order to obtain an optimal schedule, we need to focus only on transmission schedules in which each unsuccessful frames have at least a successful descendant.

**Sufficiency of One-by-One Frame Transmission.** The following theorem indicates that it suffices to consider only one-by-one frame transmission in the search for an optimal schedule.

**Theorem 1:** Every schedule can be transformed into a schedule with only one-by-one frame transmission, with at least the same successful frames and therefore, with at least the same reward function.

**Proof:** See Appendix [1]

In this case, the number of timeslots required to transmit frame $l$ is given by $\Delta t_l = \frac{S_l}{C}$. We assume the timeslot size is small enough such that $\Delta t_l$ are well approximated by integers.

**Optimal Transmission Sequence.** With one-by-one frame transmission, there is no benefit in leaving a gap between the transmission of two consecutive frames, in terms of the reward function. Therefore, in computing an optimal schedule, it suffices to consider only the transmission sequence, i.e., the set of frames selected for transmission and their order of transmission. With an optimal transmission sequence, an optimal schedule can be determined by simply transmitting the frames back-to-back without any gap between them. Hence, throughout the rest of this paper, a “transmission schedule” refers to only its “transmission sequence” without the timing information.

The action space of this problem includes all permutations of all subsets of frames in the video sequence. Therefore, the complexity of exhaustive search would be prohibitive. In the next three sections, we first present some inherent properties of an optimal transmission schedule, which will then be used in Sections V and VI to develop polynomial solutions to the scheduling problem.

**IV. Properties of an Optimal Schedule**

In this section, we first describe a transformation of the dependency DAG of a video into rooted trees that indicate decodability. Based on this representation, we then present two general classes of HPS that are of interest, and give some important properties of optimal scheduling common to both classes and essential to the solutions presented in Sections V and VI.

**A. Modified Breadth First Search (MBFS) Trees**

We adopt a version of the Breadth First Search (BFS) algorithm [32] on the video DAG $G$, which we call Modified Breadth First Search (MBFS). It takes $G$ and a node $s$ as input and outputs an MBFS tree rooted at $s$. The main difference with BFS is the following: at each node, instead of picking all unvisited children of that node, only decodable unvisited children are picked, where a node is decodable if and only if all of its ancestors have been visited. Moreover, in constructing the MBFS tree, the decodable unvisited children are sorted in ascending order of their deadlines. Such ordering is important to the concept of isomorphically ordered trees presented later. The MBFS algorithm is formally given in Algorithm [1]

We run MBFS on $G$ and each I-frame in the display order, creating an MBFS forest, whose components are MBFS trees rooted at the I-frames, each tree corresponding to a GOP. The complexity of this procedure is $O(N)$, since each MBFS tree creation has constant complexity given a fixed GOP size. Figure 2 illustrates the result of MBFS on the DAG in Figure 1 with node 16 removed. The main benefit of MBFS is to represent the dependency structure of frames in the format of trees.

We emphasize that in terms of the MBFS forest, the children of each node, in standard graph theoretic terminology for trees, is a subset of the children of that node in terms of the DAG $G$, as defined in Section [1]. This is because being a child in the MBFS forest carries the additional requirement of being decodable. Furthermore, the set of descendants of an I-frame
Algorithm 1 Modified Breadth first Search (MBFS) (G, s)

1: for each node \( u \in V[G] - \{ s \} \) do
2: \( \text{color}[u] \leftarrow \text{WHITE} \)
3: \( d[u] \leftarrow \infty \)
4: \( \pi[u] \leftarrow \text{NIL} \)
5: end for
6: \( \text{colors}[s] \leftarrow \text{GRAY} \)
7: \( d[s] \leftarrow 0 \)
8: \( Q \leftarrow \emptyset \)
9: \( \text{ENQUEUE}(Q, s) \)
10: while \( Q \neq \emptyset \) do
11: \( u \leftarrow \text{DEQUEUE}(Q) \)
12: \( \text{Adj}[u] \leftarrow \text{sequence of children of node } u \)
13: sort \( \text{Adj}[u] \) in ascending order of deadlines.
14: for each \( v \in \text{Adj}[u] \) (in ascending order) do
15: if \((\text{color}[v] = \text{WHITE}) \) and \((\forall i \rightarrow v \Rightarrow \text{color}[i] = \text{GRAY})\) then
16: \( \text{color}[v] \leftarrow \text{GRAY} \)
17: \( d[v] \leftarrow d[u] + 1 \)
18: \( \pi[v] \leftarrow u \)
19: \( \text{ENQUEUE}(Q, v) \)
20: end if
21: end for
22: end while
23: return

Fig. 2: MBFS on DAG in Figure 1 with node 16 removed.

In terms of the DAG can be partitioned into two subsets in \( GOP_{i-1} \) and \( GOP_i \), where only the latter subset is the descendants of \( I_i \) in terms of the MBFS forest. In the rest of this paper, when we refer to the parent-child and ancestor-descendant relations between frames, they are in terms of the DAG by default. When the relations are in terms of the MBFS forest, such exceptions will be clearly stated unless they are obvious from the context.

However, note that the set of descendants of a non-I-frame remains the same in terms of either the MBFS forest or the DAG. Hence, if we focus our attention within a single GOP, then any frame has the same set of descendants in terms of either the MBFS forest or the DAG. In that case we only need to distinguish the reference terms of children but not those of descendants.

B. SIO and Quasi-SIO Classes

Based on the MBFS forest, we next give two important definitions that characterize the HPSs of interest.

Definition 1: The DAG representing the HPS of a video is called sequential if the ancestors of each node lie only on the path between the node and the root of the MBFS tree that contains the node.

Thus, if a DAG is sequential, then decoding a node only requires the availability of nodes residing on the path between the node and the root of the MBFS tree that contains the node. Clearly, if a video sequence consists of only I-frames and P-frames, then its DAG is sequential, since each node only has a single parent. The situation is more complicated when there are B-frames. From Figure 2 it is easy to see that the DAG in Figure 1 is sequential if and only if edges 16 \( \rightarrow \) 14 and 16 \( \rightarrow \) 15 are removed.

In the MBFS forest, let \( T(x) \) denote the subtree rooted at node \( x \), and \( \text{min}(T(x)) \) and \( \text{max}(T(x)) \) be the minimum and maximum display deadlines among the nodes in \( T(x) \), respectively. Let \( c_i(x) \) be the \( i \)th child of \( x \).

Definition 2: An isomorphically ordered tree is a rooted tree such that, if node \( x \) has \( k \) children, they can be re-ordered (and re-indexed) such that

\[
\text{max}(T(c_i(x))) \leq \text{min}(T(c_{i+1}(x))), \quad 1 \leq i \leq k-1. \tag{2}
\]

The above definition resembles that of B-trees [32], but there is no requirement for the tree being balanced. An example of isomorphically ordered trees are binary search trees. A more general example is shown in Figure 2. We confine our discussions to HPSs whose MBFS forest contain only isomorphically ordered trees. The following lemma will be used in deriving the optimal schedules.

Lemma 1: In an MBFS forest with isomorphically ordered trees, consider two irrelevant nodes \( x \) and \( y \). If \( d_x < d_y \), then

\[
\text{max}(T(x)) < \text{min}(T(y)).
\]

Proof: See Appendix III

From the two definitions above, we say that an HPS is Sequential and Isomorphically Ordered (SIO) if its DAG is sequential and its MBFS forest contains only isomorphically ordered trees. For example, the HPS shown in Figure 1 is SIO if edges 16 \( \rightarrow \) 14 and 16 \( \rightarrow \) 15 are removed. Furthermore, to extend the concept of SIO to more general HPSs that contain B-frames that depend on an I-frame from the next GOP, we say that an HPS is Quasi Sequential and Isomorphically Ordered (quasi-SIO) if it would become SIO after such dependency was removed. For example, the HPS shown in Figure 1 is quasi-SIO.

In this work, we focus only on the SIO and quasi-SIO classes of HPSs, since many common HPSs in practice belong to either class. For example, the zero-delay structure [2] is SIO and the classical prediction structure [31] used as default in H.264/AVC is quasi-SIO. For a proof that the widely adopted hierarchical dyadic structure is quasi-SIO, see Appendix III.

C. Canonical Form of Optimal Transmission Sequence

Theorem 2 below suggests that it suffices to find an optimal transmission sequence that satisfies some special properties. In particular, for the SIO and quasi-SIO classes, we say that a transmission sequence that preserves the two properties in Theorem 2 is in the canonical form.
**Theorem 2:** In the SIO and quasi-SIO classes, any transmission sequence \( S \) can be re-ordered into a transmission sequence \( S' \) with at least the same set of successful frames (i.e., \( z(S) \leq z(S') \)) and with the following properties:

1) The ancestors of a frame are scheduled prior to that frame.

2) Consider two irrelevant frames \( l_i \) and \( l_j \) with \( d_{l_i} < d_{l_j} \). If they are neighbors in the transmission sequence, then \( l_i \) is scheduled before \( l_j \). If \( l_i \) is scheduled after \( l_j \), then not all of the frames scheduled between \( l_j \) and \( l_i \) are irrelevant with respect to \( l_j \).

**Proof:** See Appendix [V]

**Theorem 2** will be used extensively in the rest of this paper. Furthermore, the lemmas below will also have important usage in Sections [V] and [VI].

**Lemma 2:** Under the properties in Theorem 2, consider the frames \( f_1, o_1, \ldots, o_k \) such that \( f_1 \in GOP_i \), none of \( o_1, \ldots, o_k \) belongs to \( GOP_i \), and they are all irrelevant with respect to \( f_1 \).

1) If they are scheduled in the order of \( f_1, o_1, \ldots, o_k \), then

\[
d_{f_1} < d_{o_r}, 1 \leq r \leq k
\]

and all frames \( \{o_r\}_{r=1}^k \) belong to GOPs with indices greater than \( i \).

2) If they are scheduled in the order of \( o_1, \ldots, o_k, f_1 \), then

\[
d_{f_1} > d_{o_r}, 1 \leq r \leq k
\]

and all frames \( \{o_r\}_{r=1}^k \) belong to GOPs with indices less than \( i \).

**Proof:** See Appendix [V-A]

**Lemma 3:** Under the properties in Theorem 2, consider some frames \( f_1, o_1, \ldots, o_k \) in an MBFS tree, which corresponds to a GOP. Suppose all of \( o_1, \ldots, o_k \) are irrelevant with respect to \( f_1 \).

1) If they are scheduled in the order of \( f_1, o_1, \ldots, o_k \), then

\[
d_{f_1} < d_{o_r}, 1 \leq r \leq k.
\]

2) If they are scheduled in the order of \( o_k, \ldots, o_1, f_1 \), then

\[
d_{f_1} > d_{o_r}, 1 \leq r \leq k.
\]

**Proof:** See Appendix [V-B]

V. **Optimal Schedule for the SIO Class**

This section provides an algorithm to solve the Scheduling Problem for the SIO class. We first show that each transmission sequence has a unique canonical-form transmission sequence, which is a subsequence of some universal sequence. We then select an optimal schedule from the universal sequence through dynamic programming.

**A. SIO Universal Sequence**

First, we discuss how to find a canonical-form transmission sequence \( W \) given an arbitrary transmission sequence. The lemma below indicates that the frames of each GOP must be transmitted together in \( W \). Moreover, due to the second property of Theorem 2, GOPs with earlier deadlines must be transmitted first.

**Lemma 4:** In an SIO canonical-form transmission sequence, for any \( i \), no frame other than the frames of \( GOP_i \) can be scheduled between the frames of \( GOP_i \).

**Proof:** See Appendix [V-C]

For the frames inside a GOP, the properties of isomorphically ordered MBFS trees in Section [VI] can be used to determine the order of transmissions. Lemma 5 indicates that all subtrees in an MBFS tree must be scheduled back-to-back in order to preserve Theorem 2.

**Lemma 5:** In an SIO canonical-form schedule, any frame scheduled between any two frames that are both in a subtree \( T \) must itself be in \( T \).

**Proof:** See Appendix [V-D]

In \( W \), consider an arbitrary node \( x \) and the subtrees \( T(c_1(x)), \ldots, T(c_n(x)) \), where \( c_1(x), \ldots, c_n(x) \) are the children of \( x \) in the MBFS forest sorted in ascending order of display deadlines. The frame \( x \) is sent first according to the first property of Theorem 2. Then, by Lemma 5 the second property of Theorem 2 and isomorphic order, the frames in each \( T(c_i(x)) \) are sent together in the order of \( c_i(x) \). Therefore, to determine the order of transmissions for each GOP in \( W \), we should perform a generalized pre-order tree walk on each tree of the MBFS forest, where the children of each node is visited in ascending order of display deadlines. This procedure uniquely determines \( W \) and has a complexity of \( O(N) \). Therefore, a canonical form transmission sequence is uniquely determined only by the set of selected frames for transmission.

Next, consider the sequence of all frames of the original video. We call the canonical-form transmission sequence of this sequence the SIO universal sequence. Note that since we may hypothetically increase the link capacity until the entire video is successfully schedulable, such a transmission sequence always exist.

**Theorem 3:** For the SIO class of HPSs, the canonical-form of any transmission sequence is a subsequence of the SIO universal sequence.

**Proof:** See Appendix [VI]

Theorem 3 suggests that, given a link capacity limit, a transmission sequence that is a subsequence of the SIO universal sequence and which maximizes the playback quality is an optimal transmission sequence. This will be used in the next subsection to compute an optimal schedule.

**B. Computation of Optimal Schedule**

The following dynamic programming approach solves the Scheduling Problem for the SIO class. First, generate the SIO universal sequence and index its frames as \( 1, 2, \ldots, N \).

Then, define function \( h(j, t) \) as the maximum reward function, if frames \( \{j, \ldots, N\} \) are to be scheduled in the time
interval \([t, \hat{T}]\) assuming all their parents with indices in the range from 1 to \(j - 1\) (if any) are available. From the system model in Section \(\text{III}\) we have
\[
z^* = h(1, 0)
\] (7)
and the boundary conditions
\[
h(N + 1, t) = 0, \quad \forall t \in \mathbb{Z}
\] (8)
\[
h(j, t) = 0, \quad t > \hat{T}.
\] (9)

Furthermore, \(h\) adheres to the following recursive equations, corresponding to the possible actions at time \(t\):
\[
h(j, t) = \max \left\{ \begin{array}{ll}
h(j, t + 1) & q_j + h(j + 1, t + \Delta t_j), \quad d_j - t \geq \Delta t_j \\
h(j + 1, t + \Delta t_j), & d_j - t < \Delta t_j \\
\min\{k: k > j, k \ni j\} & , t). \end{array} \right.
\] (10)

In the above, the first term corresponds to the case where the optimal schedule for \(h(j, t + 1)\) is also optimal for \(h(j, t)\), so no action is needed at \(t\). The second term corresponds to starting to transmit frame \(j\) at time \(t\), and the condition \(d_j - t \geq \Delta t_j\) ensures that the transmission of \(j\) will be successful. Similarly, the third term refers to starting to transmit frame \(j\) at time \(t\) but the condition \(d_j - t < \Delta t_j\) indicates it is not successful, although its transmission can help its descendants to achieve successful transmission. The forth term corresponds to dropping frame \(j\) and moving to inspect the next frame that does not depend on \(j\). Since \(h\) only needs to be computed for \(1 \leq j \leq N, 0 \leq t \leq \hat{T}\), dynamic programming requires \(O(N\hat{T})\) processing time and \(O(N\hat{T})\) memory to determine \(h(1, 0)\) and extract the optimal policy. Furthermore, since \(\hat{T}\) is linear in \(N\) in all practical video codecs with a constant frame rate, we have an overall complexity of \(O(N^2)\).

VI. OPTIMAL SCHEDULE FOR THE QUASI-SIO CLASS

In this section, an optimal transmission sequence for the more general quasi-SIO class is presented, as an extension of the solution in Section \(\text{V}\).

A. Quasi-SIO Universal Sequence

Again, we will first show that all transmission sequences have unique canonical-form transmission sequences, and then they will be shown to be subsequences of a quasi-SIO universal sequence.

Let \(\mathcal{V}\) be a canonical-form transmission sequence of an arbitrary transmission sequence in the quasi-SIO class. From the first property of Theorem \(2\), \(I_i\) will be scheduled prior to the frames in \(N_i\) since \(I_i\) is the ancestor of all of them, and similarly \(I_{i+1}\) is scheduled prior to the frames in \(D_i\).

Consider the following lemmas:

Lemma 6: In \(\mathcal{V}\), if \(f_1\) and \(f_2\) are two frames in \(N_i\) such that no other frame in \(N_i \cup \{I_{i+1}\}\) is scheduled between them, then no frame can be scheduled between them.

Proof: The proof is similar to that of Lemma \(4\).

Lemma 7: In \(\mathcal{V}\), if \(I_{i+1}\) is scheduled after \(f_1 \in N_i\) such that no other frame in \(N_i\) is scheduled between them, then no frame can be scheduled between them.

Proof: See Appendix \(\text{VII-A}\).

Lemma 8: In \(\mathcal{V}\), if \(I_{i+1}\) is scheduled before \(f_2 \in N_i\) such that no other frame in \(N_i\) is scheduled between them, then no frame can be scheduled between them, and \(f_2\) must be a descendant of \(I_{i+1}\).

Proof: See Appendix \(\text{VII-B}\).

Lemma 9: In \(\mathcal{V}\), if \(I_{i+1}\) is scheduled before \(f \in N_{i+1}\) such that no other frame in \(N_{i+1}\) is scheduled between them, then only frames in \(GOP_i\) and frame \(I_{i+2}\) can be scheduled between them.

Proof: See Appendix \(\text{VII-C}\).

Lemmas \(6\), \(7\), and \(8\) jointly indicate that the frames of \(M_i = N_i \cup \{I_{i+1}\}\) must be sent together. Moreover, Lemma \(9\) indicates that \(M_{i+1}\) must be scheduled immediately after \(M_i\), for all \(i\).

Next, we determine the transmission order of frames inside each \(M_i\). Due to the first property of Theorem \(2\) each I-frame \(I_{i+1}\) is scheduled prior to \(D_i\) in \(\mathcal{V}\). With respect to \(I_{i+1}\), the frames of \(GOP_i\) can be divided into two groups: \(A_i\), the frames scheduled prior to \(I_{i+1}\), and \(B_i\), the frames scheduled after \(I_{i+1}\). Clearly all frames of \(D_i\) are in \(B_i\). Since the frames in \(M_i\) must be sent together, the frames in \(A_i\) are sent together, and the same holds for \(B_i\). The definition of quasi-SIO indicates that the frames of \(A_i\) can be scheduled by Lemma \(5\) using the subgraph of \(G\) which comprises the nodes of \(A_i\). Moreover, for \(B_i\) since \(I_{i+1}\) has already been scheduled, the frames of \(B_i\) can be scheduled similarly based on the isomorphic ordering property, the second property of Theorem \(2\) and Lemma \(5\). Hence, all subtrees in \(B_i\) are transmitted as contiguous blocks in the ascending order of the display deadlines of their roots.

It remains to determine the relative position of the frames in \(N_i - D_i\) with respect to \(I_{i+1}\). Let a critical node be defined as \(f, f \in D_i\), such that in the path between \(f\) and the root of the MBFS tree containing \(f\), no other frame in \(D_i\) appears. An example of this is depicted in Figure \(3\) where \(l_1, l_2\) and \(l_2^*\) are critical nodes. It is easy to see that a critical node in \(GOP_i\) is a child of \(I_{i+1}\), but a child of \(I_{i+1}\) in \(GOP_i\) is not necessarily a critical node. Let \(\Gamma_i\) be the set of all critical nodes. It is clear from Figure \(3\) that the set of nodes in the subtrees rooted at the critical nodes of \(GOP_i\) is equivalent to \(D_i\). Furthermore, due to the definition, the members of \(\Gamma_i\) are pair-wise irrelevant. Let \(\zeta_i\) be the member of \(\Gamma_i\) with the smallest display deadline.

Lemma 10: In \(\mathcal{V}\), frame \(\zeta_i\) must be scheduled immediately after \(I_{i+1}\).

Proof: See Appendix \(\text{VII-D}\).

Furthermore, we have the following result.

Lemma 11: In \(\mathcal{V}\), consider frame \(f \in N_i - D_i\) that is irrelevant with respect to \(\zeta_i\).

- If \(d_f < d_{\zeta_i}\), \(f\) is scheduled before \(I_{i+1}\).
- If \(d_f > d_{\zeta_i}\), \(f\) is scheduled after \(I_{i+1}\).

Proof: See Appendix \(\text{VII-E}\).

The above observations indicate that for any quasi-SIO transmission sequence, if we first ignore the backward pre-
special consideration for the parameter belongs to be with boundary conditions

Fig. 3: Critical nodes in the GOP, MBFS tree of the quasi-SIO class, plus edges from \( I_{i+1} \). Each triangle indicates a subtree rooted at the node located at its top.

diction edges from \( I_{i+1} \) to B-frames in GOP, for all \( i \), and create the SIO canonical form transmission sequence, then the quasi-SIO canonical form schedule can be obtained by simply moving \( I_{i+1} \), for all \( i \), in the SIO canonical form transmission sequence backward so that it is positioned immediately prior to the earliest frame of \( D_i \) in the transmission order. Furthermore, \( V \) is uniquely determined given any transmission sequence. Hence, we define the quasi-SIO universal sequence similarly to the SIO universal sequence in the previous section.

Then, similar to Section \[V\] we have the following theorem, which will be exploited to derive the optimal schedule given a link capacity limit.

**Theorem 4**: For the quasi-SIO class of HPSs, the canonical-form of any transmission sequence is a subsequence of the quasi-SIO universal sequence.

**Proof**: See Appendix \[VIII\] ■

**B. Computation of Optimal Schedule**

First, generate the quasi-SIO universal sequence and index the frames by \( 1, 2, \ldots, N \).

Then, define function \( g(j, t, s) \), similarly to \( h(j, t) \), as the maximum sum quality of successful frames, if frames \( \{j, \ldots, N\} \) are to be scheduled in the time interval \([t, T]\) assuming all their parents with indices in the range from 1 to \( j - 1 \) (if any) are available. The additional parameter \( s \) is an integer in \( \{0, 1, 2, 3\} \), whose binary representation specifies the status of the two nearest I-frames in the quasi-SIO universal sequence that precede \( j \), i.e., \( s = (x_{1}(j), x_{2}(j)) \), where \( x_{2}(j) = 1 \) if the nearest preceding I-frame is scheduled for transmission and \( 0 \) otherwise, and \( x_{1}(j) \) is similarly defined concerning the second nearest preceding I-frame.

Similarly to Section \[VIII\] we have

\[
z^* = g(1,0,0),
\]

with boundary conditions

\[
g(N + 1, t, s) = 0, \quad \forall t \in \mathbb{Z}, \forall s
\]

\[
g(j, t, s) = 0, \quad t > T.
\]

The recursive equations for \( g(j, t, s) \) are given below, with special consideration for the parameter \( s \).

1) If \( j \) is not an I-frame, let the index of the GOP that \( j \) belongs to be \( i_{j} \). Three cases are considered:

1a) Suppose \( j \in A_{i_{j}} \). This is the case where \( j \) does not depend on \( I_{i_{j}+1} \) and is located prior to \( I_{i_{j}+1} \) in the quasi-SIO universal sequence. If \( I_{i_{j}} \), the nearest preceding I-frame, is transmitted, then the outcome is similar to Section \[V\] otherwise, \( j \) should not be selected. Hence, for \( s = 1, 3 \), we have

\[
g(j, t, s) = \max \left\{ g(j, t+1, s), g(j+1, t+\Delta t_j, s), \delta_j - t \geq \Delta t_j, g(j+1, t+\Delta t_j, s), \delta_j - t < \Delta t_j, g(\min\{k : k > j, k \neq j\}, t, s) \right\}
\]

and for \( s = 0, 2 \), we have

\[
g(j, t, s) = \max \left\{ g(j, t+1, s), g(\min\{k : k > j, k \neq j\}, t, s) \right\}
\]

1b) Suppose \( j \in D_{i_{j}} \). This is the case where \( j \) depends on both \( I_{i_{j}} \) and \( I_{i_{j}+1} \) and is located behind both in the quasi-SIO universal sequence. If both I-frames are transmitted, then the outcome is similar to Section \[V\] otherwise, \( j \) should not be selected. Hence, for \( s = 3 \), we use \[14\], and for \( s = 0, 1, 2 \), we use \[15\].

1c) Suppose \( j \in B_{i_{j}} - D_{i_{j}} \). This is the case where \( j \) does not depend on \( I_{i_{j}+1} \) but is located behind \( I_{i_{j}+1} \) in the quasi-SIO universal sequence. If \( I_{i_{j}} \), the second nearest preceding I-frame, is transmitted, then the outcome is similar to Section \[V\] otherwise, \( j \) should not be selected. Hence, for \( s = 2, 3 \), we use \[14\], and for \( s = 0, 1 \), we use \[15\].

2) If \( j \) is an I-frame, then we also need to update \( s \) besides scheduling the frame. Hence,

\[
g(j, t, s) = \max \left\{ g(j, t+1, s), g(j+1, t+\Delta t_j, (2s + 1 \mod 4)), \delta_j - t \geq \Delta t_j, g(j+1, t+\Delta t_j, (2s + 1 \mod 4)), \delta_j - t < \Delta t_j, g(\bar{I}_j, t, (2s) \mod 4), \right\}
\]

where \( \bar{I}_j \) is the next I-frame after \( j \), and we update the two bits of \( s \) depending on the scheduling outcome.

In the above dynamic programming formulation, state \( s \) needs to take only four values due to the fact that the frames of a GOP only depend on at most two I-frames. Therefore, we again have complexity \( O(NT) = O(N^2) \).

**VII. EXPERIMENTS WITH VIDEO TRACES**

Even though the proposed schedule has been proven optimal, we are interested in its numeric performance when applied to real video traces.

**A. Methodology**

The proposed scheme is simulated in Matlab, using H.264 video traces provided by [31]. We present here results for the “NBC News (352 × 288)” and “SONY (1920 × 1080)” traces, representing faster moving and slower moving scenes,
respectively. We take the first 305 frames of each video. Both videos use the $G16B15$ hierarchical dyadic structure at 30 frames per second. The luminance of a frame is used as its quality measure, so the objective function averaged over the number of frames represents the average Y-PSNR at the receiver [7].

We compare the optimal solution with two suboptimal alternatives. The first is a best-effort earliest-deadline-first (EDF) algorithm, in which the frames are scheduled in their display order, and the frames that cannot be successful are dropped by the transmitter. EDF is the traditional scheme used for video streaming [4]. The second is a priority-based earliest-deadline-first (PBEDF) algorithm, which combines frame prioritization [11][12] with EDF. In PBEDF, the set of frames are partitioned into blocks of size $M$. In each block, the I-frames are scheduled first in EDF order, and the B-frames are scheduled last in EDF order. For fair comparison, in the results below we always use an optimal $M$ obtained by exhaustive search.

B. Experimental Results

Video playback quality is dictated by the channel rate and the initial playback delay, which is defined as the time the receiver waits before video playback while receiving data. In Figures [4] and [5] we consider different initial delays of 0.1, 1, and 5 seconds, and for each initial delay, we study all values of link capacity $C$ up to the point when all algorithms give lossless performance. Note that since the video trace “SONY” has very low motion, its data size is relatively smaller, so that high PSNR can be achieved even for its HD resolution.

We observe that, for a wide range of parameter settings, the optimal schedule substantially outperforms the sub-optimal alternatives. The exception is only under extremely relaxed environments, e.g., when the initial delay is large or when the link capacity is high. In practical video streaming, where the users are inpatient, and the network bandwidth limited, the benefit of the optimal schedule is apparent.

In addition, under the optimal schedule, the Y-PSNR is monotonically increasing in the link capacity as expected. This is not the case for EDF and PBEDF, which results from the dependence structure. In particular, a slightly higher link capacity may drive EDF and PBEDF to over zealously transmit a frame, which in turn reduces the time left to transmit an ancestor that is later in the display order, possibly rendering the ancestor unsuccessful. Hence, with these sub-optimal schemes, increasing the link capacity is not always beneficial.

VIII. CONCLUSIONS

A frame scheduling scheme has been developed for optimal video streaming over a link with limited capacity. For two general classes of HPSs, it is shown that an optimal transmission sequence can be found as a subsequence of some optimal universal sequence in term of frame indices. Efficient dynamic programming solutions are proposed to identify the optimal schedules with polynomial complexity. Experiments with video traces show that the optimal schedules can substantially improve the playback quality over sub-optimal alternatives.

[Fig. 4: Y-PSNR vs. link capacity for video NBC News ($352 \times 288$).]

APPENDIX I

PROOF OF THEOREM

There are two possible scenarios where the frames are not transmitted one-by-one in some part of a schedule. First, the transmission of a frame may be interrupted by the transmission of parts of other frames (preemptive transmission). Second, parts of multiple frames are transmitted simultaneously. We present in the following how to transform the schedule in both scenarios to one-by-one frame transmission without incurring new delay in the transmission end time of any frame.

In the first scenario, as illustrated in Figure 5, supposed $A_1$ and $A_2$ are consecutive fragments of a frame, and some other frames (or parts of frames) are scheduled between them.
frames are sent one-by-one under the same link capacity, then clearly the amount of data transmitted in this interval will not change and all the frames can be transmitted before time $t_2$. Next, consider the case where the frames partially overlap each other. In this case, time can be divided into subintervals, such that in each of them, the frame segments completely overlap each other. Then the same transformation above can be carried out in each subinterval, to produce a schedule without simultaneous transmissions but with preemptive transmissions. This is then reduced to the first scenario.

**Appendix II**

**Proof of Lemma 1**

Lemma 12: Let $x$ and $y$ be two irrelevant nodes in an isomorphically ordered tree. A unique node $w$ and unique indices $i$ and $j$, $i \neq j$, exist such that $x$ belongs to $T(c_i(w))$ and $y$ belongs to $T(c_j(w))$.

Proof: Let $u_1, \ldots, u_{m(x)}$ and $v_1, \ldots, v_{m(y)}$ correspond to the paths from the root of the tree to $x$ and $y$ respectively. Clearly $u_1 = v_1$ is the root, but the two sequences diverge later. Therefore, there is a unique index $\sigma$ such that $u_\sigma = v_\sigma$ but $u_{\sigma+1} \neq v_{\sigma+1}$. Clearly, $u_{\sigma+1}$ and $v_{\sigma+1}$ are children of $u_\sigma$ in the tree. In addition, $x$ resides in $T(u_{\sigma+1})$ while $y$ resides in $T(v_{\sigma+1})$.

Now back to the proof of Lemma 1 if $x$ and $y$ do not belong to the same GOP, then the lemma clearly holds. Now suppose they belong to the same GOP. Based on Lemma 12 there exist a node $w$ and indices $i$ and $j$ such that $x$ belongs to $T(c_i(w))$ and $y$ belongs to $T(c_j(w))$. If $i < j$, then the definition of isomorphically ordered trees gives $\max(T(c_i(w))) \leq \min(T(c_j(w)))$, but since none of the playback deadlines are equal, we have $\max(T(x)) < \min(T(y))$. If $i > j$, it can be shown similarly $\min(T(x)) > \max(T(y))$ which is not possible due to $d_x < d_y$. As a result, $i < j$ and $\max(T(x)) < \min(T(y))$.

**Appendix III**

**Hierarchical Dyadic Structure**

We first present a mathematical characterization of the hierarchical dyadic structure and then show that it belongs to the quasi-SIO class.

In the hierarchical dyadic structure $G_nB_m$, the GOP size $n$ is an integer power of 2, and $m$ is the number of B-frames between consecutive non-B-frames, with $m = 2^\omega - 1$ for some $\omega \in \mathbb{N}$. Each GOP contains one leading I-frame and $n/m + 1$ P-frames. Each P-frame depends on the previous I-frame/P-frame in the display order. The dependency structure among the B-frames between two consecutive non-B-frames is described by Dyadic-build$(i,j)$ in Algorithm 2 where $j - i$ is
an integer power of 2. Note that in this algorithm, the frames of the video sequence are indexed in the display order as integers.

**Algorithm 2** Dyadic-build\((i, j)\)

1. if \(|i - j| \leq 1\) or \(\log_{2} |i - j| \notin \mathbb{N}\) then
2. return
3. end if
4. \(i_0 \leftarrow \frac{i + j}{2}\)
5. \(i_0\) depends on \(i, j\)
6. Dyadic-build\((i, i_0)\)
7. Dyadic-build\((i_0, j)\)

As an example, Figure 1 shows the hierarchical dyadic structure for a GOP with \(G16B3\). A sequence of frames \(i, \ldots, j\) is called a complete sequence if, except \(i\) and \(j\), none of the frames in the sequence have parents outside this sequence. Thus, for a complete sequence, if the frames \(i\) and \(j\) are available, all frames in the sequence can be decoded. An example of a complete sequence is a GOP. It can easily be shown through induction that, in the context of the hierarchical dyadic structure, the size of any complete sequence is \(2^{\omega} + 1\), for some \(\omega \in \mathbb{N}\).

The hierarchical dyadic structure does not belong to the SIO class. This is because there is always a B-frame that is the descendant of two consecutive I-frames, which violates the sequential property. In Figure 1, frame 14 is an example of such a frame that violates the sequential property.

However, in what follows, we show that the hierarchical dyadic structure belongs to the quasi-SIO class. This is accomplished by demonstrating that a modification on the hierarchical dyadic structure, which removes the backward DAG edges emanating from each I-frame to its children in the preceding GOP, results in a structure belonging to the SIO class. In this modified hierarchical dyadic structure, no B-frame depends on a succeeding I-frame. For instance, this means in Figure 1, frames 14 and 15 no longer depend on frame 16. It is worth mentioning that this modification is used as an approximation in [16]. However, in our work, we show that an optimal schedule can be obtained for the hierarchical dyadic structure without modification, since it is a special case of the quasi-SIO class.

**Theorem 5:** The DAG of a modified hierarchical dyadic structure is sequential.

**Proof:** The MBFS tree corresponding to each GOP is always rooted at the GOP’s leading I-frame. If all B-frames are removed, only some subtrees of the MBFS forest are removed, and the resultant DAG consisting of only I-frames and P-frames is clearly sequential. Therefore, it suffices to only prove that the sequential property is preserved when the MBFS algorithm is executed on B-frames between consecutive non-B-frames.

We use mathematical induction. In particular, since the number of B-frames between two consecutive non-B-frames is \(2^{\omega} - 1\), for some \(\omega \in \mathbb{N}\), induction is carried out on \(\omega\). In the following, we address two cases based on whether the latter non-B-frame is 1) an I-frame or 2) a P-frame.

For the basis step, consider \(\omega = 1\). This situation is depicted in Figures 8(a) and 8(b) for cases 1) and 2) respectively. Note that in case 1), since frame \(z\) is an I-frame, the edge from \(z\) to \(y\) is removed in the modified hierarchical dyadic structure. In both cases, the MBFS tree clearly satisfies the sequential property.

For the induction step, suppose the statement holds for \(\omega = k\). This is shown in Figures 9(a) and 9(b) for cases 1) and 2) respectively. In both cases, the number of frames between the two end frames is \(2^{k+1} - 1\), and the resulting tree satisfies the sequential property. The triangles in both figures indicate the subtree rooted at node \(y\). We consider \(\omega = k + 1\) for the two cases separately.

**Case 1:**

Figure 10(a) depicts this situation. There are \(2^{k+1} - 1\) nodes between \(x\) and \(u\). Node \(z\) is located halfway between them. In addition, node \(y\) is halfway between \(x\) and \(z\), and node \(u\) is halfway between \(z\) and \(w\). The dependency structure among these indicated nodes is displayed, and the rest of the dependency structure, which is not shown, is determined by the modified hierarchical dyadic structure.

The MBFS algorithm first visits \(x\), then to \(z\), \(y\), and finally \(u\). It can be observed that the dependency structure
for frames between \( x \) and \( z \) is the same as Figure 9(b) and the dependency structure for frames between \( z \) and \( w \) is the same as Figure 9(a). Hence, the triangles in Figure 10(a) refer to the same triangles as in Figures 9(a) and 9(b).

For nodes \( x, y, z, \) and \( u \), it is easy to see that the sequential property holds. Within the subtree rooted at \( y \), the sequential property is satisfied. Now consider the additional ancestors \( x \) and \( z \). All paths from \( x \) to nodes in the subtree rooted at \( y \) pass through \( z \). Therefore for all nodes in the subtree rooted at \( y \), the sequential property is preserved. Similarly, the same holds for the subtree rooted at \( u \).

**Case 2):**

Figure 10(b) depicts this situation. Similar to the previous case, again there are \( 2^k + 1 - 1 \) nodes between nodes \( x \) and \( w \). Node \( z \) is located halfway between them. Node \( y \) is halfway between \( x \) and \( z \), and node \( u \) is halfway between \( z \) and \( w \). The dependency structure among these indicated nodes is displayed, and the rest of dependency structure, which is not shown, is determined by the modified hierarchical dyadic structure.

The MBFS algorithm first visits \( x \). Different from the previous case, however, it then visits \( w \), before moving to \( z \), \( y \), and finally \( u \). When it visits \( z \), nodes \( x \) and \( w \) have already been visited, so the remaining two branches of the dependency structure, from \( z \) to \( y \) and from \( z \) to \( u \), each resembles that of Figure 9(a). In particular, \( z \) in Figure 10(b) plays the role of \( x \) in Figure 9(a) while \( y \) and \( u \) in Figure 10(b) plays the role of \( y \) in Figure 9(a).

Similar to the previous case, for nodes \( x, w, z, y, \) and \( u \), it is easy to see that the sequential property holds. Within the subtree rooted at \( y \), due to the inductive hypothesis in Figure 9(a), the sequential property is satisfied. Now consider the additional ancestors \( x, w, \) and \( z \). All paths from \( x \) to nodes in the subtree rooted at \( y \) pass through \( w, z, \) and \( y \). Therefore for all nodes in the subtree rooted at \( y \), the sequential property is preserved. Similarly, the same holds for the subtree rooted at \( u \).

**Theorem 6:** The MBFS trees under the hierarchical dyadic structure (modified or not) are Binary Search Trees (BSTs) with respect to the display deadlines.

**Proof:** Consider a hierarchical dyadic GOP with \( GnBm \). To show that its corresponding MBFS tree is a BST, we need to verify the following three properties for every node in a BST: the node has at most two children, the subtree rooted at its left child contains only nodes with deadlines earlier than that of the node, and the subtree rooted at its right child contains only nodes with deadlines later than that of the node.

Since in the creation of each MBFS tree, the dependencies on the succeeding I-frame does not apply, in what follows, we only need to consider a modified version of Algorithm 2 that omits such dependencies. Assume the video sequence is sorted and indexed in display order by integers. We next consider each node in the DAG \( G \) and show that when the MBFS algorithm visits the node, the BST properties are satisfied. This is carried out for each type of nodes in the following.

- **I-frame:** The I-frame is always the root of its own MBFS tree. If \( m < n - 1 \), then there exists at least one P-frame in the GOP. After the MBFS algorithm visits this I-frame, among all of its children, only the first P-frame in display order is decodable, since the other children of the I-frame are also descendants of the first P-frame. Hence, the I-frame has only a single child with respect to the MBFS tree, so the BST properties are satisfied.

  If \( m = n - 1 \), then the B-frame in the mid point of the GOP depends only on this I-frame in the modified structure, so it acts as a P-frame. Similarly, this B-frame is the only child of the I-frame with respect to the MBFS tree, so the BST properties are satisfied.

- **P-frame:** Let the considered P-frame be \( i_0 \). The preceding and succeeding non-B-frames are the frames labelled \( i_0 - (m + 1) \) and \( i_0 + (m + 1) \) respectively. All children of \( i_0 \) are between these two frames.

  Note that the subsequence \( i_0 - (m + 1), \ldots, i_0 \) is a complete sequence. Among these frames, after the MBFS algorithm visits \( i_0 \), the only decodable child of \( i_0 \) is \( i_0 + (m + 1) \), since the other children of \( i_0 \) are all descendants of \( i_0 + (m + 1) \). The subsequence \( i_0, i_0 + (m + 1) \) is also a complete sequence, but we need to consider two cases depending on whether frame \( i_0 + (m + 1) \) is an I-frame or a P-frame. If \( i_0 + (m + 1) \) is an I-frame, then after the MBFS algorithm visits \( i_0 \), the only decodable child of \( i_0 \) among \( i_0, \ldots, i_0 + (m + 1) \) is \( i_0 + (m + 1) \), since the other children all depend on \( i_0 \). If \( i_0 + (m + 1) \) is a P-frame, then after the MBFS algorithm visits \( i_0 \), the only decodable child of \( i_0 \) among \( i_0, \ldots, i_0 + (m + 1) \) is \( i_0 + (m + 1) \), since the other children all depend on \( i_0 + (m + 1) \). In both cases, \( i_0 \) has only one decodable child among \( i_0, \ldots, i_0 + (m + 1) \). As a result, \( i_0 \) has exact two children in the MBFS
tree. Moreover, the sequences \( i_0 - (m + 1), \ldots, i_0 - 1 \) and \( i_0 + 1, \ldots, i_0 + (m + 1) \) are completely irrelevant, and all of the former have deadlines earlier than \( d_{i_0} \) and all of the latter have deadlines later than \( d_{i_0} \). Therefore, all \( \text{BST} \) properties are satisfied.

- \( \text{B-frame} \): Let the considered \( \text{B-frame} \) be \( i_0 \). This is similar to the previous case, except that instead of frames between \( i_0 - (m + 1) \) and \( i_0 + (m + 1) \), we should consider frames \( i_0 - (m_0 + 1), \ldots, i_0 + (m_0 + 1) \) for some \( m_0 \) such that \( m_0 < m + 1 \) and \( i_0 - (m_0 + 1), \ldots, i_0 + (m_0 + 1) \) is the largest complete sequence centered at \( i_0 \). We similarly see that, after the MBFS algorithm visits \( i_0 \), the only decodable children of \( i_0 \) are \( \frac{L - (m_0 + 1)}{2} \) and \( \frac{L + (i_0 + (m_0 + 1))}{2} \). The other properties of \( \text{BST} \) are also similarly satisfied.

The \( \text{BSTs} \) are a special case of isomorphically ordered trees. Combining this with Theorem \( [5] \) we see that the modified hierarchical dyadic structure belongs to the \( \text{SIO} \) class, and hence, the hierarchical dyadic structure belongs to the quasi-\( \text{SIO} \) class.

**APPENDIX IV**

**PROOF OF THEOREM 2**

Given a transmission sequence that does not meet either or both the two properties, it will be modified to meet those properties in two steps.

**First Step:** With the assumption that the first property does not hold, there must be frames \( l_1 \) and \( l_2 \) such that frame \( l_1 \) is the ancestor of frame \( l_2 \) but \( l_2 \) is scheduled before \( l_1 \). An example of this is shown in Figure 11(a). Let \( K_{l_2} = \{ \text{descendants of } l_2 \text{ scheduled after } l_2 \text{ and before } l_1 \} \). Also let the index of the timeslot in which the transmission of \( l_2 \) starts be \( e_1 \) and the index of the timeslot in which the transmission of \( l_1 \) ends be \( e_2 \). If any frame in the set \( K_{l_2} \) is successful, then since the transmission of frame \( l_1 \) has to terminate prior to the deadlines of its successful descendants, we have:

\[
e_2 \leq d_{i_l}, \forall l \in K_{l_1} \cup \{l_2\}, \text{ } l \text{ is successful} \quad (17)
\]

Now consider the frames in the interval \( [e_1, e_2] \). They can be divided into three sets: \( \{ i_1 \} \), \( A_1 = K_{l_2} \cup \{ l_2 \} \), and \( A_2 = \{ \text{all frames in interval } [e_1, e_2] \} - (A_1 \cup \{ i_1 \}) \). Now starting from timeslot \( e_1 \) first schedule \( A_2 \), then \( l_1 \), and then \( A_1 \), such that the order of frames in \( A_2 \) in the original sequence is preserved and also the same for \( A_1 \). An example of this operation is shown in Figure 11(b).

Any previously successful frame in the time interval \( [e_1, e_2] \) remains successful. In particular, since all frames in \( A_2 \) either move backward in time or do not change position, the previously successful ones remain successful. Moreover, due to inequality \( (17) \) all previously successful frames in \( A_1 \) remain successful as well. Furthermore, no new violation of the first property is created. Since this operation always repairs at least one pair that violates the first property, and the number violations is finite, repetition of the operation will end in an solution which satisfies the first property.

**SECOND PROPERTY**

First Step: Let \( U \) be the set of frames scheduled between \( l_1 \) and \( l_2 \). Suppose all of them are irrelevant with respect to \( l_2 \). Let \( t_0 \) be the timeslot when the transmission of \( l_2 \) starts and \( t_1 \) be the timeslot when the transmission of \( l_2 \) ends. If \( l_i \) is successful, then \( t_1 \leq d_{i_l} < d_{i_l} \).

Else, if \( l_i \) is not successful, then it has to have a successful descendant, namely \( f_i \). Let \( t_2 \) be the timeslot when the transmission of \( f_i \) ends. Since \( f_i \) is successful, \( t_2 \leq d_{f_i} \). Furthermore, from the First step, we have \( t_1 < t_2 \). We next consider two cases based on whether \( l_2 \) is an I-frame. If \( l_2 \) is not an I-frame, then \( f_i \) belongs to the same tree as \( l_2 \) in the MBFS forest. Hence, Lemma \( [1] \) indicates \( d_{f_i} < d_{l_2} \). If \( l_2 \) is an I-frame, let \( n_{i_l} \) be the index of the GOP that \( l_i \) belongs to. Then since \( d_{i_l} < d_{f_i} \), the frame \( l_2 \) has to belong to a GOP with an index higher than \( n_{i_l} \). But, the children of \( l_i \) are only in \( \text{GOP}_{n_{i_l} - 1} \) and \( \text{GOP}_{n_{i_l}} \). As a result, \( d_{f_i} < d_{l_2} \). Hence, in both cases, we have \( t_1 < t_2 < d_{i_l} \).

Now, we can re-order the frames starting at \( t_0 \) as follows: \( U \), then \( l_2 \), and then \( l_1 \). This clearly will not harm \( l_i \) and \( l_2 \). In addition, the frames of \( U \) are moved backward in time, so none of them is harmed either. Furthermore, none of the frames outside of \( [i_0, t_1] \) are affected. This operation is illustrated in Figure 12. A special case of this is when \( U \) is empty, i.e., when \( l_2 \) and \( l_1 \) are neighbors.

**APPENDIX V**

**PROOF OF LEMMAS 2, 5**

**A. Proof of Lemma 2**

We use mathematical induction to prove the first part.

For the basis step, consider \( k = 1 \). Since \( o_1 \) and \( f_1 \) are from different GOPs and they are irrelevant, due to the second property of Theorem 2, \( d_{f_1} < d_{o_1} \) and \( o_1 \) belongs to a GOP with an index greater than \( i \).

For the induction step, suppose the statement holds for \( k = m \). That is, \( d_{f_1} < d_{o_m} \) and \( o_m \) belongs to some \( \text{GOP}_j \) with \( j > i \). The frame \( o_{m+1} \) either is a descendant of \( o_m \) or is irrelevant with respect to \( o_m \).

If \( o_m+1 \) is a descendant of \( o_m \), then it belongs to \( \text{GOP}_j \) or \( \text{GOP}_{j-1} \). In the former case, we have \( j > i \) and hence \( d_{f_1} < d_{o_{m+1}} \). In the latter case, since \( o_{m+1} \) and \( f_1 \) do not belong to the same GOP, \( j - 1 \neq i \). Combining this with \( j > i \), we have \( j - 1 > i \) and hence \( d_{f_1} < d_{o_{m+1}} \).

If \( o_m+1 \) is irrelevant with respect to \( o_m \), due to the second property of Theorem 2, \( d_{o_m} < d_{o_{m+1}} \) and \( o_{m+1} \) cannot belong to a GOP with an index less than \( j \). As a result, \( d_{f_1} < d_{o_{m+1}} \) and \( o_{m+1} \) belongs to a GOP with an index larger than \( i \).
The second part can be similarly proven using mathematical induction.

**B. Proof of Lemma 3**

We prove the first statement using mathematical induction.

For the basis step, consider $k = 1$. Since $o_1$ is scheduled after $f_1$ and they are irrelevant, due to the second property of Theorem 2, we have $d_{f_1} < d_{o_1}$.

For the induction step, suppose the statement holds for $k = m$. Consider the case $k = m + 1$. We first note that none of $o_1, \ldots, o_k$ can be an I-frame, since otherwise $f_1$ would be scheduled after the I-frame. The inductive hypothesis says $d_{f_1} < d_{o_m}$. The frame $o_{m+1}$ either is a descendant of $o_m$ or is irrelevant with respect to $o_m$. If it is a descendant, then according to Lemma 1, $d_{f_1} < d_{o_{m+1}}$. If $o_{m+1}$ is irrelevant with respect to $o_m$, due to the second property of Theorem 2, $d_{o_m} < d_{o_{m+1}}$. As a result, $d_{f_1} < d_{o_{m+1}}$.

The second statement can be proved similarly.

**C. Proof of Lemma 2**

Assume towards a contradiction that the lemma does not hold for some $i$. Then it is easy to see that there must exist two frames, $f_1$ and $f_2$, in $GOP$, such that none of the frames scheduled between them belong to $GOP$. Let the frames between $f_1$ and $f_2$ be $o_1, \ldots, o_k$. In the SIO class, all of these frames are irrelevant with respect to $GOP$. In particular, they are irrelevant with respect to $f_1$. As a result, Lemma 2 suggests $o_k$ belongs to a $GOP$ such that $j > i$, which implies that $d_{f_2} < d_{o_k}$. Noting that $o_k$ and $f_2$ are neighbors, this violates the second property of Theorem 2.

**D. Proof of Lemma 5**

Assume towards a contradiction that the lemma does not hold. Then it can be shown easily that frames $l'$, $l_1$, and $l_2$ exist such that $l_1, l_2 \in T(l')$ and none of the frames scheduled between $l_1$ and $l_2$ belong to $T(l')$.

Without loss of generality, suppose $l_1$ is scheduled earlier than $l_2$. Index the frames scheduled between $l_1$ and $l_2$ in the ascending order of transmission starting time to obtain $j_1, \ldots, j_{k'}$ for some integer $k'$. These frames are irrelevant with respect to $l'$ and therefore irrelevant with respect to $l_1$ and $l_2$. Hence, according to Lemma 5

$$d_{l_1} < d_{j_m} < d_{l_2}, 1 \leq m \leq k'. \quad (18)$$

Now consider any specific $j_m$ among $j_1, \ldots, j_{k'}$. Lemma 1 suggests that either $\max(T(l')) < \min(T(j_m))$ or $\min(T(l')) > \max(T(j_m))$ depending on the relative sizes of $d_{j_m}$ and $d_{l'}$. In the former case, since $l_1$ and $l_2$ belong to $T(l')$, both $d_{l_1}$ and $d_{l_2}$ are smaller than $d_{j_m}$, while in the latter case, similarly both $d_{l_1}$ and $d_{l_2}$ are greater than $d_{j_m}$. This contradicts $18$.

**APPENDIX VI**

**PROOF OF THEOREM 5**

Let $W$ be the canonical-form of a transmission sequence, with associated graph $G_W \subset G$. Since $G_W$ can be obtained from $G$ by removing the nodes that are not scheduled in $W$ and their descendants in the MBFS forest, the roots of trees in $G_W$ are roots in $G$ as well. Moreover, Lemma 5 indicates that a node and its descendants in the MBFS forest are scheduled together. This, together with the fact that the pre-order tree walk is executed on both $G$ and $G_W$, implies that $W$ can be obtained from the SIO universal sequence by removing subtrees, which corresponds to dropping frames.

**APPENDIX VII**

**PROOF OF LEMMATA 7-11**

**A. Proof of Lemma 7**

Towards a contradiction, suppose some frames $o_1, \ldots, o_k$ are scheduled between them. Since none of them are in $N_i$, they are all irrelevant with respect to $f_1$, so Lemma 2 indicates $d_{f_1} < d_{o_k}$ and $o_k$ belongs to some $GOP_j$ such that $j > i$. However, if $j = i + 1$, this will violate the first property of Theorem 2, since $o_k$ has appeared prior to $I_{i+1}$ in the schedule; if $j > i + 1$, then $d_{f_{i+1}} < d_{o_k}$, which violates the second property of Theorem 2.

**B. Proof of Lemma 8**

Towards a contradiction, suppose some frames $v_1, \ldots, v_k$ are scheduled between them. Since none of them are in $N_i$, they are all irrelevant with respect to $f_2$, so Lemma 2 indicates $d_{v_1} < d_{f_2}$ and $v_1$ belongs to some $GOP_j$ such that $j < i$. As a result, $v_1$ and $I_{i+1}$ are irrelevant and $d_{v_1} < d_{I_{i+1}}$. This violates the second property of Theorem 2.

In addition, $f_2$ must be a descendant of $I_{i+1}$, since otherwise it will violate the second property of Theorem 2.

**C. Proof of Lemma 9**

Let the frames between $I_{i+1}$ and $f$ be $v_1, \ldots, v_k$. Two cases are considered: first, $I_{i+2}$ is among them, and second, it is not.

In the first case, Lemma 5 indicates $v_k = I_{i+2}$. Then, none of the frames $v_1, \ldots, v_{k-1}$ is an ancestor of $v_k$. Furthermore, the first property of Theorem 2 suggests that none of them can be a descendant of $v_k$ either. Hence, all frames $v_1, \ldots, v_{k-1}$ are irrelevant with respect to $v_k$, and Lemma 2 indicates they all belong to GOPs with indices less than $i + 2$. Since $v_1, \ldots, v_k$ do not belong to $N_{i+1}$, they all belong to GOPs with indices less than $i + 1$. The same result can be obtained in the second case, with Lemma 2 applied to frame $f$ instead of $v_k$.

Furthermore, in both cases, $v_1$ cannot be irrelevant with respect to $I_{i+1}$, since that will violate the second property of Theorem 2. Then the first property of Theorem 2 indicates $v_1$ must be a descendant of $I_{i+1}$. As a result, $v_1 \in N_i$ in both cases. Now let $\xi = \max\{i : 1 \leq i \leq k, v_i \in N_i\}$. Lemma 6 indicates that $v_1, \ldots, v_\xi$ all belong to $N_i$.

In the first case, suppose $\xi < k - 1$. Then the frames $v_{\xi+1}, \ldots, v_{k-1}$ belong to neither $GOP_i$ nor $GOP_{i+1}$. Therefore, they are all irrelevant with respect to $v_\xi$, and due to Lemma 2 they all belong to GOPs with indices higher than $i$. In addition, since none of them belong to $N_{i+1},$ they all belong to GOPs with indices higher than $i + 1$. But this contradicts the result above, so $\xi = k - 1$. Similarly, in the second case, we can show that $\xi = k$. 


D. Proof of Lemma 11

Towards a contradiction, let \( o_1, \ldots, o_k \) be the sequence of frames scheduled between \( I_{i+1} \) and \( \zeta_i \). These frames belong to \( N_i \) based on Lemma 8. Frame \( o_1 \) cannot be in \( N_i - D_i \), since otherwise the second property of Theorem 2 would be violated, due to \( d_{o_1} < d_{I_{i+1}} \). Hence, \( o_1 \in D_i \), and \( o_1 \) is a critical node. Then by the definition of \( \zeta_i \), \( d_{\zeta_i} < d_{o_1} \), and \( \zeta_i \) and \( o_1 \) are irrelevant. Furthermore, by Theorem 2 each frame \( o_k, k > 1 \), is either the descendant of \( o_{k-1} \) or irrelevant with respect to \( o_{k-1} \), and \( d_{o_{k-1}} < d_{o_k} \). Hence, the frames \( o_2, \ldots, o_k \) are either descendants of \( o_1 \), or irrelevant with respect to \( o_1 \) but have a display deadline greater than \( d_{o_1} \). Frames in the latter case cannot be ancestors of \( \zeta_i \), since otherwise according to Lemma 1 \( d_{o_1} < d_{\zeta_i} \), which is a contradiction. Therefore, \( o_2, \ldots, o_k \) are irrelevant with respect to \( \zeta_i \). In addition, based on Lemma 1 all of \( o_1, \ldots, o_k \), especially \( o_k \), have a deadline greater than \( d_{\zeta_i} \). However, since \( o_k \) and \( \zeta_i \) are neighbors, this violates the second property of Theorem 2.

E. Proof of Lemma 12

For the first statement, assume towards a contradiction that the statement does not hold, i.e., there exists some frame in \( N_i - D_i \), such that it is irrelevant with respect to \( \zeta_i \) and has a deadline earlier than \( d_{\zeta_i} \), and it is scheduled after \( I_{i+1} \). Let \( x \) be the one of such frames that has the earliest scheduled transmission start time. Then, by definition \( d_x < d_{\zeta_i} \). Furthermore, let \( y \) be the frame scheduled immediately prior to \( x \). As explained previously, Lemmas 6, 7, 8, and 9 jointly dictate that in the quasi-SIO canonical schedule all frames in \( N_i \cup \{I_{i+1}\} \) must be sent together. Hence, either \( y \in N_i \) or \( y = I_{i+1} \). However, since \( x \neq \zeta_i \), by Lemma 10 \( y \) cannot be \( I_{i+1} \). Therefore, \( y \) is scheduled between \( I_{i+1} \) and \( x \), and only the following two cases are left for \( y \):

1) \( y \in N_i - D_i \)

In this case, since \( y \notin D_i \) clearly \( y \) is not \( \zeta_i \) nor a descendant of \( \zeta_i \). Furthermore, by the first property of Theorem 2 and Lemma 10 all ancestors of \( \zeta_i \) other than \( I_{i+1} \) must be scheduled ahead of \( I_{i+1} \). Therefore, \( y \) is not an ancestor of \( \zeta_i \). Hence, \( y \) is irrelevant with respect to \( \zeta_i \). Furthermore, we have \( d_{\zeta_i} < d_y \), since otherwise the existence of \( y \) would contradict the definition of \( x \).

This leads to two observations. First, \( d_x < d_y \), which implies that \( x \) and \( y \) cannot be irrelevant, due to the second property of Theorem 2. Second, Lemma 1 suggests that \( \max(T(\zeta_i)) < \min(T(y)) \). This implies that \( y \) cannot be an ancestor of \( x \), since otherwise we would have \( d_{\zeta_i} < d_x \), contradicting the definition of \( x \).

Finally, the first property of Theorem 2 suggests that \( y \) is not a descendant of \( x \), since \( y \) is schedule ahead of \( x \). Hence, this case is not possible.

2) \( y \in D_i \)

We first note that \( x \) cannot be a descendant of \( y \) since \( x \notin D_i \), and \( x \) cannot be an ancestor of \( y \) due to the first property of Theorem 2. Hence, \( x \) and \( y \) are irrelevant.

Let \( z \) be the critical frame such that \( y \) belongs to \( T(z) \) (with possibly \( y = z \)). If \( z = \zeta_i \), then since \( x \) and \( \zeta_i \) are irrelevant, Lemma 1 indicates \( \max(T(x)) < \min(T(z)) \), so we have \( d_x < d_y \). If \( z \neq \zeta_i \), then \( z \) and \( \zeta \) are irrelevant and \( d_{\zeta_i} < d_z \); therefore Lemma 1 indicates \( d_{\zeta_i} < d_y \). Hence, in both cases, \( d_x < d_y \). This violates the second property of Theorem 2.

For the second statement, assume towards a contradiction that the statement does not hold, i.e., there exists some frame in \( N_i - D_i \), such that it is irrelevant with respect to \( \zeta_i \) and has a deadline later than \( d_{\zeta_i} \), and it is scheduled before \( I_{i+1} \). Let \( x \) be the one of such frames that has the latest scheduled transmission start time. Then, by definition \( d_{\zeta_i} < d_x \). Let \( y \) be the frame scheduled immediately after \( x \). As explained previously, Lemmas 6, 7, 8, and 9 jointly dictate that in the quasi-SIO canonical schedule all frames in \( N_i \cup \{I_{i+1}\} \) must be sent together. Hence, we have \( y \in N_i \) or \( y = I_{i+1} \).

1) \( y \notin N_i \)

In this case, since \( y \) is scheduled before \( I_{i+1} \), the first property of Theorem 2 suggests that \( y \notin D_i \). This implies that \( y \) cannot be \( \zeta_i \) or a descendant of \( \zeta_i \).

Now, suppose \( y \) is an ancestor of \( \zeta_i \). Then, \( x \) cannot be an ancestor of \( y \), since otherwise \( x \) would not be irrelevant with \( \zeta_i \). However, \( x \) cannot be a descendant of \( y \) either, due to the first property of Theorem 2. Therefore, \( x \) and \( y \) are irrelevant. Then, since \( x \) and \( y \) are scheduled next to each other, the second property of Theorem 2 suggests that \( d_x < d_y \), which implies \( \max(T(x)) < \min(T(y)) \) by Lemma 1. However, this also implies that \( d_x < d_{\zeta_i} \), which contradicts the definition of \( x \). Hence, we conclude that \( y \) cannot be an ancestor of \( \zeta_i \).

The above implies that \( y \) and \( \zeta_i \) are irrelevant. Hence, we have \( d_y < d_{\zeta_i} \), since otherwise the existence of \( y \) would contradict the definition of \( x \). This leads to two observations. First, \( d_y < d_x \), which implies that \( x \) and \( y \) cannot be irrelevant, due to the second property of Theorem 2. Second, \( y \) cannot be a descendant of \( x \). This is because \( \max(T(\zeta_i)) < \min(T(x)) \) by Lemma 1, so we would have \( d_{\zeta_i} < d_y \) if \( y \) were a descendant of \( x \).

Furthermore, by the first property of Theorem 2, \( y \) cannot be an ancestor of \( x \), since \( y \) is schedule after \( x \). Hence, this case is not possible.

2) \( y = I_{i+1} \)

In this case, Lemma 10 suggests that \( y = I_{i+1} \) is followed immediately by \( \zeta_i \). Hence, we have the subsequence \( xI_{i+1} \), with \( d_{\zeta_i} < d_x \). Furthermore, since \( x \notin N_i - D_i \), it is irrelevant with respect to \( I_{i+1} \). This violates the second property of Theorem 2.

APPENDIX VIII

PROOF OF THEOREM 3

The proof is similar to that of Theorem 3. We only need to additionally consider the case where an I-frame \( I_{i+1} \) is not included in the transmission sequence. We note that the removal of \( I_{i+1} \) induces the removal of entire blocks \( D_i \) and \( N_{i+1} \), which does not alter the canonical form.

REFERENCES

[1] ITU-T and ISO/IEC JTC 1, “Advanced video coding for generic audiovisual services,” ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC).
