Jointly Optimal Selection and Scheduling for Lossy Transmission of Dependent Frames with Delay Constraint

Saied Mehdian and Ben Liang
Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada

Abstract—**We present a jointly optimal selection and scheduling scheme for the lossy transmission of frames governed by a dependency relation and a delay constraint over a link with limited capacity. A main application for this is scalable video streaming. Our objective is to select a subset of frames and decide their transmission schedule such that the overall video quality at the receiver is maximized. The problem is solved for two of the most common classes of dependency structures for video encoding, which include as a special case the popular hierarchical dyadic structure. We formally characterize the structural properties of an optimal transmission schedule in terms of frame dependency. It is shown that regardless of the subset of frames selected for transmission, any optimal schedule has an equivalent canonical form that is a subsequence of a unique universal sequence containing all frames. The canonical form can be computed efficiently through the construction of a dependency tree. This leads to separable but jointly optimal frame selection and scheduling algorithms that have quadratic computational complexity in the number of frames. Simulation with video traces demonstrates that the optimal scheme can substantially outperform existing suboptimal alternatives.**

I. Introduction

As video streaming contributes to an ever increasing portion of the Internet traffic, efficient and adaptive transmission of video is paramount to service providers and users alike. Toward this goal, scalable video coding allows adaptation to heterogeneous access networks and user devices to achieve satisfactory viewing experience. For example, 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 popular applications such as Bluray and Youtube.

All such scalable video codecs rely on prediction between the video frames to improve coding efficiency. An important example is the Hierarchical Prediction Structure (HPS) in H.264, in which one or more frames are used as the basis to encode another frame. Such prediction structures lead to *decoding dependency* between frames in a video sequence. The implication of dependency is two-fold: *First*, the successful reception of a frame before its playback deadline does not necessarily imply the successful playback of the frame, since the frame can be decoded only when all other frames that it depends on have been decoded. *Second*, the transmitter should not necessarily drop a frame that has been delayed beyond its playback deadline, since the frame may still be useful for the receiver to decode other frames.

This dependency relation between frames, along with the delay-sensitive nature of frame playback and the need to adaptively select and drop frames for scalable transmission under limited bandwidth, creates unique challenges in the scheduling of video frames for transmission. Classical theories of deterministic scheduling with delay constrained jobs are insufficient to address these challenges [3]–[5]. In classical scheduling, 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. This is incompatible with the dependency between video frames.

Therefore, a new approach is needed to jointly optimize the selection of a subset of frames for transmission, and the schedule to transmit those frames before their display deadlines expire. Many existing studies on efficient transmission of scalable video address dependency by a simplified *flow-based* model [6]–[12], in which a video layer can be decoded only if all lower layers are already decoded. In the more realistic *frame-based* approach [13]–[19], dynamic frame dropping and retransmission are studied, but most of these studies rely on heuristics that are challenging to analyze mathematically, and none considers the utility of frames received after their playback deadline.

In this work, we present a general method for frame selection and scheduling, in order to optimize the transmission of dependent frames with delay constraint under limited link capacity. The optimization objective is to minimize the sum utility of the dropped frames, which in the application of video streaming is the loss of playback quality in the transmitted video sequence. To the best of our knowledge, a solution to this problem does not exist in the standard deterministic scheduling literature. Furthermore, in the context of video scheduling, the proposed algorithm is unique in providing a provably optimal polynomial solution to scalable transmission that accounts for popular forms of hierarchical prediction.

More specifically, the following is a summary of the main contributions in this work:

- We formally characterize the structural properties of an optimal transmission schedule in terms of frame dependency. This leads to the categorization of two general dependency structures of interest, termed Sequential Iso-

This work has been supported in part by a grant from Bell Canada and the NSERC CRD program.

978-1-4799-4852-9/14/$31.00 © 2014 IEEE

morphically Ordered (SIO) and Quasi Sequential Isomorphically Ordered (Quasi-SIO), which include as special cases many common predictive coding structures such as the H.264 AVC default *B-frame prediction* [20] and the ubiquitous *hierarchical dyadic structure* [2].
- We show that for both the SIO and Quasi-SIO structures, any given transmission sequence has an equivalent canonical form, regardless of the subset of frames selected for transmission. Furthermore, the canonical form of any transmission sequence with dropped frames is a subsequence of a unique universal transmission sequence that contains all frames. We present a method to compute the canonical form with linear complexity in the number of frames.
- The above allows a two-step approach for jointly optimal frame selection and scheduling, where the optimal transmission order can be determined independently from frame selection. We then develop dynamic programming solutions to optimize frame selection and scheduling, which have only quadratic complexity in the number of frames for video streaming.
- Finally, simulation experiments with common test video traces show that the proposed method can substantially improve the quality of lossy streaming over a link with limited capacity, in comparison with suboptimal alternatives.

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, respectively, optimal algorithms for scheduling in the SIO and Quasi-SIO cases are presented. In Section VII, simulation experiments with video traces demonstrate the performance advantage of the proposed solution. Finally, Section VIII concludes the paper.

II. Related Work

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

A. Classical Deterministic Scheduling

The literature on deterministic scheduling is rich and diverse. Deterministic scheduling considers scenarios where a set of jobs are to be processed through machines. As opposed to stochastic scheduling, all system data is known in advance. In general, each job is associated with a size, release time, and a deadline. No machine can process more than one job a time, but the processing can be preemptive or non-preemptive. Surveys such as [3]–[5] have well summarized the existing works.

In deterministic scheduling, various objective functions have been considered for optimization, but the most relevant to our work is the weighted sum value of feasible jobs [21]–[25]. The authors of [21]–[24] are concerned with scheduling independent jobs, whereas in our scheduling problem there is a dependency relation among the frames. Furthermore, preemption is not considered in studies such as [21] and [22], while for our problem, through Theorem 1, we will show that there is no benefit in preemption. In addition, jobs have different release times in studies such as [23] and [24], while in our problem all frames are initially available so release time is not a concern.

The precedence relation is considered in [25], i.e., some jobs must precede other jobs in the schedule. Furthermore, two modes of job processing are allowed, which, corresponding to our scheduling problem, can be mapped to the special cases of transmitting and dropping a frame. However, in our problem we do not explicitly impose precedence constraints on the transmission schedule, as long as the display deadline of each frame is met. More specifically, the frames in our scheduling problem depend on each other for the purpose of decoding, but it is possible to transmit a frame prior to another frame that it depends on; it will simply be stored at the receiver for later decoding. Interestingly, through Theorem 2, we will show that in our problem there exists an optimal schedule that has a certain precedence structure. Even so, since the job scheduling in [25] does not consider the effect of dropping a job on the utility of the other jobs, it is not applicable to our work. To the best of our knowledge, there is no known optimal algorithm with polynomial complexity for scheduling with dependency and delay constraint.

B. Scheduling for Scalable Video Streaming

Many prior works on scalable video streaming do not consider the dependency relation in video transmission [26]–[34]. In all of these works the video content is transmitted in an earliest-deadline-first (EDF) fashion.

Studies that do consider the dependency relation can be divided into two main groups: flow-based and frame-based. In the flow-based approach [6]–[12], the video is modelled as a set of inter-dependent data flows. There are usually one base flow that provides minimum playback quality and multiple enhancement flows. The transmitter's choice is limited to choosing the subset of flows to schedule. Therefore, it is a simplification that does not account for inter-frame dependency.

In this work, we consider the more complicated frame-based dependency relation. In the frame-based approach [13]–[15], the video sequence is generally modelled as a set of data units with coding dependency. In all three studies, heuristics are proposed to drop data units under bad link conditions according to some pre-defined 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 complex online frame-based transmission schemes for dependent frames in the lossy environment have been studied [16]–[19]. These works target more ambitious problems than ours, since they need to account for the uncertainty in the system data, including factors such as multipath congestions and buffer overflows. Optimal solutions are generally intractable, and only heuristic solutions are available. In this work, we target a simpler offline scheduling problem that is suitable for streaming pre-recorded videos over a stable link and propose an efficient but jointly optimal frame selection and scheduling algorithm.

Finally, our proposed optimal schedule permits the transmission of frames past their display deadline, for the benefit of decoding other frames. This is not available in any of the works above. More detail on this is provided in Section III-B.

III. System Model and Optimal Transmission Sequence

In this section, we detail the system model and the problem of lossy transmission of dependent frames with delay constraint. For clarity of illustration, we focus on the application of video streaming.

A. Sequence of Frames and the Dependency Relation

We consider a sequence of frames indexed by $1, 2, \ldots, N$. Each 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 deadline, and q_l is its quality measure. In the context of video frames, d_l is the time at which frame l is to be displayed, and we assume the frames are ordered, such that $d_1 < d_2 < \cdots < d_N$. The quality measure may be the expected loss of video quality if the frame is not displayed [16]. A common measure is the peak signal-to-noise ratio of the frame.

The decoding of a frame may depend on other decoded frames. In common video codecs, 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 prescribed prediction coding. Such frame grouping is used the general Hierarchical Prediction Structure (HPS) of H.264 AVC, which enables temporal scalability [2][1]. As an example, one of the most widely adopted HPS is the *hierarchical dyadic structure*. Following the notations of [20], a hierarchical dyadic structure is denoted by $GnBm$, where n is the size of each GOP and m is the number of B-frames between consecutive I-frames or P-frames, with $m = 2^{\omega} - 1$ for some $\omega \in \mathbb{N}$. Figure 1 illustrates a GOP with $G16B3$.

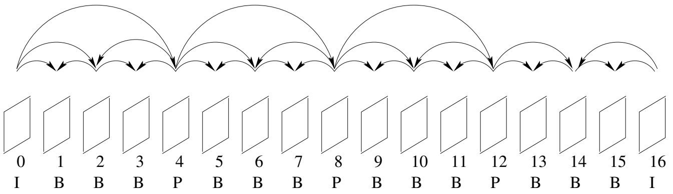


Fig. 1. Frames $(0, 1, \ldots, 15)$ form a GOP with hierarchical dyadic structure $G16B3$.

TABLE I
Table of Terminology

(S_l, d_l, q_l)	Size, deadline, and quality of frame l
Δt_l	Transmission duration of frame l
$l \not\Leftrightarrow l'$	Frames l and l' are irrelevant
$l' \rightarrow l$	Frame l depends on frame l'
GOP_i	The i-th GOP
I_i	The i-th I-frame
$\mathcal{N}_i$	The set of non I-frames in GOP_i
$\mathcal{D}_i$	The set of frames dependent on both I_i and I_{i+1}
$\mathcal{A}_i$, $\mathcal{B}_i$	The sets of frames in GOP_i that are scheduled prior and after I_{i+1}
$T(x)$	The subtree rooted at node x
$min\_dln(T)$, $max\_dln(T)$	Minimum and maximum display deadlines in rooted tree T
$c_i(x)$	The i-th child of node x
$z(\mathcal{S})$	Reward function of transmission schedule $\mathcal{S}$
transmission schedule/ sequence	The set of frames selected for transmission and their order (timing information unimportant)
decodable frame	A frame with none of its ancestors missing
successful frame	A decodable frame which arrives at the receiver prior to its deadline

The dependency between frames can be represented as a directed acyclic graph (DAG), where frames are nodes, and their dependencies are indicated by edges as shown in Figure 1. If decoding frame l requires frame l' *directly*, which is denoted by $l' \rightarrow l$, then a direct edge connects node l' to node l. In this case, l' is called a *parent* of l, and l a *child* of l'. If the decoding of frame l depends on frame l', possibly through some intermediate nodes, then l' is called an *ancestor* of l, and l a *descendant* of l'. If two frames have no ancestral relation, they are called *irrelevant* and denoted by $l \not\Leftrightarrow l'$. Throughout 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. Note that the descendants of I_{i+1} can only be in GOP_i or 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}. A partial list of the terminology is provided in Table I.

B. Quality Optimization under Limited Link Capacity

We consider a time-slotted scenario, where the transmitter sends a pre-recorded video sequence to the receiver through a link with fixed capacity C bits/timeslot. An application scenario may be the transmission of a short video clip by Youtube. We assume that lower-layer protocols in the network protocol

[1] The H.264 AVC standard is one of the most popular video compression techniques in the industry. For example, Blu-ray discs and players must support the H.264 AVC codec. In addition, this standard is widely used by video-streaming service providers such as Vimeo, Youtube, and iTuneStore, and by various HDTV broadcasting standards. The SVC extension to H.264 provides a richer set of methods to achieve scalability, but that is at the cost of significantly increased implementation complexity [2] [35]. A common practical option is to operate SVC with only a single spatial or quality layer [2].

stack ensure 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 an offset to accommodate it.

Transmission schedule. 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 timing 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.

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

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

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 frame has at least one 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, obtaining at least the same successful frames and therefore the same reward.

Proof: See Appendix I in [36]. ■

Therefore, we consider only one-by-one transmission in the rest of this paper. 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 Δt_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 maximizing the reward. Neither is there a penalty in leaving a gap, as long as the gap does not lead to out-of-date transmissions. 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" is synonymous with its "transmission sequence" without the timing information.

The decision 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 dependency structures that are of interest. We 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 Trees

We adopt a version of the Breadth First Search (BFS) algorithm [37] 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 pseudocode for MBFS is omitted to conserve space.

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 are a subset of the children of that same node in terms of the DAG G, as defined in Section III. 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 I_i 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.

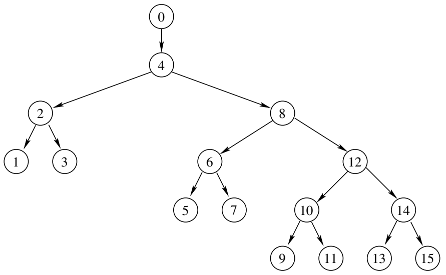


Fig. 2. MBFS on DAG in Figure 1 with node 16 removed.

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 dependency structures of interest.

Definition 1: The DAG representing the dependency structure 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 $min\_dln(T(x))$ and $max\_dln(T(x))$ be the minimum and maximum display deadlines among the nodes in $T(x)$, respectively. Let $c_i(x)$ be the ith 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

$$max\_dln(T(c_i(x))) \leq min\_dln(T(c_{i+1}(x))), 1 \leq i \leq k-1 \, . \tag{2}$$

The above definition resembles that of B-trees [37], 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 dependency structures whose MBFS forest contain only isomorphically ordered trees. The following lemma [36] 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 $max\_dln(T(x)) < min\_dln(T(y))$.

From the two definitions above, we say that a dependency structure is *Sequential and Isomorphically Ordered* (SIO) if its DAG is sequential and its MBFS forest contains only isomorphically ordered trees. For example, the dependency structure 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 dependency structures that contain B-frames that depend on an I-frame from the next GOP, we say that a dependency structure is *Quasi Sequential and Isomorphically Ordered* (Quasi-SIO) if it would become SIO after such dependency was removed. For example, the dependency structure shown in Figure 1 is Quasi-SIO.

In this work, we focus on the SIO and Quasi-SIO classes of dependency structures, since many common dependency structures in practice belong to either class. For example, the zero-delay structure [2] is SIO, and the classical B-frame prediction structure used as default in MPEG 2 and H.264 AVC [20] is Quasi-SIO. Furthermore, the widely adopted hierarchical dyadic structure [2] is Quasi-SIO (see Appendix I for more details).

C. Canonical Form of Optimal Transmission Sequence

Theorem 2 below suggests that it suffices to search for an optimal transmission sequence among those that satisfy some special properties.

Theorem 2: In the SIO and Quasi-SIO classes, any transmission sequence $\mathcal{S}$ can be re-ordered into a transmission sequence $\mathcal{S}'$ with at least the same set of successful frames (i.e., $z(\mathcal{S}) \leq z(\mathcal{S}')$) 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 IV in [36]. ■

Theorem 2 will be used extensively in the rest of this paper. 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*.

Furthermore, the lemmas below [36] 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
$$d_{f_1} < d_{o_r}, 1 \leq r \leq k \tag{3}$$
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
$$d_{f_1} > d_{o_r}, 1 \leq r \leq k \tag{4}$$

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

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

$$d_{f_1} < d_{o_r}, 1 \leq r \leq k. \tag{5}$$

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

$$d_{f_1} > d_{o_r}, 1 \leq r \leq k. \tag{6}$$

V. Optimal Schedule for 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 as defined in Section IV-C. Noting that such a sequence is a subsequence of some general pattern that we term the *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 $\mathcal{W}$ given an arbitrary transmission sequence. The lemma below [36] indicates that the frames of each GOP must be transmitted together in $\mathcal{W}$. 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.

For the frames inside a GOP, the properties of isomorphically ordered MBFS trees presented in Section IV can be used to determine the order of transmissions. Lemma 5 [36] indicates that any subtree in an MBFS tree must be scheduled together 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.

In $\mathcal{W}$, consider an arbitrary node x and the subtrees $T(c_1(x))$, …, $T(c_n(x))$, where $c_1(x)$, …, $c_n(x)$ 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 $\mathcal{W}$, we should perform a *generalized* pre-order tree walk [37] on each tree of the MBFS forest, where the children of each node are visited in ascending order of display deadlines. This procedure uniquely determines $\mathcal{W}$ 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 exists.

Then, Theorem 3 below suggests that, given a link capacity limit, a transmission sequence that maximizes the playback quality among all subsequences of the SIO universal sequence is an optimal transmission sequence. This important observation will be used in the next subsection to compute an optimal schedule.

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

Proof: See Appendix VI in [36]. ■

B. Computation of Optimal Schedule

The following dynamic programming approach selects an optimal subset of frames to transmit among the SIO universal sequence, hence solving the scheduling problem for the SIO class.

First, generate the SIO universal sequence as described in the previous subsection and re-index its frames as $f_1, f_2, \ldots, f_N$. Then, define function $h(j,t)$ as the maximum reward function if frames $\{f_i | j \leq i \leq N\}$, are to be scheduled in the time interval $[t, \infty)$, assuming all their parents with indices in the range $\{f_i | 1 \leq i \leq j-1\}$ (if any) are available.

From the system model in Section III, we have

$$z^* = h(1,0) \tag{7}$$

and the boundary conditions

$$h(N+1,t) = 0, \qquad \forall t \tag{8}$$

$$h(j,t) = 0, \qquad \forall j, \quad t > d_N. \tag{9}$$

Furthermore, h adheres to the following recursive equation:

$$h(j,t) = \max \begin{cases} h(j,t+1) \\ q_{f_j} + h(j+1, t+\Delta t_{f_j}), \quad d_{f_j} - t \geq \Delta t_{f_j} \\ h(j+1, t+\Delta t_{f_j}), \quad d_{f_j} - t < \Delta t_{f_j} \\ h(\min\{k : k > j, f_k \not\Leftrightarrow f_j\}, t). \end{cases} \tag{10}$$

In the above, $h(j,t)$ is set to be the best outcome among four possible actions at time t. 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 f_j at time t, and the frame is successful, which is ensured by the condition $d_{f_j} - t \geq \Delta t_{f_j}$. In this case, we gain q_{f_j} toward the objective along with the reward for all future frames. Similarly, the third term refers to starting to transmit frame f_j at time t and the frame is not successful as indicated by the condition $d_{f_j} - t < \Delta t_{f_j}$ (although the frame's transmission may still potentially help its descendants to achieve successful transmission). The fourth term corresponds to dropping frame f_j and moving to inspect the next frame in the universal sequence that does not depend on f_j.

Since $h(j,t)$ only needs to be computed for $1 \leq j \leq N, 0 \leq t \leq d_N$, dynamic programming requires $O(Nd_N)$ processing time and $O(Nd_N)$ memory to determine $h(1,0)$ and extract the optimal policy. Furthermore, since d_N is linear in N in all practical video codecs, and creating the MBFS forest and generating the universal sequence both require only $O(N)$ processing time, we have an overall complexity of $O(N^2)$.

VI. Optimal Schedule for Quasi-SIO Class

In this section, an optimal transmission sequence for the more general Quasi-SIO class is presented, as an extension to the solution in Section 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 $\mathcal{N}_i$ since I_i is the ancestor of all of them, and similarly I_{i+1} is scheduled prior to the frames in $\mathcal{D}_i$.

Consider the following lemmas [36]:

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

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

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

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

Lemmas 6, 7, and 8 jointly indicate that the frames of $\mathcal{M}_i = \mathcal{N}_i \cup \{I_{i+1}\}$ must be sent together. Moreover, Lemma 9 indicates that $\mathcal{M}_{i+1}$ must be scheduled immediately after $\mathcal{M}_i$, for all i.

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

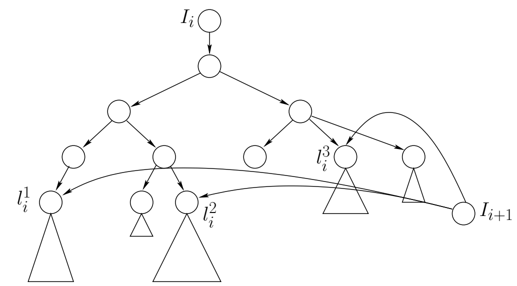


Fig. 3. Critical nodes in the GOP_i MBFS tree of the Quasi-SIO class, plus edges from I_{i+1}. Each triangle indicates a subtree rooted at the node located at its top.

It remains to determine the relative position of the frames in $\mathcal{N}_i - \mathcal{D}_i$ with respect to I_{i+1}. Let a critical node be defined as f, $f \in \mathcal{D}_i$, such that in the path between f and the root of the MBFS tree containing f, no other frame in $\mathcal{D}_i$ appears. An example of this is depicted in Figure 3, where l_i^1, l_i^2 and l_i^3 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 Γ_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 $\mathcal{D}_i$. Furthermore, by definition, the members of Γ_i are pair-wise irrelevant. Let ζ_i be the member of Γ_i with the smallest display deadline. We have the following lemmas [36]:

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

Lemma 11: In $\mathcal{V}$, consider frame $f \in \mathcal{N}_i - \mathcal{D}_i$ that is irrelevant with respect to ζ_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}.

The above observations indicate that for any Quasi-SIO transmission sequence, if we first ignore the backward prediction edges from I_{i+1} to B-frames in GOP_i, for all i, and create the SIO canonical-form transmission sequence, then the Quasi-SIO canonical-form transmission sequence 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 $\mathcal{D}_i$ in the transmission order. Furthermore, $\mathcal{V}$ 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, similarly to Section V-A, we have the following theorem, which suggests that a transmission sequence that maximizes the playback quality among all subsequences of the Quasi-SIO universal sequence is an optimal transmission sequence. This important observation will be used in the next subsection to derive an optimal schedule.

Theorem 4: For the Quasi-SIO class of dependency structures, the canonical-form of any transmission sequence is a subsequence of the Quasi-SIO universal sequence.

Proof: See Appendix VIII in [36]. ■

B. Computation of Optimal Schedule

Similarly to Section V-B, we first generate the Quasi-SIO universal sequence and re-index the frames as $f_1, f_2, \dots, f_N$. Then, define function $g(j, t, s)$, similarly to $h(j, t)$, as the maximum sum quality of successful frames if frames $\{f_i | j \leq i \leq N\}$ are to be scheduled in the time interval $[t, \infty)$ assuming all their parents with indices in the range $\{f_i | 1 \leq i \leq j-1\}$ (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 f_j, i.e., $s = (x_1(f_j), x_2(f_j))$, where $x_2(f_j) = 1$ if the nearest preceding I-frame is selected for transmission and 0 otherwise, and $x_1(f_j)$ is similarly defined concerning the second nearest preceding I-frame.

We again have

$$z^* = g(1, 0, 0), \tag{11}$$

with boundary conditions

$$g(N+1, t, s) = 0, \qquad \forall t, s \tag{12}$$

$$g(j, t, s) = 0, \qquad \forall j, s, \quad t > d_N. \tag{13}$$

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

1) If f_j is not an I-frame, let the index of the GOP that f_j belongs to be i_j. Three cases are considered:

1a) Suppose $f_j \in \mathcal{A}_{i_j}$. This is the case where f_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 the nearest preceding I-frame, I_{i_j}, is transmitted, then the outcome is similar to Section V; otherwise, f_j should not be selected. Hence, for $s = 1$ or 3, we have

$$g(j, t, s) = \max \begin{cases} g(j, t+1, s) \\ q_{f_j} + g(j+1, t+\Delta t_{f_j}, s), \quad d_{f_j} - t \geq \Delta t_{f_j} \\ g(j+1, t+\Delta t_{f_j}, s), \quad d_{f_j} - t < \Delta t_{f_j} \\ g(\min\{k : k > j, f_k \nLeftarrow f_j\}, t, s) \end{cases} \tag{14}$$

and for $s = 0$ or 2, we have

$$g(j, t, s) = \max \begin{cases} g(j, t+1, s) \\ g(\min\{k : k > j, f_k \nLeftarrow f_j\}, t, s) \end{cases} \tag{15}$$

1b) Suppose $f_j \in \mathcal{D}_{i_j}$. This is the case where f_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, f_j should not be selected. Hence, for $s = 3$, we use (14), and for $s = 0$, 1, or 2, we use (15).

1c) Suppose $f_j \in \mathcal{B}_{i_j} - \mathcal{D}_{i_j}$. This is the case where f_j does not depend on I_{i_j+1} but is located behind I_{i_j+1} in the Quasi-SIO universal sequence. If the *second* nearest preceding I-frame, I_{i_j}, is transmitted, then the outcome is similar to Section V; otherwise, f_j should not be selected. Hence, for $s = 2$ or 3, we use (14), and for $s = 0$ or 1, we use (15).

TABLE II
VIDEO TRACE SPECIFICATIONS

Sequence name	Sony_1080	NBC News
Resolution	1920×1080	352×288
FPS	30	30
Encoder	JSVM (9.15)	JSVM (9.19.14)
Encoding Type	High (Level 5)	High (Level 2.1)

2) If f_j is an I-frame, then besides scheduling the frame similarly to Section V-B, we also need to update s. It is easy to verify that, if the frame is dropped, then we update to $(2s) \bmod 4$; if the frame is transmitted, then we update to $(2s+1) \bmod 4$. Hence,

$$g(j, t, s) = \max \begin{cases} g(j, t+1, s) \\ q_{f_j} + g(j+1, t+\Delta t_{f_j}, (2s+1) \bmod 4), \\ \qquad d_{f_j} - t \geq \Delta t_{f_j} \\ g(j+1, t+\Delta t_{f_j}, (2s+1) \bmod 4), \\ \qquad d_{f_j} - t < \Delta t_{f_j} \\ g(\tilde{I}_{f_j}, t, (2s) \bmod 4), \end{cases} \tag{16}$$

where $\tilde{I}_{f_j}$ indicates the next I-frame after f_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, similarly to the SIO case, we again have an overall complexity of $O(N d_N) = O(N^2)$.

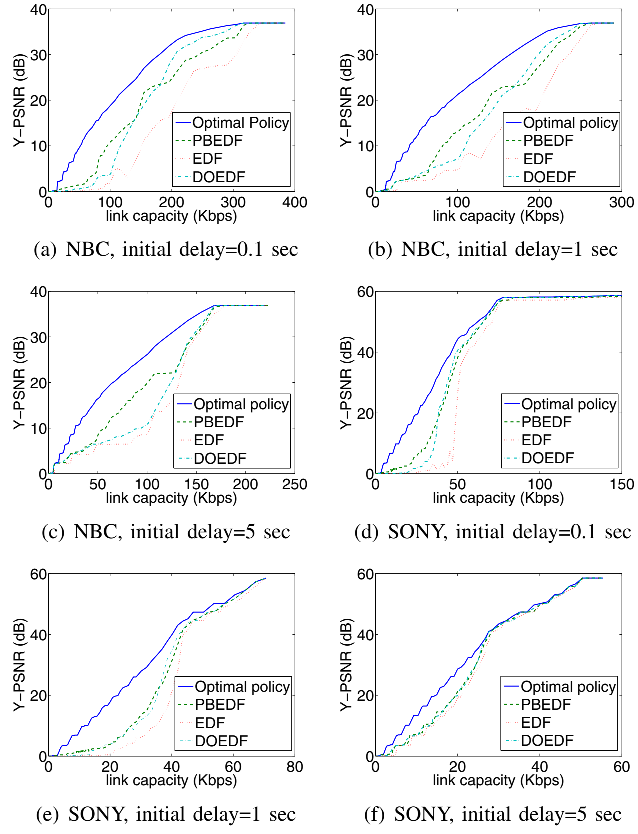
VII. EXPERIMENTS WITH VIDEO TRACES

The proposed frame selection and scheduling scheme is optimal as shown in the previous sections. In this section, we are further interested in its numeric performance when applied to H.264 video traces.

A. Methodology

The proposed scheme is studied in Matlab, using H.264 video traces provided by [20]. We present here results for the "NBC News (352 × 288)" and "SONY (1920 × 1080)" traces, representing faster moving and slower moving scenes, respectively. The details of the video traces are provided in Table II. We take the first 305 frames of each video. As an example for illustration, 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 three 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 successfully transmitted are dropped by the transmitter. EDF is the traditional scheme used for video streaming [12]. The second is a decoding order earliest-deadline-first (DOEDF) algorithm, which is similar to the first algorithm except the frames are transmitted in decoding order. The third is a priority-based earliest-deadline-first



(a) NBC, initial delay=0.1 sec

(b) NBC, initial delay=1 sec

(c) NBC, initial delay=5 sec

(d) SONY, initial delay=0.1 sec

(e) SONY, initial delay=1 sec

(f) SONY, initial delay=5 sec

Fig. 4. Comparison of algorithm performance.

(PBEDF) algorithm, which combines frame prioritization [13] [14] 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. Note that EDF and DOEDF both have $O(N)$ complexity, while PBEDF has $O(N^2)$ complexity due to the need to search over all possible values of M. We aim to study the performance gain of the proposed optimal schedule, in order to justify its tradeoff of complexity for performance over the three alternatives.

B. Experimental Results

Video playback quality is dictated by the link rate and the initial playback delay, which is defined as the time the receiver waits before video playback while receiving data. In Figure 4, 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 small, so that high PSNR can be achieved under moderate link capacity even for its high 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 impatient, and the network bandwidth is 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. Interestingly, this is not the case for EDF, DOEDF, or PBEDF, which results from the dependency structure. In particular, a slightly higher link capacity may drive these schemes to over zealously transmit a frame, which in turn reduces the time left to transmit an ancestor that appears 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. Conclusion and Discussion

A frame selection and scheduling scheme has been developed for optimal transmission of dependent and delay-constrained frames over a link with limited capacity. For two general classes of dependency structures that cover the most common video predictive coding schemes, it is shown that an optimal transmission sequence can be found as a subsequence of a canonical-form optimal universal sequence. Efficient dynamic programming solutions are then proposed to identify the optimal frame selection and scheduling with quadratic computational complexity. Experiments with video traces show that the optimal schedule can substantially improve the playback quality over suboptimal alternatives.

References

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

[2] 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, Sep. 2007.

[3] R. Graham, E. Lawler, J. Lenstra, and A. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of discrete Mathematics*, vol. 5, no. 2, pp. 287–326, 1979.

[4] E. Lawler, J. Lenstra, A. Kan, and D. Shmoys, "Sequencing and scheduling: Algorithms and complexity," *Handbooks in operations research and management science*, vol. 4, pp. 445–522, 1993.

[5] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Springer, 2012.

[6] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus priority dropping for layered video," in *Proc. ACM SIGCOMM*, 1998, pp. 131–143.

[7] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[8] J. Xu, X. Shen, J. Mark, and J. Cai, "Adaptive transmission of multi-layered video over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2305–2314, Jun. 2007.

[9] H. Zhang, Y. Zheng, M. Khojastepour, and S. Rangarajan, "Cross-layer optimization for streaming scalable video over fading wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 344–353, 2010.

[10] Q. Du and X. Zhang, "Statistical qos provisionings for wireless unicast/multicast of multi-layer video streams," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 420–433, Apr. 2010.

[11] P. Li, H. Zhang, B. Zhao, and S. Rangarajan, "Scalable video multicast with joint layer resource allocation in broadband wireless networks," in *Proc. IEEE ICNP*, Oct. 2010.

[12] T. Schierl, Y. Sanchez De La Fuente, R. Globisch, C. Hellge, and T. Wiegand, "Priority-based media delivery using SVC with RTP and HTTP streaming," *Multimedia Tools and Applications*, vol. 55, no. 2, pp. 227–246, 2011.

[13] W. Feng, M. Liu, B. Krishnaswami, and A. Prabhudev, "A priority-based technique for the delivery of stored video across best-effort networks," in *IS&T/SPIE Multimedia Computing and Networking*, 1999.

[14] J. Kritzner, U. Horn, M. Kampmann, and J. Sachs, "Priority based packet scheduling with tunable reliability for wireless streaming," in *High Speed Networks and Multimedia Communications*, ser. Lecture Notes in Computer Science, 2004, vol. 3079, pp. 707–717.

[15] G. Liebl, T. Schierl, T. Wiegand, and T. Stockhammer, "Advanced wireless multiuser video streaming using the scalable video coding extensions of H.264/MPEG4-AVC," in *Proc. IEEE ICME*, Jul. 2006, pp. 625–628.

[16] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[17] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 629–641, 2007.

[18] L. Badia, N. Baldo, M. Levorato, and M. Zorzi, "A markov framework for error control techniques based on selective retransmission in video transmission over wireless channels," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 488–500, Apr. 2010.

[19] F. Fu and M. Van Der Schaar, "A systematic framework for dynamically optimizing multi-user wireless video transmission," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 308–320, Apr. 2010.

[20] P. Seeling and M. Reisslein, "Video transport evaluation with H.264 video traces," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1142–1165, 2012.

[21] R. Karp, "Reducibility among combinatorial problems," in *Proc. Complexity of Computer Computations Symposium*, 1972.

[22] S. Sahni, "Algorithms for scheduling independent tasks," *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 116–127, 1976.

[23] E. Lawler, "A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs," *Annals of Operations Research*, vol. 26, pp. 125–133, 1990, 10.1007/BF02248588.

[24] P. Baptiste, "Polynomial time algorithms for minimizing the weighted number of late jobs on a single machine with equal processing times," *Journal of Scheduling*, vol. 2, no. 6, pp. 245–252, 1999.

[25] E. Lawler and J. Moore, "A functional equation and its application to resource allocation and sequencing problems," *Management Science*, pp. 77–84, 1969.

[26] G. Chiruvolu, R. Sankar, and N. Ranganathan, "Adaptive VBR video traffic management for higher utilization of ATM networks," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 27–40, Jul. 1998.

[27] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low-delay video streaming over error-prone channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 841–851, Jun. 2004.

[28] Z. Li, F. Zhai, and A. Katsaggelos, "Video summarization for energy efficient wireless streaming," in *Proc. SPIE*, vol. 5960, 2005.

[29] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[30] J. Huang, Z. Li, M. Chiang, and A. Katsaggelos, "Joint source adaptation and resource allocation for multi-user wireless video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 582–595, May 2008.

[31] Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, "Content-aware playout and packet scheduling for video streaming over wireless links," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 885–895, Aug. 2008.

[32] G. Liang and B. Liang, "Effect of delay and buffering on jitter-free streaming over random VBR channels," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1128–1141, Oct 2008.

[33] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *Proc. ACM SIGMETRICS*, 2008, pp. 325–336.

[34] A. Dua, C. Chan, N. Bambos, and J. Apostolopoulos, "Channel, deadline, and distortion (CD^2) aware scheduling for video streams over wireless," *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp. 1001–1011, march 2010.

[35] I. E. Richardson, *The H. 264 advanced video compression standard*, 2nd ed. Wiley, 2010.

[36] S. Mehdian and B. Liang, "Jointly optimal selection and scheduling for lossy transmission of dependent frames with delay constraint," University of Toronto, Tech. Rep. [Online]. Available: http://www.comm.utoronto.ca/∼liang/publications/techreport/depframe.pdf

[37] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. The MIT press, 2001.

APPENDIX I
HIERARCHICAL DYADIC STRUCTURE

In the hierarchical dyadic structure $GnBm$, 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 $\frac{n}{m+1} - 1$ 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 1, where $j - i$ is an integer power of 2. As an example, Figure 1 shows the hierarchical dyadic structure for a GOP with $G16B3$.

Algorithm 1 Dyadic-build$(i,\ j)$

```
if $|i - j| \leq 1$ or $\log_2 |i - j| \notin \mathbb{N}$ then
    return
end if
$i_0 \leftarrow \frac{i+j}{2}$
$i_0$ depends on $i, j$
Dyadic-build$(i,\ i_0)$
Dyadic-build$(i_0,\ j)$
```

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, we can 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 [18]. 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.

We use the following two conclusions, whose proofs can be found in [36].

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

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

The 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 SIO class, and hence, the hierarchical dyadic structure belongs to the Quasi-SIO class.