# Rate-Distortion-Authentication Optimized Streaming of Authenticated Video

Zhishou Zhang, *Student Member, IEEE*, Qibin Sun, *Member, IEEE*, Wai-Choong Wong, *Senior Member, IEEE*, John Apostolopoulos, *Senior Member, IEEE*, and Susie Wee, *Member, IEEE*

***Abstract*—We define *authenticated video* as decoded video that results from those received packets whose authenticities have been *verified*. Generic data stream authentication methods usually impose overhead and dependency among packets for verification. Therefore, the conventional rate-distortion (R-D) optimized video streaming techniques produce highly sub-optimal R-D performance for authenticated video, since they do not account for the overhead and additional dependencies for authentication. In this paper, we study this practical problem and propose an Rate-Distortion-Authentication (R-D-A) optimized streaming technique for authenticated video. Based on packets' importance in terms of both video quality and authentication dependencies, the proposed technique computes a packet transmission schedule that minimizes the expected end-to-end distortion of the authenticated video at the receiver subject to a constraint on the average transmission rate. Simulation results based on H.264 JM 10.2 and NS-2 demonstrate that our proposed R-D-A optimized streaming technique substantially outperforms both prior (authentication-unaware) R-D optimized streaming techniques and data stream authentication techniques. In particular, when the channel capacity is below the source rate, the PSNR of authenticated video quickly drops to unacceptable levels using conventional R-D optimized streaming techniques, while the proposed R-D-A Optimization technique still maintains optimized video quality. Furthermore, we examine a low-complexity version of the proposed algorithm, and also an enhanced version which accounts for the multiple deadlines associated with each packet, which is introduced by stream authentication.**

***Index Terms*—Butterfly, digital signature, R-D optimization, R-D-A optimization, stream authentication.**

## I. Introduction

VIDEO streaming applications are becoming increasingly popular and important, enabled by various video coding standards (e.g., H.264 [1]–[3]) and the rapid growth of network availability and bandwidth. This is evident in emerging commercial services like movie-on-demand, IPTV, video conference, video surveillance, and so on. However, the security issues, like confidentiality, source authentication, and secure adaptation [4], [5], are becoming serious concerns. For instance, when a video stream is transmitted over today's public and lossy Internet, the clients demand for assurance that the received video comes from the claimed source and has not been manipulated by any unauthorized third party. This paper examines the problem of streaming of authenticated video over lossy public networks.

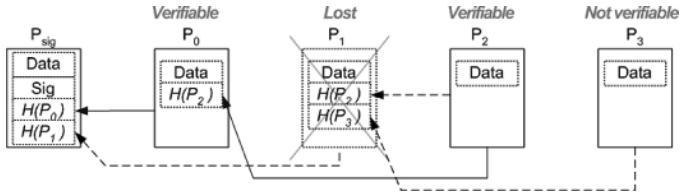


Fig. 1. Example of authentication graph.

Recent advances in media streaming include the Rate-Distortion Optimized (*RaDiO*) [6]–[8] streaming techniques and related works [9], [10], which compute a packet transmission policy that minimizes the expected end-to-end distortion at the receiver subject to a constraint on the average transmission rate. Media streams are usually assembled into packets for the convenience of network transmission, and each packet is associated with three quantities: *distortion increment*, *packet size*, and *display time*. The overall distortion will be reduced by the *distortion increment* if the packet is received before its *display time*. *Packet size* is the number of bytes in the packet. The *RaDiO* streaming techniques compute the transmission policy based on these quantities. For instance, a packet may be given more opportunities to be transmitted if it has greater *distortion increment* and smaller *packet size*, and vice versa. As a result, the performance improvement of *RaDiO* over heuristic streaming techniques like [11]–[13] is significant.

Stream authentication is applied to data packets to protect their authenticity and integrity. Common approaches of stream authentication [14]–[18] are to amortize a crypto signature among a group of packets. The packets are connected as a Directed Acyclic Graph (DAG), where a node corresponds to a packet and an edge from node $A$ to $B$ is realized by appending $A$'s hash to $B$. This approach is also referred to as graph-based stream authentication. The graph typically has one packet carrying the signature, and each node has at least one directed path to the signature packet. At the receiver, lost packets are removed from the graph, and a packet is verifiable if it has at least one path to the signature packet. A simple authentication graph with five packets is shown in Fig. 1 to illustrate the basic idea, where the signature packet $P_{\text{sig}}$ contains the signature and the hashes of $P_0$ and $P_1$, the packet $P_0$ contains the hash of $P_2$, and the packet $P_1$ contains the hashes of $P_2$ and $P_3$. Although this approach reduces authentication overhead and computational

Manuscript received May 26, 2006; revised September 11, 2006. This paper was recommended by Associate Editor X. Li.

Z. Zhang and W.-C. Wong are with the Institute for Infocomm Research (I2R), Singapore, and are also with the Department of Electronic and Computer Engineering, National University of Singapore, Singapore (e-mail: zszhang@i2r.a-star.edu.sg; lwong@i2r.a-star.edu.sg).

Q. Sun is with the Institute for Infocomm Research (I2R), Singapore (e-mail: qibin@i2r.a-star.edu.sg).

J. Apostolopoulos and S. Wee are with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: john_apostolopoulos@hp.com; susie.wee@hp.com).

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

Digital Object Identifier 10.1109/TCSVT.2006.888822

complexity, it creates authentication dependencies among the packets, i.e., some received packets may not be verified due to loss of other packets. For example, packet $P_3$ depends on $P_1$ and $P_{\rm sig}$; and packet $P_2$ depends on $P_{\rm sig}$, $P_0$ and $P_1$ for verification. If $P_1$ is lost, $P_3$ will not be verified; if $P_{\rm sig}$ is lost, all other packets will not be verified. We define the authenticated video as the video decoded from the packets that are both received and verified. Note that a received packet is discarded if it is not verified. It is worth noting that some proposed stream authentication methods have proven the optimality in terms of verification probability, i.e., it achieves the optimal verification probability given the overhead of 2 hashes per packet [18]. However, we believe that authenticating media stream still demands improved solutions because of the following two intuitive considerations. First, previous approaches assume and treat all packets as if they are of equal importance, which often is not true for media packets. For example, packets containing DC-like components are usually more important than those containing AC-like components. Similarly, packets containing P-frame coded video data are typically more important than those containing B-frame coded video data. Second, in contrast to generic data stream authentication where verification probability is deemed as the major performance measure to be optimized, for authenticated media stream the media quality of the authenticated media often is a more important factor to be optimized. As we have studied in [20], by differentiating the media packets based on their importance, we achieve significantly improved media quality at the receiver, although the verification probability may be lower than other stream authentication methods [17], [18].

When conventional *RaDiO* techniques [6]–[8] are applied to a media stream protected using graph-based authentication, they will produce highly sub-optimal performance for authenticated video, because they optimize rate and distortion only, where the "rate" includes the date rate of coded data and the "distortion" is measured by the difference between original video and decoded video for non-authenticated video. For a media stream protected with graph-based authentication, each packet is associated two more parameters: *authentication importance* and *overhead size. Authentication Importance* is the additional expected distortion increment due to the unverified packets caused by the loss of this packet, and *overhead size* is the size (in bytes) of authentication data appended to packet (including crypto signature and hashes). Conventional *RaDiO* techniques do not consider authentication and therefore are referred to as *authentication-unaware* techniques.

In this paper, we propose a Rate-Distortion-Authentication (R-D-A) Optimized streaming technique to achieve optimized quality for authenticated video. Rate-Distortion-Authentication (R-D-A) optimization is defined as rate-distortion optimization for authenticated video, where the "distortion" is measured by the difference between the original video and the authenticated video and the "rate" includes the data rate used for coded video data and the authentication overhead. The R-D-A Optimized technique is able to achieve optimized performance for authenticated video, as it accounts for the *authentication importance* and *overhead size*, in addition to the original R-D dependence and parameters.

Given a coded video with an authentication method applied, first we need to compute the quantities associated with each packet. The *distortion increment*, *packet size* and *display time* are the same as that in conventional *RaDiO* techniques [6]–[8]. The *overhead size* can be computed from the topology of the authentication graph. The *authentication importance* of a packet depends on the following factors: 1) the packet(s) whose verification is affected by this packet; 2) the *distortion increment* of the affected packets; 3) the loss probability of the affected packets; and 4) how much influence the packet has to the individual affected packets. Second, at every transmission opportunity, the R-D-A optimization process selects the best set of packet for transmission based on these parameters. For example, packets with higher importance (*distortion increment + authentication importance*) and smaller size (*packet size + overhead size*) will be assigned with more transmission opportunities. In summary, we formulate a R-D-A optimization problem to minimize the expected distortion of the authenticated video at the receiver, subject to the constraint on average transmission rate.

This paper is organized as follows. Section II gives an overview of the conventional *RaDiO* streaming techniques and existing graph-based stream authentication methods; Section III describes the proposed R-D-A Optimized streaming for authenticated video, as well as a low-complexity version of the algorithm. Furthermore, we describe how to account for multiple deadlines for authentication dependencies. Section IV illustrates how to realize the proposed R-D-A Optimization with various authentication methods. Section V validates the proposed technique with simulation results. Section VI concludes the paper.

## II. Background

This section gives a brief introduction to conventional *RaDiO* streaming techniques, assuming the streaming scenarios has sender-driven retransmission, where the receiver acknowledge every packet received. In addition, we also review existing graph-based stream authentication techniques based on crypto hash and signature, and analyze their suitability to be used with the proposed R-D-A Optimization technique.

### A. Conventional Rate-Distortion Optimized (RaDiO) Streaming Techniques

The *RaDiO* streaming techniques (e.g., [6]–[8]) assume a compressed media stream that has been assembled into packets. Each packet is associated with the *packet size* $B$, *deadline* $T$ and *distortion increment* $\Delta d$. As in the prior work, the distortion model is assumed to be additive across the lost packets. The accuracy of the additive distortion model is considered, e.g., in [19].

Suppose a packet $P_l$ has $M$ transmission opportunities before its deadline $T_l$, it is assigned with transmission policy $\pi_l$, which is an $M$-dimensional vector dictating whether or not $P_l$ will be sent at each transmission opportunity. Associated with $\pi_l$ are the cost function $\rho(\pi_l)$ and the error function $\varepsilon(\pi_l)$, where $\rho(\pi_l)$ is the expected number of transmissions for packet $P_l$ and $\varepsilon(\pi_l)$ is the probability that packet $P_l$ is not received before its deadline $T_l$. Given a group of $N$ packets, the goal is to find the optimized

policy $\Pi = [\pi_0, \pi_1, \ldots, \pi_{N-1}]$ that minimizes the Lagrange cost function

$$J(\Pi) = D(\Pi) + \lambda R(\Pi). \quad (1)$$

In (1), the Lagrange multiplier $\lambda$ controls the tradeoff between the overall expected distortion $D(\Pi)$ computed by (2) and the resulting rate $R(\Pi)$ computed by (3). A smaller Lagrange multiplier will result in the optimized transmission policy at higher rate $R(\Pi)$, and vice versa. In (2), $D_0$ represents the distortion in case that no packet is received.

$$D(\Pi) = D_0 - \sum_{l=0}^{N-1} \Delta d_l (1 - \varepsilon(\pi_l)) \quad (2)$$

$$R(\Pi) = \sum_{l=0}^{N-1} B_l \rho(\pi_l). \quad (3)$$

The optimization problem can be solved in an iterative manner. Each iteration searches for an optimal policy for only one packet, keeping the policy fixed for the rest of the packets. This is repeated until the Lagrange cost converges. For instance, at a certain iteration, the optimal policy for packet $P_l$ is given by

$$\pi_l^* = \arg\min_{\pi_l} \varepsilon(\pi_l) + \lambda_l \rho(\pi_l) \text{ where } \lambda_l = \frac{\lambda B_l}{\Delta d_l}. \quad (4)$$

The problem of finding the optimal policy $\pi_l^*$ for packet $P_l$ can be solved with dynamic programming in a Markov decision process framework [6].

Some *RaDiO* techniques, like [6] and [7], explicitly account for the decoding dependency or error concealment to compute the overall distortion. Thus, these techniques have high complexity, although they can achieve better R-D performance. A lower-complexity *RaDiO* was proposed in [8], where the distortion increment is defined to the total distortion caused by the loss of a packet, which implicitly accounts for the decoding dependency and the error concealment. It was shown in [8] that the lower-complexity *RaDiO* technique provides substantial performance improvement over non-R-D optimized streaming systems [11]–[13], and a significant fraction of the benefits provided by the high-complexity *RaDiO* streaming system [6], [7].

### *B. Existing Graph-Based Stream Authentication Methods Using Crypto Signatures*

For graph-based authentication, the main challenge is how to design a Directed Acyclic Graph (DAG) with lowest overhead, highest verification probability and lowest sender and receiver delay. However, there are tradeoffs between these performance criteria, which are summarized below.

1) *Computation complexity*: The number of hash operations and signature operations required at the sender and receiver. Note that computing a signature is much more complex than computing a hash.
2) *Overhead size*: The extra bytes introduced by stream authentication, including the hashes and signatures appended to the packets. The overhead size is determined by the number of edges in the authentication graph. Note that a signature is much bigger in size than a hash.

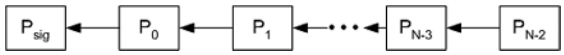


Fig. 2. Simple Hash Chain with $N$ packets.

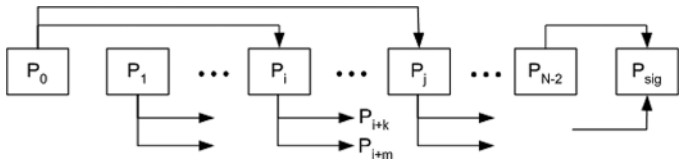


Fig. 3. Efficient Multicast Stream Signature (EMSS) with $N$ packets.

3) *Verification percentage (or verification probability)*: the percentage of verifiable packets among all the received packets. Intuitively, the more redundant paths a packet has to the signature packet, the higher the probability of being verified.
4) *Sender delay*: The delay at the sender (in number of packets) from the time when the packet is produced by the encoder to the time that all authentication data appended to this packet is ready. Real-time communication scenario requires low sender delay. For non-real-time scenario, e.g., pre-encoded content for VOD applications, it is not important because the sender has priori knowledge of all packets.
5) *Receiver delay*: The delay at the receiver (in number of packets) from the time a packet is received to the time that it can be verified. For authenticated video, each packet must be received and pass the verification before its playout deadline.

Note that the verification percentage and overhead size are two competing goals in the design of a stream authentication method. The verification percentage can be improved by increasing the number of redundant paths in the authentication graph, however this also increases the overhead size. Furthermore, the sender delay and receiver delay also compete with each other. If the signature packet is the first packet in the sequence and all edges are pointing backward, the sender delay will be high while the receiver delay will be low. On the other hand, if the signature packet is the last one in the sequence and all edges are pointing forward, the sender delay will be low while the receiver delay will be high.

We next examine five existing stream authentication techniques. The Simple Hash Chain [16] connects all packets as a single chain, as illustrated in Fig. 2. Each packet has only one edge to the previous packet, and the first packet is signed. Although it has low overhead (1 hash per packet), its verification percentage is low, because any packet loss breaks the chain and all the subsequent packets are not verifiable. The Efficient Multicast Stream Signature (EMSS) [17] extends Simple Hash Chain by adding more redundant edges in the graph. As shown in Fig. 3, each packet has its hash appended to $m$(*$m = 2$ in this example*) packets that are randomly selected from the immediately following $L$ packets (where $m < L$), and the last packet is signed. By adding more redundancy, the verification percentage is improved, but the overhead size is also increased. The Augmented Chain technique [18] is designed to combat bursty packet loss. Fig. 4 shows an example of the Augmented

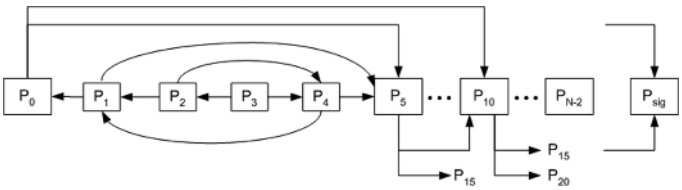


Fig. 4. Augmented Chain $C_{2,5}$.

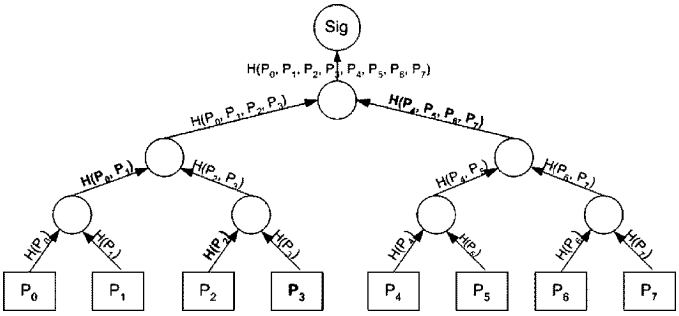


Fig. 5. Tree authentication with degree 2.

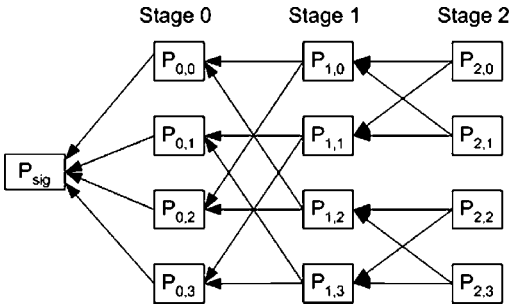


Fig. 6. Butterfly authentication graph with 13 packets.

Chain $C_{a,p}$, where $a = 2$ and $p = 5$. There are two kinds of edges, global edges and local edges. The global edges connect the packets whose indexes are multiples of $p$, while the local edges connect the packets in their own locality. It is claimed that the Augmented Chain provides optimal resistance against bursty packet loss, given an overhead size of 2 hashes per packet. The Tree-Authentication technique [15] is designed to achieve verification probability of 1 for all packets which are received, however it also has the highest overhead. As shown in Fig. 5, the packets correspond to the leaf nodes of the tree (which is of degree 2 in this example), each internal node is the concatenation of its children's hash values, and the root node is signed. Each packet has to carry the signature and the hash of its sibling nodes along the path to the root, leading to high overhead. The Butterfly Authentication technique [14] exploits the fault-tolerance of butterfly networks to improve its verification percentage. As shown in Fig. 6, the packets are connected via a butterfly graph, and the signature packet contains the hash of all the packets in the first stage. Table I summarizes the performance comparison between the different authentication schemes.

For streaming of pre-recorded video, the sender delay is unimportant because the sender is able to pre-process the packets before the streaming session actually starts. However, the high receiver delay might pose a problem because a received packet is useful only if it is verified before its deadline. In this respect, the Simple Hash Chain [16], Authentication Tree [15] and Butterfly Authentication [14] are advantageous over EMSS [17] and Augmented Chain [18].

TABLE I
COMPARISON AMONG DIFFERENT STREAM AUTHENTICATION METHODS.

| | **Simple Hash-Chain** | **Tree Authentication** | **EMSS** | **Augmented Chain** | **Butterfly Graph** |
|---|---|---|---|---|---|
| *Computation overhead* | *N,1* | *2N-1,1* | *N+1,1* | *N+1,1* | *N+1,1* |
| *Overhead Size* | *h+s/N* | *s+h*log$_2$N* | *Tunable* | *2h+s/N* | *s/N+ Kh(2log$_2$K+1)/N* |
| *Verification percentage* | *Low* | *High* | *Medium* | *Medium* | *Medium* |
| *Sender delay* | *N* | *N* | *1* | *p* | *N* |
| *Receiver delay* | *1* | *1* | *N* | *N* | *1* |
| *Possible to compute verification probability* | *Yes* | *Yes* | *No* | *No* | *Yes* |

* *Each signature is amortized among a group of N packets*
* *The hash size is h and the signature size is s.*
* *In Butterfly graph, N=K(log$_2$K+1)+1, where K is the number of packet in a stage*
* *The augmented chain graph is $C_{a,p}$*

In addition, some authentication methods enable close-form computation of the verification probability and the authentication importance from the packet loss probability, and therefore it is straightforward to use them within the proposed R-D-A Optimization framework. Such authentication methods include Simple Hash Chain [16], Tree-Authentication [15] and Butterfly Authentication [14]. However, other authentication methods like Efficient Multi-channel Stream Signature (EMSS) [17] and Augmented Chain [18] do not allow close-form computation of verification probability and authentication importance, and therefore it is more complicated to use them in the proposed framework. Nevertheless, one can still take a simulation-based empirical approach to estimate the authentication importance and the verification probability. For example, these values can be obtained through simulation and stored ahead of time, and are then read and used in real-time during the streaming session to compute the transmission schedule.

## III. RATE-DISTORTION-AUTHENTICATION OPTIMIZED VIDEO STREAMING

This section describes the proposed R-D-A optimized streaming techniques for media protected with graph-based authentication. In video authentication, a packet is decoded if and only if it is received and verified before its display time. Nevertheless, even if a packet missed its display time, it is still useful for verification of other packets that depends on this packet for verification. Therefore, each packet is actually associated with multiple deadlines: the first one is its display time, while the others are the display times of those packets that depend on this packet for verification. First, we examine the simpler case in Section III-A, where each packet is considered having a single deadline, i.e., its display time. A packet that is received and verified after its display time will be discarded. Considering that the proposed optimization algorithm has high complexity, we also propose a low-complexity version of the algorithm. Then in Section III-C, we examine the case where each packet has multiple deadlines as explained above.

### A. R-D-A Optimization With Single Deadline

In single-deadline case, each packet is associated with four parameters: *packet size* $B$, *distortion increment* $\Delta d$, *deadline* $T$, and *overhead size* $O$. The distortion of the authenticated video is reduced by $\Delta d$ if and only if the packet is received and verified before its deadline $T$.

Given that the signature packet $P_{\text{sig}}$ is received, let $\varphi_l$ be the set of packets that affects the verification probability of packet $P_l$ and $\varphi_l$ is referred to as a *determining set* of packet $P_l$. Similarly, let $\Phi_l$ be the set of packets whose verification probability is affected by packet $P_l$, and $\Phi_l$ is referred to as a *dependent set* of packet $P_l$. For example in Fig. 1, the *determining set* of $P_2$ is $\varphi_2 = \{P_0, P_1\}$ and the *dependent set* of $P_1$ is $\Phi_1 = \{P_2, P_3\}$. Note that for simplicity, $P_{\text{sig}}$ is not included in the *determining set* of any packet, but of course, all packets depend on $P_{\text{sig}}$ for verification and this is accounted for separately. We assume that for any packet $P_{l'} \in \Phi_l$, its verification probability is a linear function of the loss probability of packet $P_l$, as shown in (5), where $\alpha_l^{l'}$ and $\beta_l^{l'}$ are positive numbers. The term $\alpha_l^{l'}$ represents $P_l$'s influence on the verification probability of packet $P_{l'}$, i.e., if $P_l$ is lost, the verification probability of $P_{l'}$ will be reduced by $\alpha_l^{l'}$. Note that $P_l$ has no influence on the verification of any packet $P_{l'} \notin \Phi_l$. In Fig. 1, suppose $P_{\text{sig}}$ is received, if $P_1$ is received, $P_2$ will be verified with probability 1; Otherwise, $P_2$ depends on $P_0$ for verification, i.e., $P_2$'s verification probability is reduced to $(1-\varepsilon_0)$, where $\varepsilon_0$ is the probability of loss of packet $P_0$. Thus, the influence of $P_1$ on $P_2$'s verification probability is the difference, i.e., $\alpha_1^2 = 1-(1-\varepsilon_0) = \varepsilon_0$. Similarly, we can compute $P_1$'s influence on packet $P_3$, i.e., $\alpha_1^3 = 1$.

$$V_{l'} = -\alpha_l^{l'}\varepsilon(\pi_l) + \beta_l^{l'}. \tag{5}$$

The overhead size can be easily computed from the topology of the authentication graph. Assuming the hash size is $h$ and the signature size is $s$, the overhead size of packet $P_l$ is

$$O_l = n_l h + m_l s. \tag{6}$$

The term $n_l$ denotes the number of incoming edges to the packet $P_l$, and $m_l$ is 1 if packet $P_l$ is the signature packet and 0 otherwise.

To transmit the given $N$ packets with policy $\Pi = [\pi_{\text{sig}}, \pi_0, \ldots, \pi_{N-2}]$, the expected transmission cost is computed by summing up the transmission cost of individual packets, as shown in (7). This is similar to (3) with the only exception of the overhead size added to each packet.

$$R(\Pi) = (B_{\text{sig}} + O_{\text{sig}})\rho(\pi_{\text{sig}}) + \sum_{l=0}^{N-2}(B_l + O_l)\rho(\pi_l). \tag{7}$$

With policy $\Pi$, the expected distortion of the authenticated video is expressed in (8), where $V_l$ denotes the verification probability of packet $P_l$. The distortion of the authenticated video is computed by subtracting every packet's distortion increment weighted by its probability of being received and verified before its deadline.

$$D(\Pi) = D_0 - (1-\varepsilon(\pi_{\text{sig}}))\left(\Delta d_{\text{sig}} + \sum_{l=0}^{N-2}\Delta d_l(1-\varepsilon(\pi_l))V_l\right). \tag{8}$$

Substituting (7) and (8) into (1), we get the Lagrangian cost function

$$\begin{aligned} J(\Pi) = {} & D_0 + (-(1-\varepsilon(\pi_{\text{sig}}))\Delta d_{\text{sig}} + \lambda(B_{\text{sig}} + O_{\text{sig}})\rho(\pi_{\text{sig}})) \\ & + \sum_{l=0}^{N-2}(-(1-\varepsilon(\pi_{\text{sig}}))(1-\varepsilon(\pi_l))V_l\Delta d_l + \lambda(B_l + O_l)\rho(\pi_l)). \end{aligned} \tag{9}$$

This R-D-A optimization problem can be solved using an iterative descent algorithm, i.e., optimizing the policy for one packet at a time while keeping the other packets' policy fixed, until $J(\Pi)$ converges. For instance, the policy can be decided by (10) for $P_{\text{sig}}$ and by (11) for $P_l$:

$$\pi_{\text{sig}}^* = \arg\min_{\pi_{\text{sig}}} \varepsilon(\pi_{\text{sig}}) + \lambda'_{\text{sig}}\rho(\pi_{\text{sig}}) \tag{10}$$

$$\pi_l^* = \arg\min_{\pi_l} \varepsilon(\pi_l) + \lambda'_l\rho(\pi_l) \tag{11}$$

where $\lambda_{\text{sig}} = \lambda(B_{\text{sig}} + O_{\text{sig}})/S_{\text{sig}}$ and $\lambda'_l = \lambda(B_l + O_l)/S_l$, and where $S_{\text{sig}}$ and $S_l$ are the sensitivity factors described next. The sensitivity factors, $S_{\text{sig}}$ and $S_l$, can be obtained by taking partial derivatives of $D(\Pi)$ with respect to $\varepsilon(\pi_{\text{sig}})$ and $\varepsilon(\pi_l)$, respectively, as shown in (12) and (13):

$$S_{\text{sig}} = \Delta d_{\text{sig}} + \sum_{l=0}^{N-2}\Delta d_l(1-\varepsilon(\pi_l))V_l \tag{12}$$

$$S_l = (1-\varepsilon(\pi_{\text{sig}}))V_l\Delta d_l + (1-\varepsilon(\pi_{\text{sig}})) \cdot \sum_{P_{l'}\in\Phi_l}\alpha_l^{l'}(1-\varepsilon(\pi_{l'}))\Delta d_{l'}. \tag{13}$$

The sensitivity factor $S_l$ represents the total expected distortion increment caused by the loss of the packet $P_l$, which comprises two parts: 1) the expected distortion increment due to the unsuccessful decoding of $P_l$ (referred to as *decoding importance*), and 2) the expected distortion increment due to the reduced verification probability of all packets in its dependent set $\Phi_l$ (referred to as *authentication importance*). This is reflected in (13), where the first term is the *decoding importance* $SD_l$ in (14) and the second term is the authentication importance $SA_l$ in (15):

$$SD_l = (1-\varepsilon(\pi_{\text{sig}}))V_l\Delta d_l \tag{14}$$

$$SA_l = (1-\varepsilon(\pi_{\text{sig}}))\sum_{P_{l'}\in\Phi_l}\alpha_l^{l'}(1-\varepsilon(\pi_{l'}))\Delta d_{l'}. \tag{15}$$

In (14), the decoding importance of $P_l$ is simply the distortion increment multiplied by its verification probability and the receiving probability of the signature packet. Thus, a packet $P_l$ has higher decoding importance if one or more of the following criteria are met: 1) $P_l$ has higher distortion increment $\Delta d_l$; 2) $P_l$ has higher verification probability; and 3) $P_{\text{sig}}$ has lower loss probability.

In (15), the authentication importance of $P_l$ is the sum of the distortion increments of each packet $P_{l'} \in \Phi_1$ weighted with the receiving probability of $P_{l'}$, $P_l$'s influence on $P_{l'}$'s verification and the receiving probability of the signature packet. Thus, a packet $P_l$ has higher authentication importance if one or more of the following criteria are met: 1) there are more packets in $\Phi_l$; 2) the packets in $\Phi_l$ have higher distortion increment; 3) the packets in $\Phi_l$ have lower loss probability; 4) $P_l$ has higher influence on the packets in $\Phi_l$; and 5) $P_{\text{sig}}$ has lower loss probability.

The signature packet $P_{\text{sig}}$ is a special case in that it does not depend on any other packet for verification, and thereby its decoding importance is simply $SD_{\text{sig}} = \Delta d_{\text{sig}}$. On the other hand, all other packets depend on $P_{\text{sig}}$ for verification, i.e., $\Phi_{\text{sig}} = \{P_0, P_2, \ldots, P_{N-2}\}$. If $P_{\text{sig}}$ is lost, no packet will be verified, thereby its authentication importance is $SA_{\text{sig}} = \sum_{l=0}^{N-2} \Delta d_l (1 - \varepsilon(\pi_l)) V_l$. Therefore, the signature packet is the most important packet with the highest sensitivity factor $S_{\text{sig}}$.

From (10) and (11), the sensitivity factor, together with the size (including *packet size* plus *overhead size*), determines the transmission policy which corresponds to the bandwidth allocation among the packets. In particular, the optimization process accounts for the *distortion increment*, *packet size*, *deadline*, *authentication dependency* and *overhead size* for each packet to generate the optimized policy that minimizes the distortion of authenticated video at the receiver. In the resulting policy, a packet will have more transmission opportunities if its Lagrange multiplier is smaller, i.e., smaller size and greater sensitivity factor.

Searching for the optimized transmission policy is computationally expensive, due to the dependency imposed by graph-based authentication. The iterative process has to run for multiple iterations before the Lagrangian value converges. For instance, the complexity of proposed method is in the order of $N_i * N * (C + 2^M)$, where $N_i$ is the number of iterations that the optimization algorithm performs before convergence (typically on the order of 2–5), and $N$ is the number of packets considered for transmission. The term $C$ is the complexity of computing the sensitivity factor for a packet, and it depends on the size of the determining set and dependent set. The term $M$ is the number of transmission opportunities in the Markov decision process [6]. The computational complexity is exacerbated by the fact that the optimization algorithm has to run at every transmission opportunity.

### B. Low-Complexity R-D-A Optimization

We propose a low-complexity algorithm for selecting the packets for transmission. At each transmission opportunity, packets are selected in the following four steps.

1) In the first step, for the I-frame packets and signature packets, their current action in the policy vector is set to "SEND" if they have never been sent before or their last transmission is more than one round-trip-time ago. Note that setting the current action to "SEND" does not necessary mean that it will be transmitted in this transmission opportunity. The steps below may change the action. The purpose of this step is to avoid the deadlock scenario where a high-importance packet *A* (like I-frame packet) depends on a low-importance packet *B* (like P-frame packet serving as the signature packet) for verification. At the very beginning when neither packet has been transmitted and their policies are initialized to "NO SEND", the sensitivity factor (to be computed in the next step) of packet *B* is very small, because *A* was not yet transmitted and its importance is not reflected in *B*'s sensitivity factor. On the other hand, the sensitivity factor of packet *A* is zero because *A* would not be verified as *B* was not yet transmitted. Thus, this forms a deadlock scenario, where both packets have small sensitivity factor and therefore neither packet has much chance to get transmitted. Note that this is not a problem in the original proposed R-D-A optimization algorithm, because it will run for many iterations starting from the initial all-"SEND" policy.
2) In step 2, based on the transmission history of the other packets and the actions set in Step 1, we can compute the sensitivity factor $S_l$ using (12) and (13), and hence the sensitivity per unit cost is $S_l/(B_l + O_l)$, which is the ratio of the packet's sensitivity factor over its total size. It accounts for the authentication dependency, overhead, and packet size.
3) In step 3, the sensitivity factor per unit cost is adjusted based on the transmission history and the remaining time before its deadline. If its deadline is less than one forward-trip-time away from the current time, the adjusted sensitivity per unit cost is set to 0, because it makes no difference to send the packet. Otherwise, the sensitivity per unit cost is multiplied by a factor $\tau = (1 - \varepsilon_f)(\varepsilon_f/\varepsilon_r)^m (\varepsilon_f)^n$, where $\varepsilon_f$ and $\varepsilon_r$ are forward-trip and round-trip loss probability, $m$ is the number of previous transmissions that are more than one round-trip-time ago and $n$ is the number of previous transmissions that are less than one round-trip-time ago. That is, the sensitivity factor is discounted by $\varepsilon_f/\varepsilon_r$ for each transmission that is more than one round-trip-time ago and by $\varepsilon_f$ for each transmission that is less than one round-trip-time ago.
4) Finally, in step 4 the packets are sorted in the decreasing order of the adjusted sensitivity per unit cost. Based on this rank ordering, we choose to send the first $k$ packets whose size in total does not exceed the transmission budget.

Compared with the original proposed R-D-A optimization algorithm, this algorithm has significantly reduced complexity, as no iterations are required and it only needs to compute the sensitivity per unit cost for every packet. At each transmission opportunity, the complexity is $N * C$, where $N$ is the number of considered packets and $C$ is the complexity to compute a sensitivity factor.

### C. R-D-A Optimization With Multiple Deadlines

The multiple-deadline problem for conventional *RaDiO* has been addressed in [21], where a packet has multiple deadlines for decoding. For instance, when retroactive decoding is used, an I-frame could be still useful to decode the subsequence frames in the GOP even though it missed its own display deadline. Furthermore, the graph-based authentication also introduces the multiple-deadline problem, where an expired packet could be still useful to verify subsequent packets. In this

section, we formulate R-D-A optimization for the multiple-deadline problem caused by authentication.

As mentioned earlier, packet $P_l$ in authenticated video may have multiple deadlines; the first one is its display time, while the others are the display times of all packets in its dependent set $\Phi_l$, assuming the display times of all packets in $\Phi_l$ are later than that of $P_l$. Thus, $P_l$ has $|\Phi_l|+1$ deadlines, and each deadline is associated with a sensitivity factor. Let $S_{l,l'}$ be the $P_l$'s sensitivity factor associated with the display time of packet $P_{l'}$, where $P_{l'}=P_l$ or $P_{l'}\in\Phi_l$, $S_{l,l'}$ can be computed with the formula below:

$$S_{l,l'}=\begin{cases}(1-\varepsilon(\pi_{\text{sig}},l'))\,V_l\Delta d_l, & \text{if } l=l'\\ (1-\varepsilon(\pi_{\text{sig}},l'))\\ \quad\cdot\sum\limits_{\substack{P_{l''}\in\Phi_l\\ T_{l''}=T_{l'}}}\alpha_l^{l''}(1-\varepsilon(\pi_{l''},l'))\,\Delta d_{l''}, & \text{if } l\neq l'.\end{cases}\tag{16}$$

Therefore, each packet has multiple error probabilities, one for each deadline. For instance, the quantity $\varepsilon(\pi_l,l')$ is the probability that packet $P_l$ does not arrive by the display time of $P_{l'}$. With multiple deadlines, the expression in (11) becomes

$$\pi_l^*=\arg\min_{\pi_l}\rho(\pi_l)+\sum_{P_{l'}=P_l \text{ or } P_{l'}\in\Phi_l}\nu_{l,l'}\varepsilon(\pi_l,l').\tag{17}$$

The quantity $\nu_{l,l'}$ can be computed by $\nu_{l,l'}=S_{l,l'}/\lambda(B_l+O_l)$, which is analogous to the reciprocal of $\lambda_l'$ in (11). Note the quantity $S_{l,l'}$ is the sensitivity of overall distortion to the arrival of packet $P_l$ by deadline $T_{l'}$. However, the computational complexity is further increased by consideration of multiple deadlines due to two reasons: 1) The optimization process has to compute multiple sensitivity factors for every packet and the R-D-A optimization process has to consider that each packet has multiple error probabilities, one for each deadline. 2) The transmission window of a packet is extended to the display time of the last packet in its dependent set and the complexity of the Markov decision process exponentially increases with the length of the transmission window. This is especially true for the signature packet and early-stage packets in the authentication graph.

## IV. Rate-Distortion-Authentication Optimization With Specific Authentication Methods

This section describes how to realize the proposed R-D-A Optimization with various graph-based authentication methods. In particular, we illustrate the computation of *decoding importance*, *authentication importance*, and *overhead size*, which can be substituted into (10) and (11) to compute the optimized transmission policy.

In the previous section, we assume that for any packet $P_{l'}\in\Phi_l$, its verification probability is a linear function of the loss probability of packet $P_l$, as in (5). This is true for Simple Hash Chain [16], Tree-Authentication [15] and Butterfly Authentication [14], which can be directly incorporated into the proposed R-D-A Optimization framework. However, for authentication methods like EMSS [17] and Augmented Chain [18], it is more difficult to compute the verification dependency and authentication importance in closed form. Nevertheless, one can still take a simulation-based empirical approach to estimate the verification probability and authentication importance.

### A. R-D-A Optimization With Tree-Authentication

With Tree-Authentication [15], each packet carries the signature and the hashes of its sibling node along its paths to the root. Thus, each packet is individually verifiable, i.e., there is no authentication dependency among the packets. As such, for a packet $P_l$, its *authentication importance* is 0 ($SA_l=0$), its *decoding importance* is simply the associated distortion increment ($SD_l=\Delta d_l$), and its overhead size is $s+h*\log_2 N$, where $N$ the number of packets. Therefore, in this case, the R-D-A Optimization is very similar to conventional *RaDiO* with the only exception of authentication overhead.

Note that Tree-Authentication does not impose any authentication dependency, resulting in very low complexity. In this case, the complexity is the same as lower-complexity *RaDiO* using $DC^0$ [8]. At each transmission opportunity, we simply sort the packets in decreasing order of $\Delta d_l/(B_l+O_l)$ and choose to send the first $k$ packets whose size in total does not exceed the transmission budget.

### B. R-D-A Optimization With Simple Hash Chain

Given $N$ packets connected as a hash chain as shown in Fig. 2, a packet $P_l$ can be verified if and only if all preceding packets are received. The *determining set* of $P_l$ is $\varphi_l=\{P_{l'}|0\le l'<l\}$, the *dependent set* of $P_l$ is $\Phi_l=\{P_{l'}|l<l'\le N-2\}$, the verification probability of $P_l$ is $V_l=\sum_{l'=0}^{l-1}(1-\varepsilon(\pi_{l'}))$. Therefore, the decoding importance of $P_l$ can be computed using (14). Furthermore, for any packet $P_{l'}\in\Phi_l$, its verification probability $V_{l'}$ is a linear function of $\varepsilon(\pi_l)$, as shown in (18):

$$\begin{aligned}V_{l'}&=-\alpha_l^{l'}\varepsilon(\pi_l)+\beta_l^{l'}\ \text{ where}\\ \alpha_l^{l'}&=\beta_l^{l'}=\sum_{\substack{0\le l''<l'\\ l''\neq l}}(1-\alpha(\pi_{l''})).\end{aligned}\tag{18}$$

Substituting (18) into (15), we can obtain the authentication importance. The overhead size is $s+h$ for the signature packet $P_{\text{sig}}$, 0 for the last packet $P_{N-2}$, and $h$ for the rest.

Therefore, the decoding importance, authentication importance and overhead size can be substituted into (10) and (11) to compute the optimized transmission policy.

### C. R-D-A Optimization With Butterfly Authentication

Given $N$ packets (where $N=K(\log_2 K+1)+1$) connected as a Butterfly Authentication graph, the first packet is the signature packet, the rest of the packets are divided into $\log_2 K+1$ stages and each stage has $K$ packets. Fig. 6 gives an example of a Butterfly Authentication graph with 13 packets. Let us denote packet $P_l$ as $P_{s,j}$, where $s$ is the stage number, $j$ is the packet index within a stage, and $l=s*K+j$. The signature packet $P_{\text{sig}}$ contains the hashes of all packet in stage 0, packet $P_{s,j}$ has its hash appended to $P_{s-1,j}$ and $P_{s-1,k}$, where $k$ and $j$ are $\log_2 K$-bit numbers differing only at the $(s-1)$th most significant bit. $P_{s,j}$ is verifiable if either $P_{s-1,j}$ or $P_{s-1,k}$ is received and verified. Thus, its verification probability $V_{s,j}$ can be

expressed using the loss probability and verification probability of connected packets in previous stage, as in (19).

$$V_{s,j}=\begin{cases}\left(\begin{array}{l}\left(1-\varepsilon\left(\pi_{s-1,j}\right)\right)V_{s-1,j}+\left(1-\varepsilon\left(\pi_{s-1,k}\right)\right)V_{s-1,k}\\ -\left(1-\varepsilon\left(\pi_{s-1,j}\right)\right)V_{s-1,j}\left(1-\varepsilon\left(\pi_{s-1,k}\right)\right)V_{s-1,k}\end{array}\right),\\ \quad \text{when } 0<s\le\log_2 K\\ 1, \quad \text{when } s=0.\end{cases} \tag{19}$$

For packet $P_{s,j}$, its *determining set* $\varphi_{s,j}$ includes all packets to which $P_{s,j}$ has a directed path in the butterfly graph, i.e., $\varphi_{s,j}=\left\{P_{s',j'}|0\le s'<s, P_{s,j}\xrightarrow{path}P_{s',j'}\right\}$; its *dependent set* $\Phi_{s,j}$ includes all packets that has a path to $P_{s,j}$, i.e., $\Phi_{s,j}=\left\{P_{s',j'}\left|s<s'\le\log_2 K, P_{s',j'}\xrightarrow{path}P_{s,j}\right.\right\}$. For example, in Fig. 6, $\varphi_{1,1}=\{P_{0,1},P_{0,3}\}$ and $\Phi_{1,1}=\{P_{2,0},P_{2,1}\}$.

By iteratively applying (19), $V_{s,j}$ can be expressed using loss probabilities of all packets in its *determining set* $\varphi_{s,j}$, which can be substituted into (14) to compute the decoding importance of packet $P_{s,j}$.

*Lemma 1:* In a Butterfly Authentication graph, if packet $P_{s',j'}$ depends on packet $P_{s,j}$ for verification, i.e., $P_{s',j'}\in\Phi_{s,j}$, the dependent set of $P_{s',j'}$ is a subset of the dependent set of $P_{s,j}$, i.e., $\Phi_{s',j'}\subseteq\Phi_{s,j}$. Similarly, if the packet $P_{s,j}$ depends on $P_{s',j'}$ for verification, i.e., $P_{s',j'}\in\varphi_{s,j}$, the determining set of $P_{s',j'}$ is a subset of the determining set of $P_{s,j}$, i.e., $\varphi_{s',j'}\subseteq\varphi_{s,j}$.

*Proof:* The first statement can be proved as follows: Let $P_{s'',j''}$ be any packet in $\Phi_{s',j'}$. As $P_{s'',j''}$ has a path to $P_{s',j'}$ which in turn has a path to $P_{s,j}$, $P_{s'',j''}$ also has a path to $P_{s,j}$, i.e., $P_{s'',j''}\in\Phi_{s,j}$. Therefore, $\Phi_{s',j'}\subseteq\Phi_{s,j}$. The second statement can be proved as follows: Let $P_{s'',j''}$ be any packet in $\varphi_{s',j'}$. As $P_{s,j}$ has path to $P_{s',j'}$ which in turn has path to $P_{s'',j''}$, there is path from $P_{s,j}$ to $P_{s'',j''}$, i.e., $P_{s'',j''}\in\varphi_{s,j}$. Therefore, $\varphi_{s',j'}\subseteq\varphi_{s,j}$.

The dependency relationship in a Butterfly Authentication graph has the transition property. In short, if packet *A* depends on packet *B* which in turn depends on packet *C*, then A also depends on C.

*Lemma 2:* In a Butterfly Authentication graph, for any packet $P_{s',j'}$ in the dependent set of $P_{s,j}$, i.e., $P_{s',j'}\in\Phi_{s,j}$, its verification probability $V_{s',j'}$ can be expressed as a linear function of the loss probability of $P_{s,j}$, i.e., $V_{s',j'}=-\alpha_{s,j}^{s',j'}\varepsilon\left(\pi_{s,j}\right)+\beta_{s,j}^{s',j'}$, where $\alpha_{s,j}^{s',j'}$ and $\beta_{s,j}^{s',j'}$ are positive numbers.

*Proof:* This lemma can be proved using the induction method in two steps: First, when $s'=s+1$, for any packet $P_{s+1,j'}\in\Phi_{s,j}$, it is obvious from (19) that $V_{s+1,j'}=-\alpha_{s,j}^{s+1,j'}\varepsilon\left(\pi_{s,j}\right)+\beta_{s,j}^{s+1,j'}$ with $\alpha_{s,j}^{s+1,j'}=V_{s,j}\left(1-V_{s,k}\left(1-\varepsilon\left(\pi_{s,k}\right)\right)\right)$ and $\beta_{s,j}^{s+1,j'}=V_{s,j}+V_{s,k}\left(1-\varepsilon\left(\pi_{s,k}\right)\right)-V_{s,j}V_{s,k}\left(1-\varepsilon\left(\pi_{s,k}\right)\right)$, where $k$ and $j$ are $\log_2 K$-bit numbers that differs only at the $s$th most significant bit. Thus, the statement is true when $s'=s+1$. Second, suppose the statement is true for $P_{s',j'}\in\Phi_{s,j}$, i.e., $V_{s',j'}=-\alpha_{s,j}^{s',j'}\varepsilon\left(\pi_{s,j}\right)+\beta_{s,j}^{s',j'}$. For any packet $P_{s'+1,j''}\in\Phi_{s',j'}$, it is obvious from (19) that $V_{s'+1,j''}=aV_{s',j'}+b$, where $a$ and $b$ are positive numbers. Thus, the verification probability of $P_{s'+1,j''}$ can be written as $V_{s'+1,j''}=-\alpha_{s,j}^{s'+1,j''}\varepsilon\left(\pi_{s,j}\right)+\beta_{s,j}^{s'+1,j''}$, where $\alpha_{s,j}^{s'+1,j''}=a\alpha_{s,j}^{s',j'}$ and $\beta_{s,j}^{s'+1,j''}=a\beta_{s,j}^{s',j'}+b$. From *Lemma 1* we know that $P_{s'+1,j''}\in\Phi_{s,j}$. Therefore, this Lemma is proved.

Lemma 2 says that the authentication probability of a packet in the dependent set of packet $P_{s,j}$ can be expressed as a linear function of the loss probability of packet $P_{s,j}$. Therefore, we can compute packet $P_{s,j}$'s influence $\alpha_{s,j}^{s',j'}$ on a packet $P_{s',j'}\in\Phi_{s,j}$, which can be substituted into (15) to compute the authentication importance of $P_{s,j}$.

In a Butterfly Authentication graph with $(\log_2 K+1)+1$ packets, the signature packet contains the digital signature and hashes of all packets in stage-0, thereby the *Overhead Size* of $P_{\text{sig}}$ is $O_{\text{sig}}=s+Kh$, where $s$ and $h$ denote signature size and hash size, respectively. The packets in stage-$\log_2 K$ do not contain any hash, and the rest of the packet contains two hashes each.

$$O_{s,j}=\begin{cases}2h, & 0\le s<\log_2 K\\ 0, & s=\log_2 K.\end{cases} \tag{20}$$

Therefore, we can substitute the decoding importance, authentication importance, and overhead size into (10) and (11) to compute the optimal transmission policy.

## V. Analysis and Experimental Results

In this section, the proposed R-D-A Optimization technique is benchmarked against: 1) the *authentication-unaware RaDiO* technique coupled with graph-based authentication, and 2) straightforward streaming of video data protected by the graph-based authentication method.

### A. Experiment Setup

We consider a video streaming scenario where every received packet is acknowledged by the receiver and retransmission is driven by the sender. A packet is discarded if it is not delivered or verified before its deadline. Our experiments use the same settings as [6]–[8] for media streaming. Specifically, the network is assumed to be a packet-erasure channel, where a packet is either correctly received or lost. The packet loss and delay are random and independent in the forward and backward channel. Packet loss follows a uniform distribution (the loss rate is denoted by $e$) while packet delay follows a shifted Gamma distribution with parameters $k$ (constant delay in the network path), $n$ (number of routers in the network path), $1/\alpha$ (mean queueing delay per router), and $1/\alpha^2$ (variance of the queueing delay per router). In our experiments, the forward and backward channels are modeled as having the same loss rate and delay distribution: the loss rate $e$ is set to 0.03, 0.05, 0.1, and 0.2, as recommended by JVT for error resilience test [22], the delay parameters are set to $k=50$ ms, $n=2$ and $1/\alpha=25$ ms. The interval between two consecutive transmission opportunities is 100 ms and playout delay is $\delta=600$ ms. At any time $t$, only those packets whose deadline is in window $[t+k,t+k+\delta]$ are eligible for transmission. NS-2 [23] is used for the simulation. For *RaDiO* streaming, the Lagrange multiplier $\lambda$ is used to control the transmission rate (recall that smaller $\lambda$ results in higher transmitted bit rate) and is fixed for one streaming session.

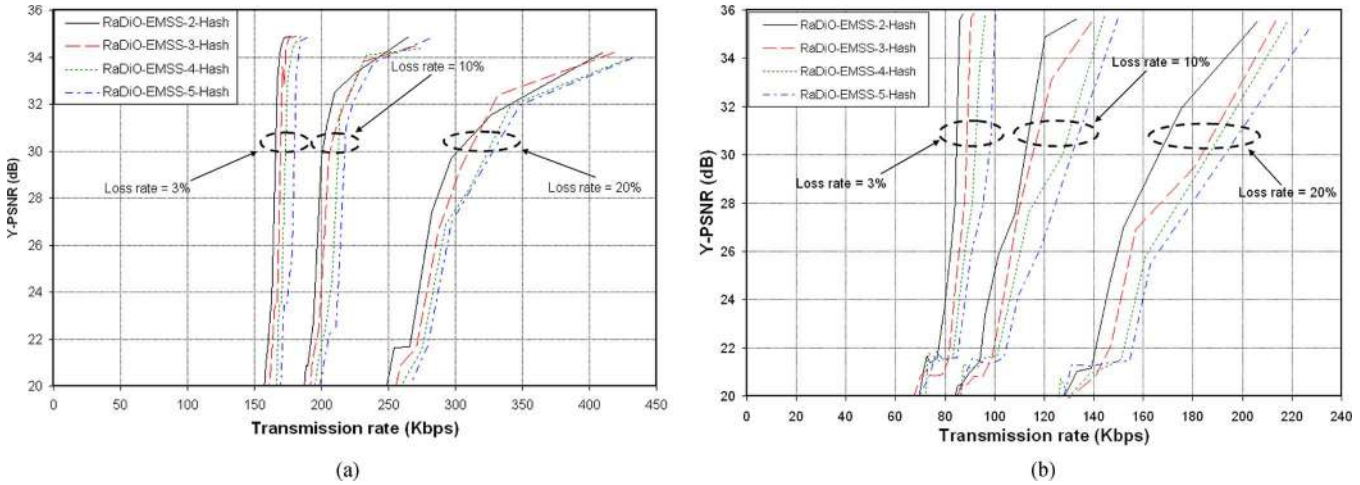


Fig. 7. Authentication-unaware *RaDiO* and EMSS authentication at different overhead sizes (2, 3, 4, and 5 hashes per packet) and different network loss rates ($e = 0.03$, 0.1, and 0.2). (a) Foreman (QCIF). (b) Container (QCIF).

Two QCIF video sequences, *Foreman* (400 frames) and *Container* (300 frames), are encoded using H.264/ACV reference software JM 10.2 [24] at around 150 kb/s and 70 kb/s, respectively. We select these two sequences in our experiment, namely *Foreman* with fast motion, and *Container* with slow motion. The frame rate is 30 frames per second and each GOP comprises of one I-frame followed by 14 P-frames. For the convenience of network transmission, each frame is divided into slices (or slice NAL units) based on the coding length. As such, an I-frame may be divided into more than one slice while a P-frame may comprise one slice only. A slice is wrapped by one RTP packet, similar to the single NAL unit mode in [25]. The parameter sets (including sequence parameter set and picture parameter set) are transmitted out-of-band. Other NAL unit types, like SEI and EOS [1], are not used here. In addition, no slice NAL unit shall exceed 1200 bytes and the network MTU is set to 1500 bytes. The space of 300 bytes is reserved for authentication overhead (signatures and hashes appended) and RTP/UDP/IP headers (around 40 bytes). Therefore, no packet segmentation is required in the network.

For authentication, we use SHA-1 Hash (160-bit) and RSA Signature (1024-bit) [26] to construct authentication graphs like EMSS [17], Augmented Chain [18], and Butterfly [14], which are all configured with their respective optimal parameters. A signature is amortized among a group of 33 consecutive packets, corresponding to around one-second of video data. As such, for an overhead size of 2 hashes per packet and frame rate of 30 frames per second, the authentication overhead constitutes around 8 kb/s of extra data rate, on top of the data rate of the original video.

In total, we implemented six systems, described as follows:

1) The first system, *Dumb-AC*, implements a straightforward transmission of video packets protected with Augmented Chain which is claimed optimal for generic data stream [18]. If there is sufficient bandwidth, the sender will retransmit the packets that has been sent but not yet acknowledged. *Dumb-AC* is the baseline system in our experiments.
2) The second system, *RaDiO*, implements authentication-unaware *RaDiO* for unauthenticated video, whose performance is used as the upper bound for all other systems. We measure the R-D performance for unauthenticated video with 1) no loss and no delay; 2) loss but no delay; and 3) loss and delay, which can demonstrate the impact of network loss and delay on the R-D performance.
3) The third system, *R-D-A-Opt-Butterfly*, implements our proposed R-D-A optimized streaming and Butterfly Authentication. We choose Butterfly Authentication because Simple Hash Chain [16] is not robust against packet loss, while Tree-Authentication [15] has too high authentication overhead. The performance of *R-D-A-Opt-Butterfly* is used to validate our proposed R-D-A Optimization technique.
4) The fourth system, *RaDiO-Butterfly*, implements authentication-unaware *RaDiO* and Butterfly Authentication. This system is used to benchmark the R-D-A Optimization technique.
5) The fifth system, *RaDiO-EMSS*, implements authentication-unaware *RaDiO* and EMSS.
6) The sixth system, *RaDiO-AC*, implements authentication-unaware *RaDiO* and Augmented Chain.

These last two systems are used to demonstrate that the proposed R-D-A Optimization outperforms authentication-unaware *RaDiO* not only for Butterfly Authentication but also for other authentication methods, out of which Augmented Chain is claimed to be optimal for generic data authentication [18]. In our experiment, EMSS and Augmented Chain were slightly modified: the modified graphs have exactly the same structure as the original ones, except that the signature packet is the first packet (instead of the last one) and the original forward edges are pointing backward. This minor modification has two advantages: 1) the authentication dependency is to align with the frame prediction dependency, because the edges point backward; 2) this modification helps to reduce the receiver delay, although it also increases the sender delay. Recall that for media streams that are pre-stored at the sender, receiver delay is more critical than sender delay.

Note that Augmented Chain and Butterfly Authentication have fixed overhead size (i.e., 2 hashes per packet) while EMSS has tunable overhead size. As such, in *RaDiO-EMSS* system, we empirically determined the optimal overhead size that produces the best R-D performance. Fig. 7 gives the R-D curves of *RaDiO-EMSS* for *Foreman* and *Container* at different overhead

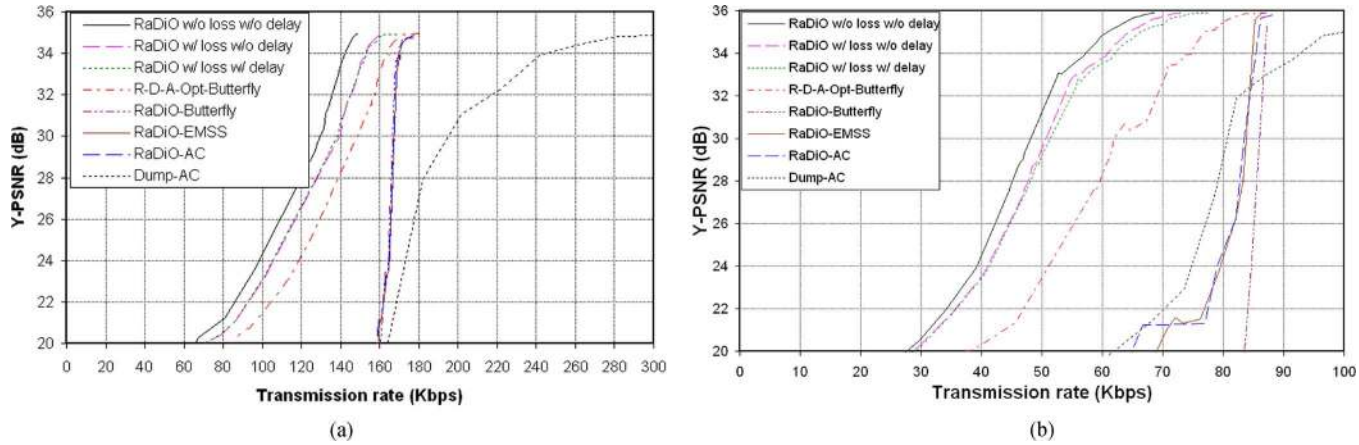


Fig. 8. R-D curves for the following systems: 1) *RaDiO* without loss and without delay; 2) *RaDiO* with 3% loss and without delay; 3) *RaDiO* with 3% loss and delay; 4) *R-D-A-Opt-Butterfly* with 3% loss and with delay; 5) *RaDiO-Butterfly* with 3% loss and with delay; 6) *RaDiO-EMSS* with 3% loss and with delay; 7) *RaDiO-AC* with 3% loss and with delay; 8) *Dumb-AC* with 3% loss and with delay. (a) Foreman (QCIF). (b) Container (QCIF).

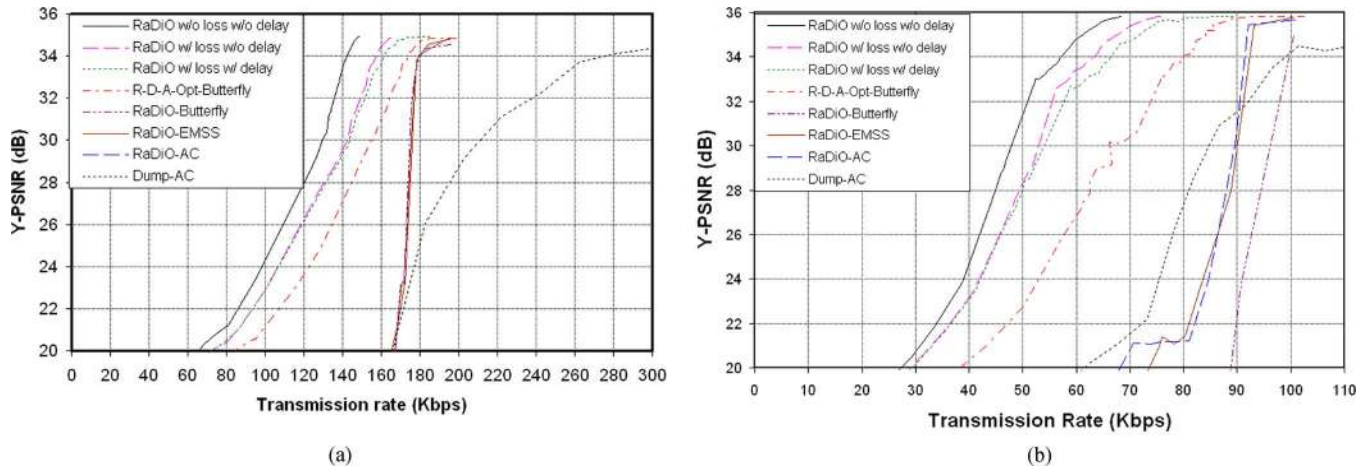


Fig. 9. R-D curves for the following systems: 1) *RaDiO* without loss and without delay; 2) *RaDiO* with 5% loss and without delay; 3) *RaDiO* with 5% loss and delay; 4) *R-D-A-Opt-Butterfly* with 5% loss and with delay; 5) *RaDiO-Butterfly* with 5% loss and with delay; 6) *RaDiO-EMSS* with 5% loss and with delay; 7) *RaDiO-AC* with 5% loss and with delay; 8) *Dumb-AC* with 5% loss and with delay. (a) Foreman (QCIF). (b) Container (QCIF).

sizes (2–5 hashes per packet) and different loss rates (0.03, 0.1, and 0.2). Note that due to space limitation we omit the set of R-D curves at loss rate 0.05, which has a similar R-D performance gap. Although the R-D curve has a less steep slope with a higher overhead size, it is shifted toward the right-hand side due to the extra overhead. Therefore, the best R-D performance is achieved when the overhead size is 2 hashes per packet. In subsequent experiments, we use the optimal overhead size of 2 hashes per packet for *RaDiO-EMSS*.

### *B. Performance Analysis of R-D-A Optimization*

The R-D performance of the six systems are given in Fig. 8 ($e = 0.03$), Fig. 9 ($e = 0.05$), Fig. 10 ($e = 0.1$), and Fig. 11 ($e = 0.2$). The authentication-unaware techniques like *RaDiO-EMSS*, *RaDiO-AC* and *RaDiO-Butterfly* do not perform well at low rates, as the Y-PSNR drops quickly to unacceptable levels due to the lack of awareness of authentication. When bandwidth is scarce, packets with smaller distortion increments will have less transmission opportunities, and thereby lead to low probability of reception. However, these packets might be very important for verifying other packets and their loss will greatly degrade the video quality. The steep slope and quick dropoff in performance for the authentication-unaware *RaDiO* techniques may be reduced by increasing the packets' verification probability, but this would require significant additional authentication overhead which would negatively impact the overall R-D performance. This is also demonstrated in Fig. 7 which shows the R-D curves of *RaDiO-EMSS* with different overhead sizes. At higher overhead size, although the performance dropoff is slightly slower, the R-D curve is shifted towards the right (lower performance).

An interesting observation is the performance difference between *Dumb-AC* and authentication-unaware *RaDiO* techniques for the different sequences *Container* and *Foreman*. For the *Foreman* video, the packets' distortion increments vary drastically among packets and therefore the authentication-unaware *RaDiO* techniques exploit this to provide better performance than *Dumb-AC*. In the *Container* video, the ship is moving slowly at constant velocity and thereby the packets' distortion increments have small variance, i.e., most packets have approximately similar distortion increments. Therefore, the authentication-unaware *RaDiO* and *Dumb-AC* techniques

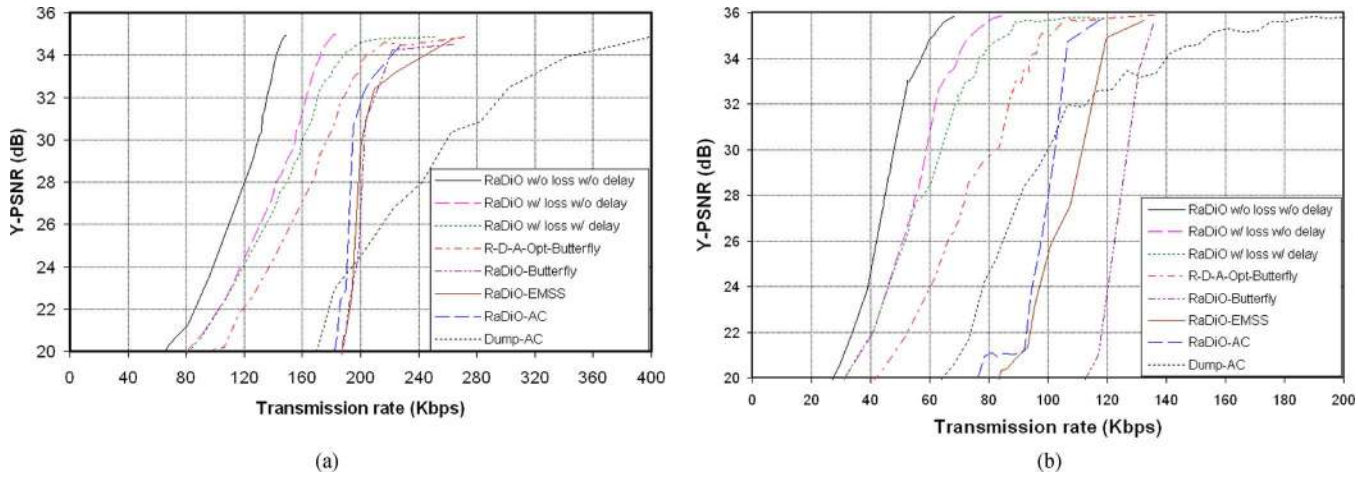


Fig. 10. R-D curves for the following systems: 1) *RaDiO* without loss and without delay; 2) *RaDiO* with 10% loss and without delay; 3) *RaDiO* with 10% loss and delay; 4) *R-D-A-Opt-Butterfly* with 10% loss and with delay; 5) *RaDiO-Butterfly* with 10% loss and with delay; 6) *RaDiO-EMSS* with 10% loss and with delay; 7) *RaDiO-AC* with 10% loss and with delay; 8) *Dumb-AC* with 10% loss and with delay. (a) Foreman (QCIF). (b) Container (QCIF).

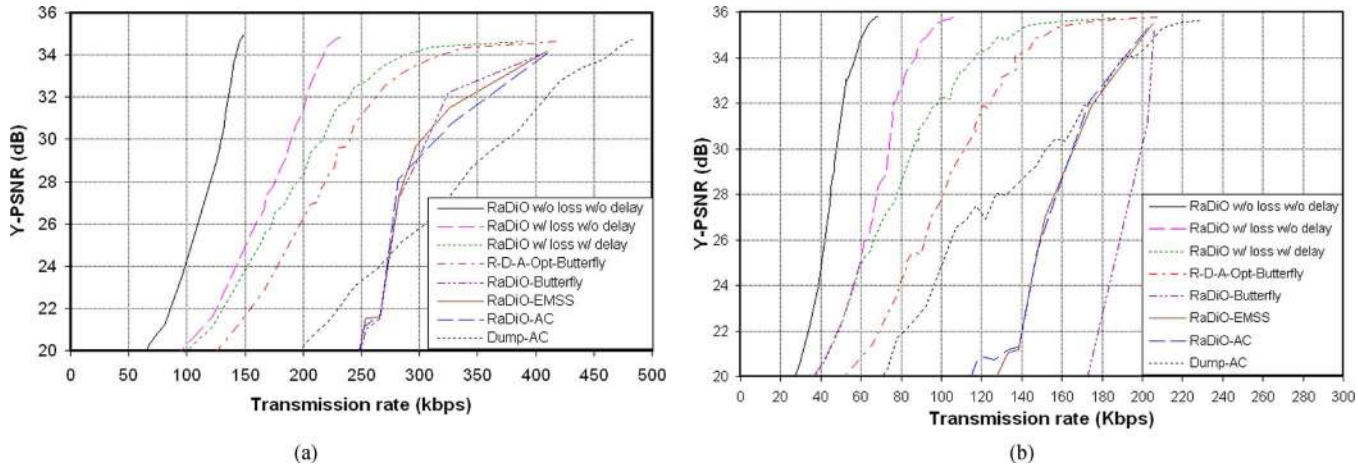


Fig. 11. R-D curves for the following systems: 1) *RaDiO* without loss and without delay; 2) *RaDiO* with 20% loss and without delay; 3) *RaDiO* with 20% loss and delay; 4) *R-D-A-Opt-Butterfly* with 20% loss and with delay; 5) *RaDiO-Butterfly* with 20% loss and with delay; 6) *RaDiO-EMSS* with 20% loss and with delay; 7) *RaDiO-AC* with 20% loss and with delay; 8) *Dumb-AC* with 20% loss and with delay. (a) Foreman (QCIF). (b) Container (QCIF).

have similar performance, and in some cases the *Dumb-AC* has slightly better performance, which is perhaps due to the randomness of the pattern of delivered packets.

As shown in Figs. 8–11, *R-D-A-Opt-Butterfly* outperforms all systems, because it computes the transmission policy based on both packets' distortion increments and authentication importance. At low bandwidth, the authentication-unaware *RaDiO* does not work anymore as its R-D curves drop quickly to unacceptable levels. Nevertheless, at the same low bandwidth the proposed R-D-A Optimization technique is still a workable solution whose R-D curves drop gracefully in parallel with the upper bound, *RaDiO* for unauthenticated video. However, we still notice that there is a performance gap between *RaDiO* and *R-D-A-Opt-Butterfly*. The possible reasons could be: 1) *R-D-A-Opt-Butterfly* has extra data rate due to the overhead size (the overhead constitutes 8 kb/s data rate on top of the data rate of the coded video); 2) the packets' authentication importance is not fully aligned with their distortion increments. If we could design an authentication graph such that packets' authentication importance and distortion increments are fully aligned, the horizontal gap between *RaDiO* and *RaDiO_Butterfly_Aware* should be reduced to the data rate of authentication overhead (i.e., 8 kb/s). However, this task constitutes future work, as there are many constraints on the graph topology (due to factors like frame prediction, playout sequence, etc.) and we cannot arbitrarily rearrange the packets in the authentication graph.

As a further observation to understand the plots, from the sender's point of view, the channel capacity is $(1-e)^2 R_{\mathrm{C}}$, where $R_{\mathrm{C}}$ is the channel bandwidth, because the sender considers a packet as successfully delivered only after the packet is acknowledged by the receiver. Therefore, to transmit all packets at source rate $R_{\mathrm{S}}$, the required bandwidth is $R_{\mathrm{S}}/(1-e)^2$. More sophisticated acknowledgement schemes can reduce this required bandwidth to close to $R_{\mathrm{S}}/(1-e)$ (depending on the playout delay), however, we keep the current approach for conceptual simplicity. When channel bandwidth drops below $R_{\mathrm{S}}/(1-e)^2$, the Y-PSNR of authenticated video starts to drop, which is validated by all R-D curves provided. For example, in Fig. 8(a), the source rate is 158 kb/s including 150 kb/s for video data and 8 kb/s for authentication overhead, so the

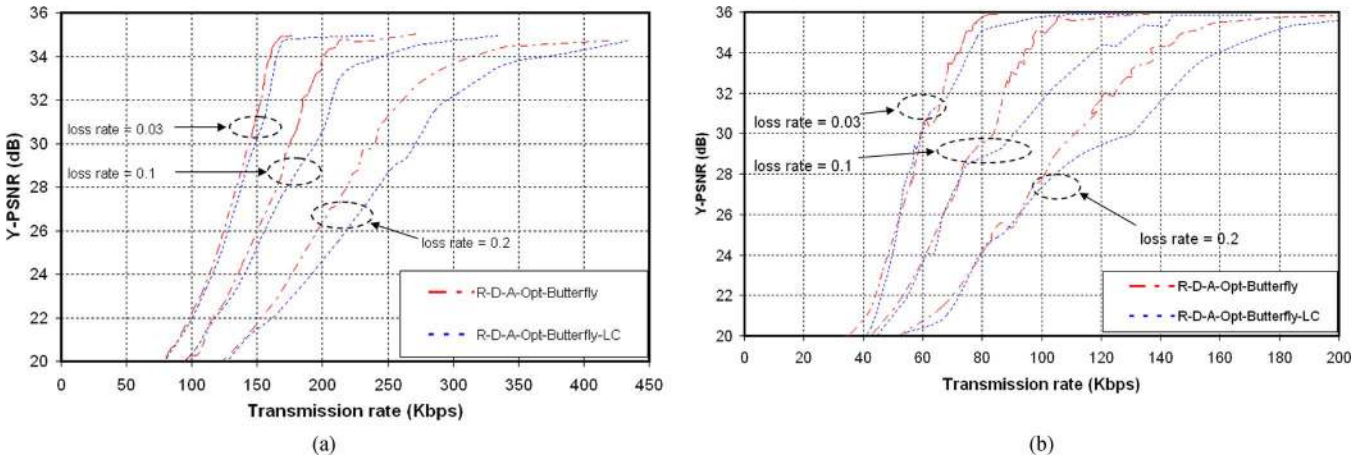


Fig. 12. R-D Curve for the following systems: 1) *R-D-A-Opt-Butterfly* at various loss rates (3%, 10%, and 20%); 2) *R-D-A-Opt-Butterfly-LC* at various rates (3%, 10%, and 20%). (a) Foreman (QCIF). (b) Container (QCIF).

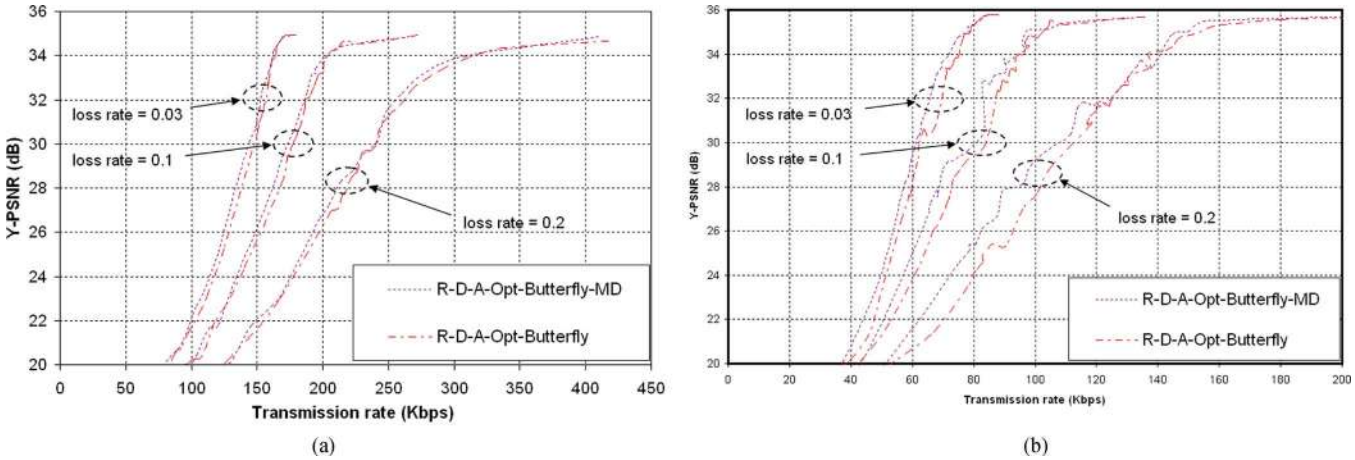


Fig. 13. R-D Curve for the following systems: 1) *R-D-A-Opt-Butterfly* at various loss rates (3%, 10%, and 20%); 2) *R-D-A-Opt-Butterfly-MD* at various rates (3%, 10%, and 20%). (a) Foreman (QCIF). (b) Container (QCIF).

knee of the R-D curve of *R-D-A-Opt-Butterfly* is located at $158/(1-0.03)^2 = 168$ kb/s when loss rate is 0.03. Similarly, when the loss rate is 0.05, 0.1, and 0.2, the knee of the R-D curve is 175 kb/s in Fig. 9(a), 195 kb/s in Fig. 10(a), and 246 kb/s in Fig. 11(a), respectively. Similar observations exist for the *Container* sequence in Figs. 8(b), 9(b), 10(b), and 11(b).

### *C. Performance Analysis of Low-Complexity R-D-A Optimization*

We also implement the low-complexity streaming of authentication video, as described in Section III-A. Fig. 12 compares the performance of optimized streaming (*R-D-A-Opt-Butterfly*) and low-complexity (*R-D-A-Opt-Butterfly-LC*) streaming, both using butterfly authentication. Note that we omit the R-D curve at loss rate 0.05 due to the space limitation, however it has similar trends with other loss rates. At lower bandwidth, the low-complexity algorithm has R-D curve close to the optimized algorithm, because it is able to identify those packets with substantially higher importance. However, the performance gap increases with the bandwidth, because the low-complexity algorithm has limited capability to differentiate among the low-importance packets which do not vary much in their associated importance. In addition, the performance gap also increases with the packet loss rates. At lower loss rate, there is little uncertainty of packet delivery and, hence, the low-complexity algorithm achieves R-D performance close to the optimized algorithm. At higher loss rates, there is higher uncertainty and the low-complexity algorithm is therefore unable to handle the situation, leading to poorer performance.

### *D. Performance Analysis of R-D-A Optimization With Multiple Deadlines*

We implement the rate-distortion optimized streaming of authentication video with consideration of multiple deadlines, as described in Section III-C. As mentioned earlier, the exponentially increased complexity is not tractable in multiple-deadline optimization, and we have to reduce it to run the simulation. We take a packet as a single-deadline packet $k_f$ seconds before its display time, where $k_f$ is the minimum forward delay. After that time, the packet is considered as multiple-deadline packet, where each deadline corresponds to the display time of a packet in its dependent set. In this manner, the long transmission window is divided into two segments which greatly reduces the space over which the optimization algorithm has to search.

Fig. 13 compares the performance of the single-deadline optimization algorithm (*R-D-A-Opt-Butterfly*) and multiple-dead-

line optimization (*R-D-A-Opt-Butterfly-MD*). We can see that the multiple-deadline optimization provides improved R-D performance, where the gain depends on the sequence and increases with packet loss rate. It is interesting to note that the performance gain achieved by using multiple-deadline optimization is limited when using the Butterfly technique, by the improved reliability to packet erasures provided by the Butterfly authentication graph. In the butterfly authentication graph, packets have multiple paths to the signature packet and they can be verified via any of these paths. Retransmitting a packet after its missed display time increases the probability of other packets' verification. However, this benefit is somewhat limited since the packet may not be used for the verification of the packets in its dependent set, due to the existence of other paths in the Butterfly graph. The potential benefits provided by multiple-deadline optimization are likely to be higher for authentication graphs which have weaker resilience to packet losses.

## VI. Conclusion

This paper has proposed an R-D-A Optimization technique for authenticated video. The simulation results demonstrate that the proposed R-D-A Optimization has substantial performance gains over all other methods tested.

In particular, our main contributions are summarized as follows:

1) By observing that an authenticated video is decoded only from packets that are both received and verified (i.e., a received packet that is not successfully verified will be discarded), we introduce a new concept for authenticated media streaming. That is, instead of optimizing the verification probability of received packets, we should optimize the quality of media decoded from those received and verified packets.
2) We further propose a R-D-A Optimized streaming technique that computes the transmission policy to minimize the expected distortion of the authenticated video at the receiver. This is achieved by accounting for the *authentication importance* and *overhead size*, in addition to the original *distortion increment* and *packet size* used in conventional authentication-unaware *RaDiO*.
3) We also show how to realize the proposed R-D-A Optimization using various authentication methods. Indeed, the proposed technique works with any authentication method as long as the packets' verification probability and authentication importance can be computed analytically or empirically.
4) We conduct simulation to compare the R-D-A Optimization and the authentication-unaware *RaDiO*. Simulation results demonstrate that the R-D-A Optimization has the best R-D performance among all systems. Indeed, the authentication-unaware *RaDiO* systems do not work at low bandwidths, as the video quality drops quickly to unacceptable levels.
5) Considering that R-D-A optimization has high complexity, we propose a low-complexity algorithm. Experimental results show that the low-complexity algorithm performs well at low bandwidth and low loss rates, compared with the optimized algorithm.
6) We also show how to account for the multiple deadlines provided by the authentication graph, and evaluate the performance improvement of multiple-deadline versus single-deadline optimization for the proposed R-D-A Optimized streaming technique.

It is worth noting that the horizontal gap between the proposed R-D-A Optimization and *RaDiO* for unauthenticated video is still greater than the data rate due to the authentication overhead. This gap could be further reduced by jointly designing the authentication graph (allocating authentication overhead) and scheduling packet transmissions, which is our future work. In [20], we formulate the distortion-overhead optimization problem to allocate overhead among the packets for a given transmission policy, and in this paper we formulate the R-D-A optimization problem to schedule packet transmissions (i.e., to allocate bandwidth among packets) for a given authentication graph topology. We believe that jointly allocating authentication overhead and bandwidth for packet transmissions could further improve the performance.

## Acknowledgment

The authors would like to thank Y. Li (Stanford University) and K. Suehring (Fraunhofer Institute) for their help with the H.264 JM software.

## References

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264/ISO/IEC 14 496–10 AVC, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[4] J. Apostolopoulos, "Secure media streaming and secure adaptation for non-scalable video," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, pp. 1763–1766.

[5] *Proc. IEEE, Special issue on enabling security technologies for digital rights management*, vol. 92, no. 6, Jun. 2004.

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

[7] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. Data Compression Conf. (DCC '03)*, 2003, pp. 203–203.

[8] J. Chakareski, J. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1257–1269, Oct. 2005.

[9] Z. Miao and A. Ortega, "Optimal scheduling for streaming of scalable media," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2000.

[10] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for Layered Streaming Media," Univ. California, Comput. Sci. Div., Berkeley, CA, Tech. Rep. UCB/CSD-98-1024, 1998.

[11] C. Papadopoulos and G. M. Parulkar, "Retransmission-based error control for continuous media applications," in *Proc. NOSSDAV*, Apr. 1996, pp. 5–12.

[12] M. T. Lucas, B. J. Dempseyt, and A. C. Weaver, "MESH: Distributed error recovery for multimedia streams in wide-area multicast networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 1997, vol. 2, pp. 1127–1132.

[13] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable internet video using MPEG-4," *Signal Process.: Image Communication*, vol. 15, pp. 95–126, 1999.

[14] Z. Zhang, Q. Sun, and W. C. Wong, "A proposal of butterfly-graph based stream authentication over lossy networks," in *Proc. IEEE Int. Conf. Multimedia and Expo. (ICME 2005)*, Jul. 2005, 4 pp.

[15] C. K. Wong and S. Lam, "Digital signature for flows and multicasts," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 502–513, Aug. 1999.
[16] R. Gennaro and P. Rohatgi, "How to sign digital streams," in *Proc. Advances in Cryptology (CRYPTO'97)*, 1997, pp. 180–197.
[17] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Security and Privacy*, 2000, pp. 56–73.
[18] P. Golle and N. Modadugu, "Authenticating streamed data in the presence of random packet loss," in *Proc. ISOC Network and Distributed System Security Symp.*, 2001, pp. 13–22.
[19] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst-length matter?," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 684–687.
[20] Z. Zhang, Q. Sun, W. C. Wong, J. Apostolopoulos, and S. Wee, "A content-aware stream authentication scheme optimized for distortion and overhead," in *Proc. IEEE Int. Conf. Multimedia & Expo. (ICME 2006)*, Jul. 2006, pp. 541–544.
[21] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," in *Proc. IEEE Int. Conf. Image Process. (ICIP 2003)*, Barcelona, Spain, Sep. 2003, pp. 661–664.
[22] Y.-K. Wang, S. Wenger, and M. M. Hannuksela, "Common Conditions for SVC Error Resilience Testing," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 JVT-P206d0, 2005.
[23] The Network Simulator (ns-2). [Online]. Available: http://www.isi.edu/nsnam/ns/
[24] H.264/AVC Reference Software, JM Version 10.2. [Online]. Available: http://iphome.hhi.de/suehring/tml
[25] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," RFC 3984, 2005.
[26] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996, pp. 429–502.

**Zhishou Zhang** (S'04) received the Bachelor (Honors) degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 1998, and the Master degree from the School of Computing, National University of Singapore, Singapore, in 2002.

Since 2000, he has been a Researcher with the Institute for Infocomm Research, a national research institute in Singapore. He is also working part-time toward the Ph.D degree in the Department of Electronic and Computer Engineering, National University of Singapore. He actively participates in standardization activities of the JPEG committee and serves as co-editor for JPEG-2000 File Format Security (FFSEC). His research interests are in media security and media communication systems.

**Qibin Sun** (M'98) received the Ph.D. degree in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1997.

Since 1996, he has been with the Institute for Infocomm Research (I2R), Singapore, where he is responsible for industrial as well as academic research projects in the areas of media security, image and video analysis. He worked at Columbia University, New Hork, during 2000–2001, as a Research Scientist. He is currently leading the Media Understanding Department at the Institute for Infocomm Research, conducting research and development in media (text, audio, image, video) analysis, retrieval and security. He is also the Head of Delegates of Singapore in ISO/IEC SC29 WG1(JPEG).

Dr. Sun actively participates in professional activities including IEEE ICME, IEEE ISCAS, IEEE ICASSP and ACM MM. He serves as the member of Editorial Board of *IEEE Multimedia Magazine*, the member of Editorial Board in *LNCS Transactions on Data Hiding and Multimedia Security*, and the Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

**Wai-Choong (Lawrence) Wong** (S'76–M'77–SM'93) received the B.Sc. (Honors) and Ph.D. degrees in electronic and electrical engineering from Loughborough University, Leicestershire, U.K.

Since November 2002, he has been Executive Director of the Institute for Infocomm Research, a national research institute under the Agency for Science, Technology and Research in Singapore. He is also a Professor in the Department of Electrical and Computer Engineering, National University of Singapore (NUS). Before joining NUS in 1983, he was a Member of Technical Staff at AT&T Bell Laboratories, Crawford Hill Laboratory, Holmdel, NJ, from 1980 to 1983. He has published over 170 papers in international journals and conferences. He also co-authored the book *Source-Matched Mobile Communications* (Pentech Press, 1995). His research interests are in wireless communication systems, including ad hoc and sensor networks, and multimedia signal processing and compression.

Dr. Wong was a recipient of the IEE Marconi Premium Award in 1989, the IEEE Millenium Award in 2000, and the e-nnovator Award in 2000.

**John Apostolopoulos** (S'91–M'97–SM'06) received the B.S., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge.

He joined Hewlett-Packard Laboratories, Palo Alto, CA, in 1997 where he is currently a Principal Research Scientist and Project Manager for the Streaming Media Systems Group. He also teaches at Stanford University, Stanford, CA, where he is a Consulting Assistant Professor of electrical engineering. His research interests include improving the reliability, fidelity, scalability, and security of media communication over wired and wireless packet networks.

Dr. Apostolopoulos received a best student paper award for part of his Ph.D. thesis, the Young Investigator Award (best paper award) at VCIP 2001 for his paper on multiple description video coding and path diversity for reliable video communication over lossy packet networks, and in 2003 was named "one of the world's top 100 young (under 35) innovators in science and technology" (TR100) by *Technology Review*. He contributed to the U.S. Digital Television and JPEG-2000 Security (JPSEC) standards. He served as an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS, and currently serves as vice-chair of the IEEE Image and Multidimensional Digital Signal Processing (IMDSP) technical committee. Recently, he was also co-guest editor of a special issue of *IEEE Network* on Multimedia over Broadband Wireless Networks, and was general co-chair of VCIP'06.

**Susie Wee** (S'95–M'96) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

She is the Director of the Mobile and Media Systems Laboratory in HP Labs, Palo Alto, CA. She is responsible for research programs in multimedia, networked sensing, next-generation mobile multimedia systems, and experience design. Her lab has activities in the U.S., Japan, and England, and includes collaborations with partners around the world. Her research interests broadly embrace the design of mobile streaming media systems, secure scalable streaming methods, and efficient video delivery algorithms. In addition to her work at HP Labs, she is a Consulting Assistant Professor at Stanford University.

Dr. Wee received *Technology Review*'s Top 100 Young Investigators award in 2002, served as an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON CIRCUITS, SYSTEMS, AND VIDEO TECHNOLOGIES. She is currently a co-editor of the JPEG-2000 Security standard (JPSEC).