JOINT SERVER/PEER RECEIVER-DRIVEN RATE-DISTORTION OPTIMIZED VIDEO STREAMING USING ASYNCHRONOUS CLOCKS

*Danjue Li * Gene Cheung[+] Chen-Nee Chuah * S.J.Ben Yoo**

| *Department of ECE | [+] HP Laboratories, Japan |
| University of California, Davis | HP Japan Ltd. |

ABSTRACT

This paper proposes a joint server/peer video streaming architecture for wireless networks, where a receiver can access a video server via an access point using the *infrastructure mode* and at the same time communicate with its peers using the *ad hoc mode* of its IEEE 802.11 interface card. We introduce a joint infrastructure/peer-to-peer, receiver-driven streaming scheme, and formulate it as a combinatorial optimization problem. We decouple the problem into two steps: first selecting the sender (server or peer) by introducing asynchronous clocks, and then applying point-to-point rate-distortion optimization algorithm between a specific sender-receiver pair. Simulation results show that our joint approach has better performance than those systems with single server or with round-robin selection scheme.

1. INTRODUCTION

In recent years, there has been an increasing demand to deliver high-quality and high-bandwidth video streams over wireless networks. However, wireless networks present a number of unique challenges as compared to delivering the same multimedia content through the traditional wired networks. In particular, the end-to-end perceived video quality can fluctuate vastly due to time-varying channel fading and the higher bit error rates.

To address the above challenges, a number of researchers have employed protection-based approaches, such as forward error correction (FEC), automatic retransmission request (ARQ), and multiple description (MD) techniques. Krishnamachari et al.[1] proposed an adaptive cross-layer protection strategy to enhance the robustness and efficiency of scalable video transmission. Majumdar et al.[2] presented a hybrid FEC/ARQ scheme to increase the robustness of the video streaming in wireless LAN. While these approaches enhance the performance of video streaming over wireless networks, they do not exploit the capability of a receiver to directly communicate with its peers which are within its transmission range. In fact, IEEE 802.11b interface cards already support two communication modes: (a) *infrastructure mode*, where a receiver connects to the network via an access point (one wireless hop), and (b) *ad hoc mode*, where a receiver can communicate directly with its peers, sometimes via multiple wireless hops if ad hoc routing protocols are implemented.

The goal of this paper is to support high quality video streaming service such as video-on-demand over IEEE802.11b wireless LAN by distributing the streaming load among the video server

and neighboring peers. We propose a joint server/peer video streaming architecture to optimize the streaming performance from the perspective of a single wireless client for applications. We consider the scenario where part of the multimedia content that a receiver desires already reside on its peers that are within its transmission range. Connections to these peers typically have shorter delay and better performance than communicating (via the infrastructure) with a remote video server. However, the reliability of the ad hoc connections heavily depends on the mobility patterns of the receiver relative to its peers.

Our contributions in this paper are twofold. First, we present a novel architecture that combines the advantages of two communication modes offered by IEEE802.11b networks: *infrastructure and ad-hoc modes*. The infrastructure-based communication is used between the video server and the client while the ad-hoc mode is used between the client and its peers. We introduce a receiver-driven approach for selectively streaming video from a joint sender group, composed of nearby wireless peers and a fixed video server, to a single wireless client. Compared with sender-driven approach, our approach avoids the synchronization problem among multiple senders and can easily adapt to instantaneous changes in the network. Second, we formulate the receiver-driven streaming scheme as a combinatorial optimization problem. We decouple the problem into two steps: first selecting the sender (server or one of the peers) and then applying point-to-point rate-distortion optimization algorithm between a specific sender-receiver pair. To solve the sender selection problem, a set of asynchronous clocks are introduced at the client, and each clock is responsible for one particular sender-receiver pair. We implement and evaluate our scheme in network simulator NS-2.26 [3].

The rest of the paper is organized as the following. Section 2 describes the related work. We present the joint server/P2P video streaming architecture and model the whole system in Section 3. Section 4 presents our solution, specifically, the rate-distortion optimization of our streaming scheme. NS simulations results are presented in Section 5. We conclude the paper and discuss future research directions in Section 6.

2. RELATED WORK

To the best of the authors' knowledge, the most thorough treatment of point-to-point rate-distortion optimized streaming is [4]. Our work leverages heavily on [4] and in fact, our approach is a combined application of asynchronous clocks and [4] to path diversity streaming. Among [4]'s many extensions, [5] is the most related to our work. We do differ in two significant aspects. First, given M distinct delivery paths, each with different path characteristics, we consider M rate constraints, one for each path, while

Legend:
- - - - Layered Contents from Peers
- - - - Contents from Server via. AP

A: Destination Client
B: Peers which host the desired content
C: Infrastructure/Access Point (AP)
D: Relay Client to Peers
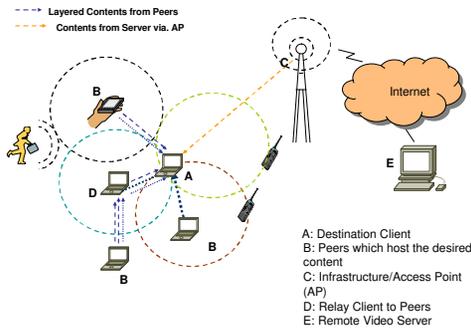E: Remote Video Server

Fig. 1. Joint Server/Peer Video Streaming over Wireless Network

[5] considered only one. Second, the optimization algorithm in [5] scaled exponentially with M, while our algorithm scales linearly.

The notion of path diversity video streaming from multiple sources has been explored in the pervious literature. [6, 7] presented algorithms to stream video from multiple servers. [8] proposed a system to tackle the quality degradation of streaming video over IEEE 802.11b wireless networks by leveraging the path diversity between multiple access points and the mobile client. However, none of them is rate-distortion optimized. [9] discussed a novel peer-to-peer (P2P) multicast streaming mechanism to stream a flash crowd over the Internet. Instead, we are concerned with optimizing streaming for a single client for applications like movie preview or video-on-demand.

3. SYSTEM OVERVIEW

In this section, we describe the proposed joint server/peer video streaming architecture in details and then model the system.

Figure 1 shows the architecture of our proposed streaming system, where the receiver and its peers can communicate with the video server through an *access point* to communicate with the video server. Meanwhile, the receiver can communicate with its peers using the ad-hoc mode of its IEEE 802.11b interface card. Each time when a client node wants to download a video file, it will send a request-for-packet to the video server or a peer within its transmission range. Based on the network feedback, the client will decide which part of video file it will request from which sender. Currently, we assume that the client has the full knowledge of the video content kept in its peers and the remote server, i.e. which part of the video is available at which sender.

3.1. Source Model

Similar to [4], we model the video source using a directed acyclic graph (DAG) $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Each frame i is represented by a data unit DU_i, and data unit is the smallest granularity we will consider in the optimization. Each edge $e_{i \to j}$ from DU_i to DU_j indicates that correct decoding of DU_j depends on the correct decoding of DU_i. Each DU_i has three associated constants: size of DU_i in RTP packets n_i, playout buffer deadline T_i, and reduction in distortion D_i. T_i is the time by which DU_i must reach the client and be decoded for DU_i to be

useful. D_i is the distortion reduction at the receiver if the DU_i is decoded timely and correctly.

3.2. Channel Model

3.2.1. Loss and Delay

For each sender-receiver pair j, we construct the following channel model for loss and delay. We first define a *round-trip packet loss* as one minus the probability that a transmission request packet is correctly received *and* the requested packet is correctly delivered. We similarly define a *round-trip packet delay* as the delay of a request packet being sent from receiver to the sender *plus* the delay of the requested packet being sent from the sender to the receiver, given neither the request packet nor the requested packet are lost. For round-trip packet loss, we use a time-invariant packet erasure model with parameter ϵ_j. For peer-to-peer connection, the loss rate is likely much higher due to unpredictability of mobile user movement. For round-trip packet delay, we use an exponential distribution $f_j(x) = \gamma_j e^{-\gamma_j x}, x \geq 0$. A request packet sent by the receiver at time T will result in a requested packet arrival at receiver by time T' with probability $p_j(T' - T)$:

$$p_j(T' - T) = (1 - \epsilon_j) \int_0^{T' - T} f_j(x) dx \qquad (1)$$

3.2.2. Bandwidth Constraints

For the server-receiver connection using IEEE802.11b infrastructure mode, since the connection traverses the Internet, it needs to be TCP-friendly. In order not to claim more bandwidth than what a normal TCP connection would use under the same network condition, the transmitted IP packets will be spaced by Δ_0 using a well-known TCP-friendly congestion control equation [10]:

$$\Delta_0 = \mu_0 \sqrt{2\alpha_0/3} + 3(\mu_0 + 4\sigma_0)\alpha_0(1 + 32\alpha_0^2)\sqrt{3\alpha_0/8} \quad (2)$$

where μ_0 and σ_0^2 are the estimated mean and variance of RTT for the link between the server-receiver pair, respectively.

For the peer-receiver connections using IEEE802.11b ad-hoc mode, each peer decides how much bandwidth it is willing to contribute as part of system initiation phase. The contributing bandwidth of each peer j is delimited by the frequency of individual request sent from client to peer j, which is $1/\Delta_j$. Peer j can only send packet at time $t_j + m\Delta_j, m \in \{1, \cdots, M-1\}$ during ad-hoc mode. We assume such a policy is used.

4. RATE-DISTORTION OPTIMIZED TRANSMISSION POLICY SELECTION

4.1. Mathematical Formulation

In this section, we formalize the joint server/peer streaming problem as a formal combinatorial optimization problem. Like [4], we will first assume the optimization is performed every P_o seconds, and at a given optimization instant t, N data units in a selected optimization window are under consideration for (re)transmission. As output of the optimization, the algorithm must decide the transmission strategy for the upcoming P_o seconds.

4.1.1. Defining the Variables

We formulate the rate-distortion optimized streaming problem as follows. For each DU_i in the optimization window, we define a *transmission request policy* π_i and a *transmission request policy vector* $\pi = [\pi_1, ..., \pi_N]$ for all N data units. Each transmission policy includes two parts, $\pi_i = \{h_i, c_i\}$, where h_i is *transmission history* of DU_i and c_i is the *transmission decision* taken during optimization at instant t. Transmission history h_i is defined by:

$$h_i = \left\{ (t_i^{(1)}, s_i^{(1)}), \ldots, (t_i^{(l_i)}, s_i^{(l_i)}) \right\} \qquad (3)$$

where l_i is the number of previous transmission request attempts, and $t_i^{(k)}$ and $s_i^{(k)}$ are the timestamp and sender ID of attempt k, respectively. Sender ID $s_i^{(k)}$ identifies the sender to which the request packet was sent at attempt k, where $s_i^{(k)} \in \{0, \ldots, M-1\}$.

Transmission decision c_i determines the transmission strategy for DU_i during the next P_o seconds. It is defined similar as h_i:

$$c_i = \left\{ (\tau_i^{(1)}, \xi_i^{(1)}), \ldots, (\tau_i^{(z_i)}, \xi_i^{(z_i)}) \right\} \qquad (4)$$

where z_i is the number of transmission requests to be sent in the next P_o seconds, and $\tau_i^{(k)}$ and $\xi_i^{(k)}$ are similarly defined quantities as $t_i^{(k)}$ and $s_i^{(k)}$ in (3).

Now we can calculate the probability $q_i(\pi_i)$ that DU_i is received correctly and on time given transmission request policy π_i:

$$q_i(\pi_i) = 1 - \prod_{k=1}^{l_i} \left(1 - p_{s_i^{(k)}}(T_i - t_i^{(k)}) \right) \prod_{k=1}^{z_i} \left(1 - p_{\xi_i^{(k)}}(T_i - \tau_i^{(k)}) \right) \qquad (5)$$

Given probability $q_i(\pi_i)$ of each DU_i and the DAG source model, we can deduce a resulting expected distortion of a group of N data units under transmission request policy vector π as:

$$D(\pi) = \left\{ D_0 - \sum_{i=1}^{N} d(\pi_i) \prod_{j \preceq i} q_j(\pi_j) \right\} \qquad (6)$$

where $j \preceq i$ denotes the set of DU_j's that preceed or equal to DU_i in DAG G.

4.1.2. M Rate Constraints

Given the optimization period P_o, we can construct the M rate constraints for the M sender-receiver pairs simply as follows. For each sender-receiver pair j, P_o yields $W_j = \lfloor P_o/\Delta_j \rfloor$ transmission opportunities until the next optimization instant $t + P_o$. Having W_j's, the rate constraints are then:

$$R_j(c_i) = \sum_{k=1}^{z_i} n_i \, \delta(\xi^{(k)} - j)$$

$$\sum_{i=1}^{N} R_j(c_i) \leq W_j \qquad j = 0, \ldots, M-1 \qquad (7)$$

where $\delta(x)$ is the delta function: $\delta(x) = 1$ if $x = 0$ and 0 otherwise. If sender-receiver pair j is selected at attempt k of DU_i, i.e. $\xi_i^{(k)} = j$, then the request time must be one of the transmission opportunities of pair j, i.e.:

$$\tau_i^{(k)} \in \{t + m\Delta_j, \, m \leq W_j\} \qquad (8)$$

The optimization problem is to find c_i's of DU_i's in policy vector π to minimize (6) under the M constraints (7) and (8).

4.2. Solution

4.2.1. Asynchronous Clocks

Solving the rate distortion optimization problem defined by (6) given (7) and (8) directly is difficult. Instead, we simplify the problem with the following trick: suppose we are permitted to vary optimization period P_o as often as we like, we can then set P_o to be *exactly* the amount of time until the next transmission opportunity of any sender-receiver pair becomes available. In one implementation, at start time t, we send no requests and set P_o as follows:

$$P_o = \min_{j=0,\ldots M-1} \{\Delta_j\} \qquad (9)$$

$$j = \arg \min_{j=0,\ldots,M-1} \{\Delta_j\} \qquad (10)$$

At $t + P_o = t + \Delta_j$, one transmission opportunity of sender-receiver pair j is immediately available, so we select exactly *one* data unit (to be discussed) to request from sender j. We then set the next optimization period P_o as:

$$P_o = \min\{\Delta_0 - \Delta_j, \ldots, \Delta_j, \ldots, \Delta_{M-1} - \Delta_j\} \qquad (11)$$

where for sender-receiver pair j we return it to Δ_j. The process then repeats, each time we select one data unit for the immediately available transmission opportunity of sender-receiver j.

The process is equivalent to setting M *asynchronous clocks*, each clock j to awake in Δ_j seconds. When a clock j goes on, we are notified of an immediately available transmission opportunity of sender-receiver pair j. We select a data unit to request from sender j, and reset the clock to awake in another Δ_j seconds.

In the above formulation, this process amounts to setting one W_j to be one and the rest zero, and setting all transmission decision lengths z_i's of DU_i's to be one. This method of staggering the optimization across time not only permits us to select sender automatically upon clock awakening, but it also allows us to solve instead the much simpler point-to-point RD optimized streaming to be discussed next.

4.2.2. Point-to-point RD Optimized Streaming

After committing to a sender j for requesting a data unit, the problem simplifies to a point-to-point RD optimized streaming problem, for which Section 5 of [4] provides a simple version of the complete solution. In summary, the optimal data unit DU_i for transmission is the one with the largest $\lambda_i = \lambda_i' S_i / B_i$, where λ_i' and S_i are defined as follows:

$$\lambda_i' = q_i(\pi_{i,1}) - q_i(\pi_{i,0}) \qquad (12)$$

$$S_i = \sum_{k \succeq i} D_k \prod_{\substack{j \preceq k \\ j \neq i}} q_j(\pi_j) \qquad (13)$$

where $\pi_{i,1} = \{h_i, (j, t)\}$ is the transmission policy of DU_i given a transmission request is sent to j at time t, and $\pi_{i,0} = \{h_i\}$ is the policy of DU_i given no request is sent at time t. See [4] for more details.

5. SIMULATIONS AND RESULTS

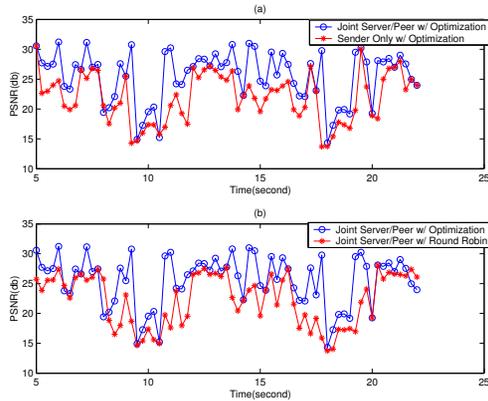This section presents the simulation results of the joint sender/peer optimized video streaming scheme.

Fig. 2. PSNR versus Time ('foreman')

PLR		PSNR w/ JSPO	PSNR w/ SOO	PSNR w/ JSPR
P_{server}	P_{peer}			
4%	4%	28.95	24.24	25.17
3%	6%	27.53	22.21	23.60
2%	7%	29.41	21.60	26.23

JSPO: Joint Sender/Peer w/ Optimization
SOO: Server Only w/ Optimization
JSPR: Joint Server/Peer w/ Round-Robin

Fig. 3. Average Streaming Performance ('foreman')

5.1. Simulation Framework

We compare the Peak Signal-to-Noise Ratio (PSNR) of received videos achieved by three different schemes: Joint Server/Peer with Optimization (JSPO), Server Only with Optimization (SOO) and Joint Server/Peer with Round- Robin (JSPR). In the first scheme, we implement the asynchronous clocks and point-to-point optimized streaming as presented in Section 4.2. In the second scheme, we disable the connection with the peer node and the client can only request for video transmission from the server. In the third scheme, instead of using rate-distortion optimization, we implement round-robin scheduling to select data units for (re)transmission.

We use the 300-frame standard video sequence `foreman` as the source. The sequence is encoded using H.263 version2 at QCIF, 30 frames per second and 120kps. The I-frame frequency in the sequence is 1 in 25 frames. We set the round-trip-time (RTT) to be 100ms in the link between the server and the receiver, and 30ms between the peer and the receiver.

5.2. Simulation Results

Figure 2 compares the performance of the three different schemes. For this set of results, the average packet loss rate (PLR) is 0.02 between the server and the receiver, and 0.06 between the peer and the receiver. Figure 2 shows that JSPO proposed in this paper has much better performance than the other two schemes. Figure 3 shows how the three schemes perform under different link condi-

tions. The average PSNR of `foreman` sequence as perceived by the client is computed for each case. Again, our proposed scheme, JSPO, consistently achieves better performance. Results also show that using multiple senders via round-robin scheduling achieves better PSNR than performing rate-distortion optimization for a single sender when a peer-client link has slightly higher loss rate but lower delay than the server-client link.

6. CONCLUSION AND FUTURE WORK

We propose a joint server/peer receiver-driven video streaming scheme that leverages infrastructure and ad-hoc communication modes to optimize the performance for a single wireless client. We formulate this path diversity streaming as a combinatorial optimization problem, and combine the application of asynchronous clocks and rate-distortion framework to solve it. Our NS simulations show that our scheme can provide better performance than the single sender scheme or the round-robin scheme. Currently, we assume that the client has full knowledge of the video content kept in its peers and the remote server. As part of our future work, we will address the dynamic content discovery problem. We will also investigate how the mobility patterns of the client relative to its peers affect the streaming performance.

7. REFERENCES

[1] S. Krishnamachari, M. van der Schaar, S. Choi, and X. Xu, "Video streaming over wireless LANs: A cross-layer approach," in *The 13 th International Packetvideo Workshop*, Nantes, France, April 2003.

[2] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," in *IEEE Transcations on Circuits and Systems for Video Technology*, June 2002, vol. 12, no.6.

[3] "Network simulator," http://www.isi.edu/nsnam/ns/.

[4] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," February 2001, Microsoft Research Technical Report MSR-TR-2001-35.

[5] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized media streaming," in *ICIP*, Sept. 2003.

[6] T. Nguyen and A. Zakhor, "Distributed video streaming over the internet," in *SPIE Multimedia Computer and Networking*, Jan. 2002.

[7] R. Rejaie and A. Ortega, "Pals: Peer-to-peer adaptive layered streaming," in *ACM NOSSDAV*, Monterey, California, USA, June 1-3 2003.

[8] A. Miu, J. G. Apostolopoulos, W. Tan, and M. Trott, "Low-latency wireless video over 802.11 networks using path diversity," in *ICME*, Baltimore,USA, July 2003.

[9] V. N. Padmannabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," Microsoft Research Technical Report MSR-TR-2003-11, March 2003.

[10] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM*, 2000.