# MAI–JSQ: A Cross–Layer Design for Real–Time Video Streaming in Wireless Networks

Yo Huh, Ming Hu, Martin Reisslein, and Junshan Zhang,
Dept. of Electrical Engineering,
Arizona State University, Tempe AZ 85287–7206,
{yohuh, ming.hu, reisslein, junshan.zhang}@asu.edu,
Web: http://www.eas.asu.edu/~{mre, junshan}

***Abstract*— We propose a cross–layer design for the real–time streaming of prerecorded video with prefetching to clients in wireless CDMA networks. Our design exploits the recently discovered temporal correlation structure of the multiple access interference (MAI) which enables accurate prediction of the MAI levels. Based on the MAI prediction, we optimize the transmissions of the video traffic so as to make judicious use of the radio resources and therefore to achieve small video playback starvation probabilities. A key component of our transmission control is the Join–the–Shortest–Queue (JSQ) scheduling, which selects the client with the smallest prefetched reserve for transmission (provided that the client's predicted MAI is low). Our simulation results indicate that in typical scenarios, the starvation probabilities using our cross–layer design with the MAI prediction, are at least one order of magnitude smaller than of the conventional JSQ protocol with link probing at the link/MAC layer.**

## I. Introduction

Real–time streaming of prerecorded continuous media, such as CD–quality audio clips, entertainment or instructional video, or news video clips, is expected to account for a large portion of the traffic in future wireless networks. The variability of wireless link conditions, together with the high (bit rate) variability and stringent timing constraints of continuous media, makes streaming in wireless networks very challenging. In this paper we propose a cross–layer design for the real–time streaming of prerecorded video to clients in wireless CDMA networks. Our basic strategy is to overcome the variabilities of the wireless links and the video traffic by prefetching parts of the ongoing streams into prefetch buffers in the clients. We propose a novel cross–layer design which integrates prediction of the multiple access interference (MAI) with the scheduling of the video frames. More specifically, our design exploits the recent findings on the temporal correlation structure in the MAI to predict the future MAI levels based on measurements of the past MAI levels. (These measurements are standard in modern wireless systems.) The MAI level predictions are utilized in our transmission control. We control the code assignment (packet scheduling), as well as spreading gain/coding rate, such that the wireless transmission resources are utilized judiciously and the video clients experience small playback starvation probabilities. More specifically, priority is given to video clients with small prefetched reserves according to the Join–the–Shortest–Queue (JSQ) scheduling policy while making sure that the spreading gain/coding rate is set such that packet transmissions are successful (based on the MAI level predictions).

We note that video streaming over wireless links has recently attracted a great deal of attention. The existing studies attempt to overcome the challenges of wireless video streaming by employing a wide range of techniques, such as adaptive encoding (e.g. [1], [2], [3]), hybrid automatic repeat request and forward error correction (e.g. [4], [5]), adaptive resource allocation and scheduling (e.g. [6]), and scalable encoded video (e.g. [7], [8]). Our work has two unique aspects with respect to the existing techniques. First, we exploit the buffering capability in the wireless clients to prefetch segments of the ongoing streams. The prefetched segments help in overcoming the variability of the wireless links. Secondly, we exploit (rather than combat) the burstiness of the wireless link conditions in that we incorporate physical layer MAI prediction, in our cross–layer design. Our techniques are largely orthogonal to the techniques that have already been studied in the literature and may be combined with the existing approaches to form hybrid schemes.

The contributions of this paper are threefold. First, we evaluate the JSQ prefetching protocol for wireless links [9] in a more general setting. The evaluation in [9] focused on a single cell serving only video clients, and orthogonal CDMA codes were assumed. In contrast, we consider a multi–cell network (with intra– and inter– cell interference) serving a mix of data clients and video clients. In addition, we consider correlated (pseudo–noise) CDMA codes for the different clients which are more applicable to multipath fading environments. Secondly, we propose a cross–layer design for video streaming, which incorporates MAI prediction and controls the transmission according to the amount of prefetched video (application layer) traffic. Thirdly, we present simulation results that demonstrate the benefits of the proposed cross–layer design. We find that in typical scenarios, our cross–layer design achieves playback starvation probabilities that are at least one order of magnitude smaller than with the conventional JSQ scheme with link layer probing [9].

## II. System Model

### A. Overview of Wireless System

We consider the downlink transmission in a cellular CDMA wireless system. Let $C$ denote the number of cells under consideration. Let $J_c,\ c = 1, \ldots, C$, denote the number of users in cell $c$. We consider a generic rate adaptive system [10] that adapts the transmission (bit) rates in the downlink direction (from a base station to one of its wireless clients) by varying the number of used CDMA codes, or the spreading gain/coding rate, or a combination thereof. Let $R_{jc}$ denote the maximum number of parallel codes that can be processed by the radio front–end of client $j$, in cell $c$. Let $Z_{jc}(t),\ 0 \leq Z_{jc}(t) \leq R_{jc}$, denote the number of parallel CDMA codes used for transmission to wireless client $j$ in cell $c$. Throughout we assume that the parallel codes used for transmission to a given client are orthogonal, whereas the codes used for transmissions to different clients are correlated pseudo–noise codes. We initially consider a system with a slotted time division duplex (TDD) timing structure, which provides alternating forward (base station to clients) and backward (clients to base station) transmission slots. We assume that $(i)$ the clients acknowledge packets sent in a forward slot by the end of the subsequent backward slots, and $(ii)$ the client feeds back its measured received signal strength and interference level to the base station in every slot. Both of these are standard features in 3G wireless systems, such as UMTS and CDMA2000.

### B. Multiple Access Interference (MAI) Model

Without loss of generality, we consider client $j$, in cell 1. Let $I_{jc}$ denote the total received power (interference) at client $j$ from the base station $c,\ c = 1, \ldots, C$. Assuming the matched filter is employed, the intercell MAI from cell $c,\ c = 2, \ldots, C$ is approximately

$$I_{jc}(t) = \left( \sum_{l=1}^{J_c} P_{lc}(t) \cdot Z_{lc}(t) \right) \cdot g_{jc}(t), \tag{1}$$

where $P_{lc}(t)$ is the transmission power from base station $c$ to its client $l$, and $g_{jc}(t)$ is the fading coefficient from base station $c$ to client $j$. Similarly, the intracell MAI is approximately $I_{j1}(t) = \left( \sum_{l \neq j} P_{l1}(t) \cdot Z_{l1}(t) \right) \cdot g_{j1}(t)$. We assume that fading is due to $(i)$ propagation attenuation of the form $d^{-3.5}$, $(ii)$ log–normal shadowing, and $(iii)$ Rayleigh fading with a Doppler shift of 5Hz. We use a filtered Gaussian noise model in our simulations. Let

$$I_j^{MAI}(t) = \sum_{c=1}^{C} I_{jc}(t), \tag{2}$$

denote the MAI for client $j,\ j = 1, \ldots, J_1$, in cell 1.

Let $\mathrm{SINR}_j(t)$ denote the signal–to–interference–plus–noise ratio experienced by client $j$. In a large network with many clients, $\mathrm{SINR}_j(t)$ is well approximated by

$$\mathrm{SINR}_j(t) = \frac{P_{j1}(t) \cdot g_{j1}(t)}{\sigma^2 + I_j^{MAI}(t)/G_j(t)}, \tag{3}$$

where $\sigma^2$ is the variance of the ambient additive white Gaussian noise and $G_j(t)$ is the processing gain of client $j$ in cell 1. In spread spectrum systems, $G_j(t) = W/R_j(t)$, where $R_j(t)$ is the transmission (bit) rate on a given code to client $j$ in cell 1, and $W$ is the bandwidth. A packet sent by the base station on a given code to client $j$ during a forward slot is considered successful if the $\mathrm{SINR}_j(t)$ is above a threshold $\gamma$ throughout the forward slot.

### C. Wireless Client and Traffic Model

In our study we assume that each cell has possibly data clients and video clients. Data clients receive data traffic, such as web pages, e–mail, ftp, from their base station. We follow the data traffic model studied in [11], where data transmissions to a given client are conducted in a ON–OFF fashion. The ON and OFF periods are modeled by heavy tailed distributions. The heavy tailedness of the ON–OFF transmission process which "modulates" the fading process gives rise to self–similarity in the MAI process [11]. We exploit this self–similarity for interference prediction and adaptive transmission control in our MAI–JSQ cross–layer design.

Continuous media clients receive continuous media streams, such a CD quality audio or video. Our focus is on the real–time streaming of prerecorded variable–bit–rate encoded video. Without loss of generality, we consider the video clients in cell 1. Let $J_v\ (\leq J_1)$ denote the number of video clients in cell 1. We assume that each video client receives one stream, thus there are $J_v$ ongoing streams in cell 1. We note that for the prerecorded videos the frame sizes and the frame periods are fully known when the streaming commences.

We assume that each video client $j$ has a prefetch buffer of capacity $B_j$ (in bits). When a video client requests a new video, the base station packetizes the video frames and transmits the packets over the wireless link. The arriving packets are placed in the clients prefetch buffer. The video playback on the client's monitor commences as soon as a few frames have arrived. Under normal circumstances the client displays frame $n$ of stream $j$ for its frame period, then removes frame $n + 1$ from its prefetch buffer, decodes it, and displays it for its frame period. If at one of these epochs there is no complete frame in the buffer, then the client suffers playback starvation and loses a part or all of the current frame. As measures for the clients' playback starvation probability, we define the following two probabilities. First, we define the *information loss probability* of client $j$, $P^{\mathrm{P}}_{\mathrm{loss}}(j)$ as the long run fraction of video encoding information (bits) that misses its playback deadline at client $j$. We define the average information loss probability as

$$P^{\mathrm{P}}_{\mathrm{loss}} = \frac{1}{J_v} \sum_{j=1}^{J_v} P^{\mathrm{P}}_{\mathrm{loss}}(j). \tag{4}$$

Similarly, we define the *frame loss probability* of client $j$, $P^{\mathrm{F}}_{\mathrm{loss}}(j)$ as the long run fraction of frames that miss their playback deadline of client $j$ and the average frame loss probability as

$$P^{\mathrm{F}}_{\mathrm{loss}} = \frac{1}{J_v} \sum_{j=1}^{J_v} P^{\mathrm{F}}_{\mathrm{loss}}(j). \tag{5}$$

For each video client $j,\ j = 1, \ldots, J_v$, the base station maintains two counters: $p(j)$ gives the current length of the prefetched video segment in client $j$ in seconds, and $b(j)$ gives the current prefetch buffer occupancy in bits.

## III. MAI–JSQ Cross–Layer Design

We propose a novel cross layer design for minimizing the loss probabilities of the video clients (while ensuring sufficient

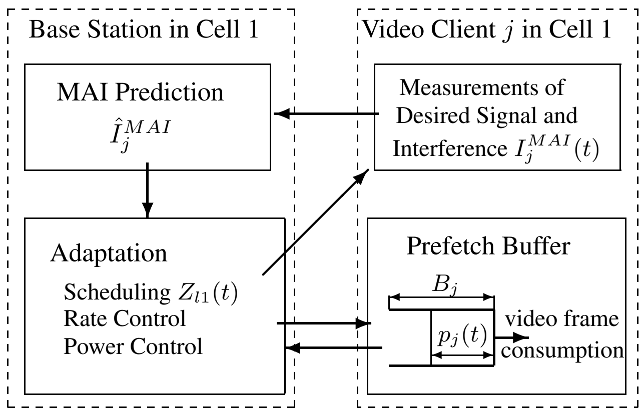


Fig. 1. Conceptual illustration of MAI–JSQ cross–layer design

quality of service for data users). The two key challenges in providing good video quality at the wireless video clients are $(i)$ the burstiness of the VBR video traffic, and $(ii)$ the burstiness of the wireless link conditions (i.e., the SINR's).

Our basic strategy is to build up prefetched reserves in the clients' prefetch buffers when the link conditions are good and spare transmission capacity is available. These reserves allow the clients to $(i)$ play out very–high–bit rate scenes (e.g., action scenes), and $(ii)$ continue playback during periods of adverse link conditions (e.g., long deep fades). The two ingredients of our design are Join–the–Shortest–Queue (JSQ) scheduling and Multiple Access Interference (MAI) prediction.

### *A. Outline of Approach*

Our approach has four components, as conceptually illustrated in Figure 1. These are $(i)$ measurement of signal strength and interference at the client, $(ii)$ prefetching buffering at the client, $(iii)$ MAI prediction at the base station, and $(iv)$ adaptive transmission control at the base station. At the end of each backward slot, each client $j$ feeds back its measured received signal strength over the expired slot and its measured MAI $I_j^{MAI}(t)$ to the MAI prediction at the base station. During each backward slot each client $j$ also acknowledges all packets that were successfully received in the preceding forward slot. With these acknowledgments and the known frame sizes and frame periods, the base station updates the buffer occupancy counters $p_j$ and $b_j$. Based on the signal strength and MAI measurements, the MAI prediction (see Section III-B for more details) provides MAI predictions $\hat{I}_j^{MAI}$ for each client $j$ for the next forward slot. In turn, based on the predicted MAI levels, $\hat{I}_j^{MAI}$, and the current prefetched reserves $p_j$, the transmission control (see section III-C for more details) determines the transmission schedule for the next forward slot. The base station then executes the packet transmission schedule during the next forward slot and waits for the arrival of the measurements and acknowledgments during the subsequent backward slot.

### *B. MAI Prediction*

We exploit the self–similarity of the MAI (see section II-B) for MAI prediction. Initially, we follow [11] and consider prediction of the total MAI $\hat{I}_j^{MAI}$ at the packet level and longer time scales. The packet level MAI predictor takes the form

$$\hat{I}_j^{MAI,p}(n) = I_j^{MAI}(n-1), \tag{6}$$

where $n$ denotes the slot index (discrete time). The large time scale MAI prediction takes the form

$$\hat{I}_j^{MAI,l}(n) = \frac{1}{M}\sum_{m=n-M}^{n-1} I_j^{MAI}(m), \tag{7}$$

where the length $M$ of the estimation interval is pre–determined. For multi–time–scale MAI prediction we employ

$$\hat{I}_j^{MAI}(n) = \xi \cdot \hat{I}_j^{MAI,p}(n) + (1-\xi)\cdot \hat{I}_j^{MAI,l}(n),\ 0 \le \xi \le 1. \tag{8}$$

We note that these MAI predictions rely on temporal correlations in the MAI process. It has been proven formally that the MAI process generated by data clients is self–similar, and thus highly correlated [11]. The MAI process generated by video clients, on the other hand, has yet to be analyzed in detail. While the video traffic is typically self–similar, the JSQ scheduling may smooth out some of the burstiness of the video traffic. However, we expect that the total MAI is sufficiently correlated for a typical mix of data clients and video clients in the cells. Indeed our initial simulation results (see Section V) confirm this conjecture.

### *C. Transmission Adaptation*

The transmission control in our MAI–JSQ cross–layer design encompasses the scheduling of the (application layer) video (frame) traffic, the packet scheduling, as well as the control of the spreading gain/coding rate. The joint optimal control of the tuning parameters and algorithms across the classical network protocol layers of our wireless video streaming system is a complicated stochastic optimization problem. As noted in Section I, significant advances have recently been made in the various aspects of video streaming over wireless links. However, the optimal transmission control for the real–time streaming of continuous media in wireless networks with *prefetching* is largely an open problem, which we are addressing in our ongoing work.

To demonstrate the benefits of the proposed cross–layer design for video streaming, we start with some heuristic transmission control strategies. In our most basic strategy, we employ multi–code CDMA with fixed spreading gain/coding rate (and assumed power control). For a fixed processing gain $G_j$, the MAI predictor makes a prediction as to whether the $\mathrm{SINR}_j$ of client $j$ will be above or below the required threshold $\gamma$ for the upcoming forward slot. If yes, client $j$ is considered "eligible" for packet scheduling and participates in the JSQ scheduling; if no, client $j$ is ignored in the scheduling (and we fix $Z_{j1} = 0$ for this slot). In the JSQ scheduling, packet transmissions are iteratively assigned to the eligible video clients with the shortest prefetched reserve $p(j)$ (see [9] for the detailed assignment algorithm and video frame scheduling). For every packet transmission assigned to client $j$ a separate CDMA code is used, and $Z_{j1}$ is incremented by one. (The $Z_{j1}$'s are initialized to zero at the beginning of the assignment.) At most $R_{j1}$ parallel packet transmissions are assigned to client $j$ (in cell 1) in a given forward slot, i.e., $Z_{j1} \leq R_{j1}$. Also, the total number of packet transmissions to video clients in a given forward slot is limited to a pre–specified parameter $S$, i.e., $\sum_{j=1}^{J_v} Z_{j1} \leq S$, which ensures that data clients receive a sufficient level of quality of service.

## IV. Simulation Model

In this section we describe our simulation set–up. We simulate a network consisting of $C = 7$ cells. In each cell, 30 video clients and 64 data clients share correlated pseudo–noise (PN) CDMA codes for the downlink transmission. Each base station allocates at most $S = 32$ codes in parallel to its video clients. We determine the performance metrics information loss probability and frame loss probability for each of the $J_v = 30$ video clients in cell 1. These video clients are placed randomly in cell 1. Based on each video client's location and a standard log–normal shadowing model, we calculate the large time scale fading from the base stations in cells $1, 2, \ldots, 7$ to each video client in cell 1. A filtered Gaussian noise model is employed to model the Rayleigh fading with a Doppler shift of 5 Hz. The large time scale fading is combined with the Rayleigh fading and propagation attenuation of the form $d^{-3.5}$ to give the fading coefficient $g_{jc}(t)$ from each base station $c,\ c = 1, 2, \ldots, 7,$ to each video client $j,\ j = 1, 2, \ldots, J_v$, in cell 1.

The video traffic from the base station in cell 1 to each of its $J_v$ clients is simulated using 18 traces of MPEG–4 encoded video [12], which have highly variable bit rates. The traces have a fixed frame rate of 25 frames per second, i.e., each video frame has a frame period (display time) of 40 msec. Each video frame is packetized into 80 byte packets. The traces are scaled such that the packetized video traffic has an average bit rate of 64 kbps. In the simulated wireless system, time is divided into 10 msec slots. Each slot is subdivided into a downlink transmission slot and an uplink transmission slot, in a time division duplex (TDD) fashion. The spreading gain and coding rate are set such that one CDMA code channel accommodates one 80 byte packet in one downlink transmission slot, i.e., such that one CDMA code channel provides a downlink transmission capacity of 64 kbps. For each of the $J_v$ video clients we randomly select one of the 18 traces at the beginning of the simulation. We generate random starting phases into each selected trace. The starting phases are independent and uniformly distributed over the lengths of the selected traces. All video clients start with empty prefetch buffers. The first frame is removed from the prefetch buffer at the end of the first frame period (that is, at the end of the fourth time slot). Furthermore, we generate random stream lengths (life times) for each selected video. The stream lengths are independent and are drawn from an exponential distribution with a fixed average stream life time. (The traces are wrapped around if a stream extends beyond the end of the trace.) When the last frame of a given video stream is removed from the prefetch buffer, we assume that the video client immediately requests a new video stream. For the new video stream we again select randomly one of traces, a new independent random starting phase into the trace, and a new independent random stream lifetime. Thus, there are always $J_v$ video streams in progress in cell 1. For simulating the intercell interference due to the video transmissions by the base stations $2, 3, \ldots, 7$, we conservatively assume that these base stations use the 32 codes available for video transmission all the time. The downlink traffic to the data clients in all 7 cells is simulated as ON/OFF traffic with heavy–tailed ON and OFF periods, following [11].

Throughout we consider scenarios where all video clients in cell 1 have the same prefetch buffer capacity $B$ and support the same number of parallel channels, i.e., $B_{jc} = B$ and $R_{jc} = R$ for all $j = 1, \ldots, J_c$, and $c = 1, \ldots, C$. We consider $B = 0.8$, 1.6, 3.2, 8, 16, 32, 64, 80, 128, and 256 kBytes and $R = 3$ or 8. In all our simulations, we allow for a warm–up period of 60 minutes (i.e., 90,000 simulated frame periods). We estimate the information loss probabilities $P_{\text{loss}}^{\text{P}}(j)$ and the frame loss probabilities $P_{\text{loss}}^{\text{F}}(j)$ using the method of batch means with a batch length of 10 minutes (15,000 frame periods) and a separation of 10 minutes between successive batches. We run all simulations until the 90% confidence intervals of the average frame loss probability $P_{\text{loss}}^{\text{F}}$ are less than 20% of the corresponding sample means.

In our simulations, we consider two schemes: the cross–layer MAI–JSQ scheme introduced in this paper and the JSQ scheme with link probing [9]. The JSQ scheme with link probing works exclusively at the link layer. If a packet sent in a forward slot is not acknowledged by the end of the subsequent reverse slot, the scheme "probes" the affected wireless link by sending at most one (probing) packet per slot, i.e., by restricting $R_{j1}$ to one. If a probing packet is successfully acknowledged, then $R_{j1}$ is reset to its original value.

## V. Simulation Results

We now present simulation results for our cross–layer design. Figure 2 gives the average loss probabilities $P_{\text{loss}}^{\text{P}}$ and $P_{\text{loss}}^{\text{F}}$ with their 90% confidence intervals as a function of the prefetch buffer capacity $B$. We observe that for the MAI–JSQ scheme the average loss probability steadily drops as the buffer size increases. For JSQ with link probing, on the other hand, the loss probability drops quickly for small buffers. However, for buffers larger than 3.2 kBytes the drop–off is very slow. For small buffers, up to 3.2 kBytes (equivalent to $3.2 \cdot$ 8 kbit/64 kbps = 0.4 sec run time of average bit rate video) both schemes give roughly the same loss probability. In this small buffer regime the performance of both prefetching schemes is limited by the available buffer space. Both schemes tend to keep the small buffers completely filled. However, the consumption of high bit rate scenes or fades that cut off a client from the base station can quickly deplete the small buffers and lead to starvation. After the bit rate returns to around the average and the link conditions improve the transmission control (JSQ scheduler) gives priority to the depleted client. The small buffer is quickly refilled but its small size prevents the build–up of larger reserves (the transmission controller schedules only clients that can accommodate at least one additional packet in their buffers). For buffers larger than 0.4 sec run time of average bit rate video, the performance of the two schemes differs dramatically. For buffers of 128 kBytes (=16 sec run time of average bit rate video) and larger, MAI–JSQ achieves the average loss probabilities that are over one order of magnitude smaller than with JSQ with link probing. For these large buffers, the client buffer capacity is no longer the primary limitation to building up prefetched reserves and reducing the loss probability. Instead, the number $S$ of available codes (i.e., the available transmission capacity) becomes the primary bottleneck. By employing the physical layer MAI prediction, the MAI–JSQ scheme uses the codes more judiciously, as it assigns codes only to clients for which the MAI predictor predicts successful packet transmission. The JSQ scheme with link layer probing, on the other hand, uses up CDMA codes to probe out the links with adverse transmission conditions, which is a poor strategy when the codes are the primary bottleneck.

We also observe from Figure 2 that with both schemes, $P_{\text{loss}}^{\text{F}}$ is smaller than the corresponding $P_{\text{loss}}^{\text{P}}$. This is because we employed a skipping rule in the JSQ scheduling which $(i)$ skips packets from a frame that will likely miss its playout deadline, and $(ii)$ prefetches instead packets for the next frame. The skipping rule tends to skip a few extremely large frames; the loss of these large frame contributes proportionally more to $P_{\text{loss}}^{\text{P}}$ than to $P_{\text{loss}}^{\text{F}}$, resulting in the results given in Figure 2. The advantage of the skipping rule is that it does not expend wireless transmission resources on frames that can not be completely delivered by their deadline. A potential drawback of the skipping rule is that it may skip the large intra–coded (I) frames in MPEG–coded videos, which are required for the decoding of the subsequent predictive encoded (P and B) frames. In our ongoing work we are addressing this issue in detail. We are exploring the following priority scheme. In case the timely delivery of an I frame is endangered, we skip the B (and possibly P) frame(s) preceding the I frame and prefetch instead for the I frame to ensure its timely delivery. With such a priority scheme, we expect the $P_{\text{loss}}^{\text{P}}$ to drop and $P_{\text{loss}}^{\text{F}}$ to increase. To avoid excessive clutter in the plots, we focus on $P_{\text{loss}}^{\text{F}}$ in the remainder of this paper.

In Figure 3 we plot the average frame loss probability $P_{\text{loss}}^{\text{F}}$ as a function of the client buffer capacity $B$. We consider systems where the clients can be assigned up to $R = 3$ or $R = 8$ parallel CDMA codes. We observe that the different limitations on the maximum number of assignable codes $R$ have a relatively small impact on the loss probability. Interestingly, the larger $R$ results in slightly larger loss probabilities, (except for MAI–JSQ with large buffers). This slightly worse performance for larger $R$ is because the JSQ scheduling tends to "overreact" to the client buffer contents when $R$ is too large, which tends to result in imbalanced code assignments with many codes being assigned to a few clients. With an imbalanced code assignment the system faces many dropped packets when the channel of a client with many assigned codes was incorrectly predicted as good. The important conclusion from this experiment is that low–cost clients which can handle only a relatively small number of parallel code channels at their radio front–end obtain significant benefit from the proposed MAI–JSQ scheme.

In Table I we give the average frame loss probability as a function of the average lifetime (duration) of the video streams. The client buffers are fixed at $B = 128$ kBytes (= 16 sec of run time of average bit rate video) in this experiment and the clients are assigned at most $R = 3$ codes in parallel. We observe that for a very short average stream lifetime of 2 seconds, both MAI–JSQ and JSQ with link probing give roughly the same loss probability. Stream durations this short allow only for limited accumulation of prefetched reserves. With longer stream durations the MAI–JSQ scheme—in contrast to the JSQ–probe scheme—is able to efficiently take advantage of the prefetch buffer in the clients. The loss probability is approximately cut in half as the average stream duration increases from 2 to 5 seconds; with a further increase to 10 seconds, the loss probability is cut in half once more. Overall, we conclude that with the proposed MAI–JSQ scheme relatively short lived streams allow already for effective prefetching.

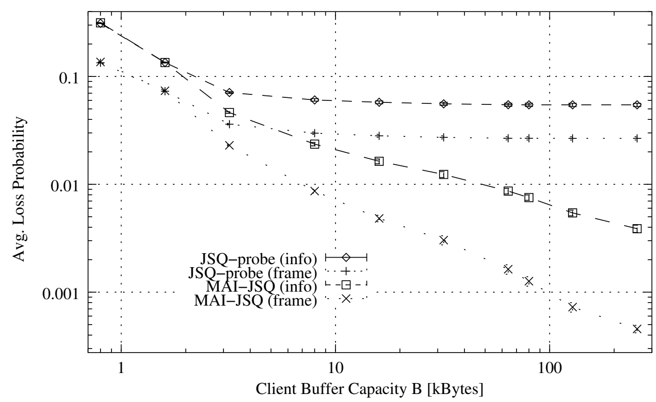


Fig. 2. Average loss probabilities, $P_{\text{loss}}^{\text{P}}$ and $P_{\text{loss}}^{\text{F}}$ with 90% confidence intervals as a function of buffer capacity $B$. ($J_v$ = 30 video clients share $S = 32$ codes, at most $R$ = 3 codes per client, average video stream lifetime = 10 minutes)

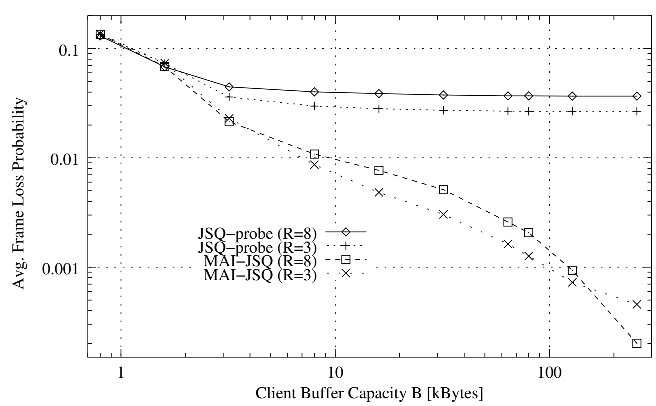


Fig. 3. Average frame loss probability $P_{\text{loss}}^{\text{F}}$ as a function of client buffer capacity $B$ for $R = 3$ and 8 parallel codes

## VI. Conclusion

We have proposed a cross–layer design which incorporates MAI prediction at the physical layer and JSQ scheduling at the link/MAI layer for the downlink streaming of prerecorded video with prefetching. Our simulation results indicate that the novel cross–layer design achieves video playback starvation probabilities that are over one order of magnitude smaller than with conventional JSQ scheduling with link layer probing.

TABLE I

AVERAGE FRAME LOSS PROBABILITY AS A FUNCTION OF AVERAGE VIDEO STREAM LIFE TIME (CLIENT BUFFER $B = 128$ KBYTES, FIXED.)

| Life T. [sec.] | $P^{\rm F}_{\rm loss}$(MAI–JSQ) | $P^{\rm F}_{\rm loss}$(JSQ-probe) |
|---|---|---|
| 2 | $2.0 \times 10^{-2}$ | $2.0 \times 10^{-2}$ |
| 5 | $1.0 \times 10^{-2}$ | $2.0 \times 10^{-2}$ |
| 10 | $5.6 \times 10^{-3}$ | $2.0 \times 10^{-2}$ |
| 50 | $1.7 \times 10^{-3}$ | $2.0 \times 10^{-2}$ |
| 100 | $1.3 \times 10^{-3}$ | $2.0 \times 10^{-2}$ |
| 600 | $7.3 \times 10^{-4}$ | $2.0 \times 10^{-2}$ |
| 1200 | $8.5 \times 10^{-4}$ | $2.0 \times 10^{-2}$ |

In our ongoing work we are developing more sophisticated algorithms to refine the MAI predictor and transmission control components in our design.

## ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation through grants Career ANI–0133252 and ANI–0208135 and Arizona State University through the IT301 initiative.

## REFERENCES


[1] C. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst–error wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, May 1999.

[2] W. Kumwilaisak, J.W. Kim, and C.C.J. Kuo, "Reliable wireless video transmission via fading channel estimation and adaptation," in *Proceedings of IEEE WCNC*, Chicago, IL, Sept. 2000, pp. 185–190.

[3] A.S. Tosun and W.C. Feng, "On improving quality of video for H.263 over wireless CDMA networks," in *Proceedings of IEEE WCNC*, Chicago, IL, Sept. 2000, pp. 1421–1426.

[4] H. Liu and M. El Zarki, "Performance of H.263 video transmission over wireless networks using hybrid ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1775–1786, Dec. 1997.

[5] D. Qiao and K. G. Shin, "A two–step adaptive error recovery scheme for video transmission over wireless networks," in *Proceedings of IEEE Infocom*, Tel Aviv, Israel, Mar. 2000.

[6] M. Liu, W.-C. Feng, U. Mitra, and M. T. Liu, "Managing code assignment in multi-code CDMA wireless networks for the delivery of H.263 video streams," in *International Conference on Computer Networks and Mobile Computing 2001*, 2001, pp. 473–478.

[7] M. Naghshineh and M.W. LeMair, "End–to–end QoS provisioning in multimedia wireless/mobile networks using an adaptive framework," *IEEE Communications Magazine*, vol. 35, no. 11, pp. 72–81, Nov. 1997.

[8] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broad–band wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 6–20, Jan. 2001.

[9] F. Fitzek and M. Reisslein, "A prefetching protocol for continuous media streaming in wireless environments," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2015–2028, Oct. 2001.

[10] S. Nanda, K. Balachandran, and S. Kumar, "Adaptation techniques in wireless packet data services," *IEEE Communications Magazine*, vol. 38, no. 1, pp. 54–64, Jan. 2000.

[11] J. Zhang, M. Hu, and N. Shroff, "Bursty data over CDMA: Self similarity, rate control and admission control," in *Proceedings of IEEE Infocom '02*, New York, NY, June 2002.

[12] F. Fitzek and M. Reisslein, "MPEG–4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, November/December 2001, Video traces available at `http://www.eas.asu.edu/trace`.