A Game-Theoretic Approach to Low Energy Wireless Video Streaming

Ali Iranli, Hanif Fatemi, and Massoud Pedram
Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
Email: {iranli, fatemi, pedram}@usc.edu

Abstract: *This paper presents a dynamic energy management policy for a wireless video streaming system, consisting of a battery-powered client and a video server. The video quality in wireless streaming is a function of three factors: the encoding aptitude of the server, the decoding aptitude of the client, and the wireless channel conditions. First, the energy consumption of a wireless video streaming system is modeled considering these factors, and then, a model of the wireless video streaming system is presented. Using the proposed model, the optimal energy assignment for each video frame is derived so as to maximize the system lifetime while satisfying a given minimum video quality requirement. Experimental results show that the proposed policy increases the system lifetime of a wireless video streaming system by an average of 20%.*

1 Introduction

We have seen an explosive growth in wireless multimedia applications, e.g., streaming audio and video in today's consumer electronics. This trend is the result of the availability of mobile communication and computing systems. The mobility of these systems, in turn poses two challenges for the designers: (1) establishing and maintaining a stable communication channel for real-time operation and (2) power-aware operation so as to increase the lifetime of the battery-powered wireless system while meeting a minimum quality of service (QoS) requirement. Furthermore, it is desirable to provide a mechanism for graceful degradation in QoS so that a dynamic power manager (DPM) can incrementally trade off QoS for higher energy efficiency. Fine Granularity Scalability (FGS) coding technique [1], which was adopted as the standard in MPEG-4, provides an effective mechanism for graceful video quality degradation based on its hierarchical layer structure, which consists of a base layer and one or more (optional) enhancement layers. Although extensive studies have been conducted on the hierarchical layer structure of MPEG-4 and its error resiliency under fluctuations in the channel bandwidth [2][3][4], energy efficiency in a battery-powered server-client system has received little attention.

For the video streaming application, there are two sources of energy consumption in wireless mobile hosts: the computation energy for processing a video stream and the communication energy for transmitting and receiving the data. The computation energy of a server and a client is usually a strong function of the CPU frequency, which can be changed by employing methods such as dynamic voltage and frequency scaling (DVFS) [5]. The communication energy, on the other hand strongly affects the bit-error-rate (BER), and hence, the video quality.

A modern digital communication system, as depicted in Figure 1, comprises of two transceivers. A base-band transceiver, which uses digital signal processing, encodes the input data bits so as to increase the data fidelity against unexpected changes in the channel characteristics. A pass-band transceiver, which uses analog signal processing, modulates digital data into analog symbols and guarantees a minimum received signal-to-noise-ratio (SNR). In order to design a low-energy communication system, the overall energy consumption of the transmitter and receiver should be considered. There are detailed studies of the trade-off between energy consumption and BER in the communications field [6]. These studies can be divided into two main categories. The first set of techniques, which focus on the pass-band transceiver, exploit the fact that different modulation schemes result in different BER vs. SNR characteristics. The basic idea is that by adaptively changing the modulation and/or equalization, while keeping the received SNR at the receiver constant, one can achieve different BER. The second set of techniques, which focus on the base-band transceiver, study the interaction between code performance and encoder/decoder design complexity. The main idea is to add a number of error controlling bits to the original data bits to protect them from channel changes. The key trade off is between the complexity of the encoding/decoding algorithms and the BER.

The achievable video quality in the streaming video systems is determined by three factors: the encoding capability of the server, the decoding capability of the client, and the wireless channel error rate. It is well known that channel bandwidth fluctuation due to various external causes can result in the severe degradation in the video quality. This is due to the streaming nature of this real-time operation and the extra time overheads required for retransmissions if errors occur in the data packets.

The *encoding (decoding) aptitude* of the server (client) is defined as the amount of data that can be processed in a given deadline. When the server (or/and the client) changes its operating frequency and voltage to extend its battery lifetime, the encoding (decoding) aptitude is also affected; so is the quality of the streaming video. This scenario is not uncommon because many of the state-of-the-art processors which are designed for mobile applications are equipped with DVFS for low-power operation [8]. In [9] a low energy MPEG-4 streaming policy using a client-feedback method was proposed where the client's decoding capability at each time slot is sent to the server and the server adjusts its transmission rate based on the feedback from the client. By using this feedback approach, a significant amount of communication energy can be saved. However, the authors considered only energy consumption on the client side, and ignored the server in their analysis. Other related work include reference [10], where the authors proposed an energy-optimized image transmission system for indoor wireless applications, which exploits the variations in the image data and the wireless multi-path channel by employing dynamic algorithm

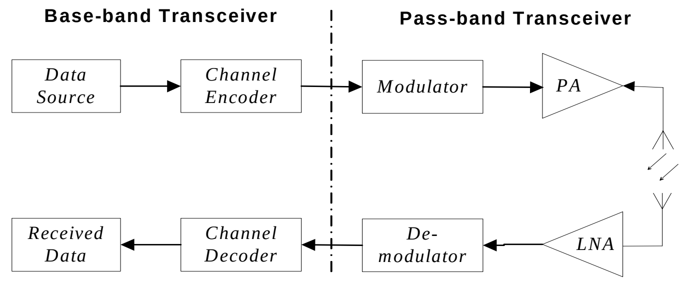


Figure 1. Communication system model

transformations and joint source-channel coding. A detailed energy model for the client-server system was proposed and a global optimization problem solved by using feasible direction methods that resulted in an average of 60% energy saving for different channel conditions.

In this paper, we propose an adaptive policy for a wireless video streaming system in which the optimal energy assignment to each video frame considering energy consumptions of both the server and the client is employed. The system energy consumption is thereby minimized, while meeting a required video quality constraint. Hierarchical game theory is used to solve the corresponding mathematical optimization problem. Experimental results show an average of 20% increase in the overall system lifetime. In [7] a low energy wireless communication system is described, where the modulation level and transmit power of the transmitter and the aptitude of channel decoder of the receiver are dynamically changed to match the characteristics of the communication channel, thereby, minimizing the energy consumption of the transceivers. The game theoretic framework proposed in this paper is similar to that of [7]. However, in our proposed work we consider not only the power dissipation of the transceivers but also the power dissipations of the encoding and decoding cores; furthermore, the application scenario that is targeted in our paper is video streaming in a client-server system whereas in [7] the application domain was related to mobile ad-hoc networks. In addition, in our work we satisfy an additional constraint, which is related to the average video quality received by the client. Finally, in our application, we impose a maximum power dissipation constraint on the server and client, which captures the electrochemical constraint on the maximum current output of the battery sources.

The remainder of this paper is organized as follows. Section 2 includes backgrounds on MPEG-4 FGS, model for energy consumption of the server and the client in the streaming system. Section 3 describes our energy assignment problem, and section 4 discusses the game theoretic formulation for this problem. Experimental results are described in Section 5 and it is followed by conclusion in Section 6.

2 Background

2.1 Fine Granularity Scalability (FGS)

To adapt to a time-varying channel capacity (which is in turn due to changes in the channel conditions, for example because of congestion or fading phenomena), a number of scalable video coding techniques have been proposed. Typical techniques include SNR scalability, temporal scalability, and spatial scalability in MPEG-2 and MPEG-4 [11]. In these layered scalable coding techniques, the encoded bit-stream consists of a *base layer* and several *enhancement layers*. The bit-rate of the base layer is determined by the minimum channel bandwidth and is sufficient to ensure a minimum video quality. The enhancement layers provide higher video quality when the channel has extra bandwidth for the transmission of extra layers.

The FGS video coding technique, which has been adopted as the standard in MPEG-4, provides a graceful adjustment to the video quality compared to other scalable coding technique because with this technique, any number of bits in the enhancement layers may be truncated according to the channel condition. Therefore, the *Video quality* (*VQ*) can be represented as a continuous (linear) function of the number of transmitted bits:

$$VQ = k.R_{send} = k.(R_b + R_e) \tag{1-a}$$

where k is a regression coefficient, R_{send} is the total bit-rate (in bits/sec), R_b is the base layer bit-rate, and R_e is the enhancement layer bit-rate. Note that R_b must be less than the minimum attainable bandwidth in the wireless channel; otherwise, no useful video transmission is possible and VQ goes to zero. R_e is varied in response to the channel conditions. R_{send} is thereby set to provide an acceptable level of video quality by transferring the minimum amount of video data to the client subject to the existing channel conditions and the remaining battery lifetimes of the video server and/or client. Note that the following inequality should also be satisfied,

$$R_{send} \le \kappa_f . f + \varepsilon_{VQ} \tag{1-b}$$

This inequality simply states that the total bit-rate in the channel should be less than or equal to the bit generation rate of the encoder's processing core, which in turn can be expressed, as first order approximation, in terms of operating frequency, f. κ_f and ε_{VQ} denote the coefficients of this first order approximation.

2.2 Energy Model of the Server

The energy consumption of the server for processing and transmitting a video frame may be written as:

$$E^S = E^S_{Comp} + E^S_{Comm} \tag{2}$$

where E^S_{Comp} and E^S_{Comm} denote the per-frame energy consumption costs of the computation and communication processes in the server. E^S_{Comp} and E^S_{Comm} are in turn calculated as follows:

$$\begin{aligned} E^S_{Comp} &= C^S_{eff} \cdot V_S^2 \cdot f^S \cdot \tau \\ E^S_{Comm} &= \left(P_{Enc} + P_{Mod} + p_{amp}\right) \cdot \tau \end{aligned} \tag{3}$$

where C^S_{eff} denotes the effective switched capacitance per clock cycle time in the server, V_S is the supply voltage level (assuming full swing transitions) of the server's processing core, f^S is the clock frequency of the server CPU, and τ is the time duration of a frame (i.e., inverse of its frame rate). P_{Enc}, P_{Mod}, and p_{amp} denote power consumptions of the corresponding blocks in the transmitter. In this equation, p_{amp} is the dominant term. The other terms tend to be smaller in magnitude and depend linearly on the symbol rate with an additional constant. Hence, for our optimization purposes, the communication energy consumption of the server may be approximated as:

$$E^S_{Comm} = \left(P_{Tx} \cdot R_s + P_{const} + p_{amp} \right) \cdot \tau \tag{4}$$

where P_{Tx} and P_{const} are the symbol-rate-dependent and constant power consumption components of the base-band transmitter. R_s denotes the symbol rate.

To characterize the bit error rate (BER) in terms of the power consumption of the transmitter, the relationship between the received signal-to-noise ratio (SNR) and the BER of the pass-band transceiver, i.e., the modulating/demodulating pair, can be used. For example, consider a Quadrature Amplitude Modulation (QAM) scheme where the BER is related to the received SNR by the following equations (cf. [6]):

$$\begin{aligned} BER &= 1-(1-P_{\sqrt{M}}) \\ P_{\sqrt{M}} &= 2.\left(1-\frac{1}{\sqrt{M}}\right).Q\left(\sqrt{3\cdot\frac{SNR_{rcvd}}{M-1}}\right) \end{aligned} \tag{5}$$

where M is the number of constellation points in the QAM modulation, typically $M = 2^b$ where b is the number of information bits represented by each constellation point. SNR_{rcvd} is the received signal-to-noise-ratio at the receiver. Let N_0, β, and R_s denote the noise spectral density, the spectral shaping factor, and the symbol rate, respectively. The received SNR is related to the transmit power level p_{amp}, noise in the channel P_{Noise}, and the path loss parameter, σ, by [6]:

$$SNR_{rcvd} = \frac{p_{amp}}{P_{Noise}} \cdot \sigma = \frac{p_{amp}}{N_0 \cdot \beta \cdot R_s} \cdot \sigma \tag{6}$$

For a given BER and modulation scheme, i.e., for fixed b, one can calculate the required SNR, from Eqn. (5), and then use Eqn. (6) to find the minimum required transmit power level. The overall energy consumption of the transmitter for transmitting a single symbol is then calculated from Eqn. (4).

2.3 Energy Model of the Client

The energy consumption of the client for receiving and processing a video frame may be written as:

$$E^C = E^C_{Comp} + E^C_{Comm} \tag{7}$$

where E^C_{Comp} and E^C_{Comm} denote the per-frame energy consumption costs of the computation and communication processes in the client. They are calculated as follows:

$$E^C_{Comp} = C^C_{eff} \cdot V_C^2 \cdot f^C \cdot \tau \tag{8}$$

where C^C_{eff} denotes the effective switched capacitance per clock cycle time in the client, V_C is the supply voltage level (assuming full swing transitions) of the client CPU, and f^C is the clock frequency of the client CPU. E^C_{Comm} is due to energy consumptions of the low noise amplifier, the demodulating block, and the channel decoding block and may be written as:

$$E^C_{Comm} = \left(P_{LNA} + P_{Demod} + P_{Dec} \right) \cdot \tau \tag{9}$$

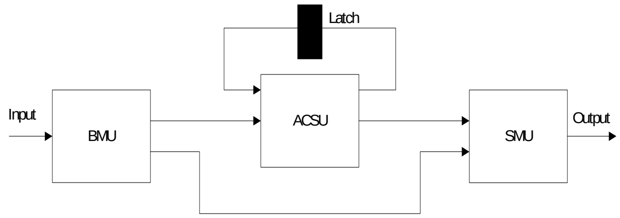
where P_{LNA}, P_{Demod}, and P_{Dec} denote the power consumptions of the corresponding blocks in the receiver. Considering that all blocks except the channel decoder consume fixed amount of power and do not respond to changes in channel conditions, for optimization purposes, the client energy consumption may be approximated as:

$$E^C_{Comm} \cong \left(P_{Rx} \cdot R_s + P_{const} + P_{Dec} \right) \cdot \tau \tag{10}$$

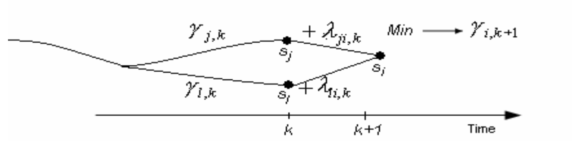
where P_{Rx} and P_{const} are the symbol-rate-dependent and constant components of power consumption of the pass-band receiver.

Typically, a channel decoder is a multi-stage implementation of a recursive decoding function. Therefore, the accuracy of decoding is increased as the number of decoding stages (iterations) increases. On the other hand, increasing the number of stages would increase the power consumption of the decoder. In this work, a Viterbi decoder is studied as the channel decoder. In *adaptive Viterbi algorithms* (AVA), developed in [12]-[14], the decoding performance is increased by reducing the number of operations required to decode a single bit. This is achieved by reducing the *Truncation Length* (*TL*) or by reducing the number of *Survivor Paths* (*SP*), i.e., those paths that are kept in order to find the optimum path. There are two main variations of the AVA. In the first variation, which is called the *T-Algorithm* [16], a fixed Threshold *T*, is chosen and then those paths that have path metrics equal to or less than T are included in the SP memory. In the second variation, called the *M-Algorithm* [15], a fixed number (M) of paths are kept and all other paths are discarded. These paths are selected by choosing the first M paths with the minimum path metric values.

Consider an adaptive Viterbi decoder with the functional block diagram depicted in Figure 2a. The decoder can be divided into three basic units. The input data (i.e., the noisy observation of the encoded information bits) is used in the Branch Metric Unit (BMU) to calculate the set of branch metrics $\lambda_{ji,k}$. These are then fed to the Add-Compare-Select Unit (ACSU) to update the path metric cost according to the following recursive equation:



a. Block diagram of the Viterbi decoder



b. Finding the optimum path

Figure 2: Adaptive Viterbi decoder

$$\gamma_{i,k+1} = \min(\gamma_{j,k} + \lambda_{ji,k}, \gamma_{l,k} + \lambda_{li,k}) \tag{11}$$

where $\gamma_{i,k}$ is the path metric cost for state s_i in time step k, and $\lambda_{ji,k}$ is the branch metric cost between states s_i and s_j from time instances k and $k+1$, respectively (cf. Figure 2b). The Survivor Memory Unit (SMU) processes the decisions that are being made in the ACSU in order to carry out the ACS-recursion and outputs the estimated path, with a latency of at least TL.

Power consumption for an adaptive Viterbi decoder may be macro-modeled by summing up the power consumption of each block times the number of paths that block is being used. This would result in following proposed power macro-model:

$$P_{Dec} = \left(P_{BMU} + 2^K.(P_{ACSU} + TL.P_{SMU})\right) \tag{12}$$

where P_{BMU}, P_{ACSU}, and P_{SMU} are the per-operation power consumptions of the corresponding modules and K represents the memory depth of the corresponding convolutional encoder. Notice that the ACSU module performs two additions and one comparison operation in each step (cf. Eqn. 11).

3 Lifetime Optimization Problem

The encoding/decoding aptitude of an image processing core is a strong function of its operating frequency and voltage level. Thus, one can characterize the video quality VQ of frame j as:

$$VQ_i = f(e_j^s, e_j^c, \omega_j) \tag{13}$$

where e^S_j and e^C_j denote the server and the client energy consumptions for frame j while ω_j denotes the wireless channel conditions for transmission of frame j.

We consider a wireless system operating over a fading channel. Time is assumed to be discrete. Each frame is processed in one timeslot of duration τ, where τ is the inverse of the frame rate. In each timeslot, the channel state changes among a number of different states chosen from a finite set $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$ according to a probabilistic model [16]. The server and the client are assumed to be battery-powered, each with a fixed number of energy units available for use. Each channel state ω_j determines the throughput that can be achieved per unit energy expended by the server/client. The video encoding/decoding processing cores can operate with frequencies f_S and f_C in a range bounded by a lower bound f_{min} and an upper bound f_{max}.

The problem at hand is to maximize the lifetime of the system, given the remaining energy levels of the server (E_0^S) and the client (E_0^C). In other words, the objective is to find an energy allocation pair (e_j^S and e_j^C) for each timeslot j of duration τ so as to maximize the overall *system lifetime*, Λ, (which is an integer multiple of τ) subject to:

$$\begin{aligned}
&\text{(I)} \quad \forall j: \frac{e_j^S}{\tau} \le p_{\max}^S \;\; and \;\; \frac{e_j^C}{\tau} \le p_{\max}^C \\
&\text{(II)} \quad avg(VQ_j) \ge VQ_{min} \\
&\text{(III)} \quad \sum_{j=1}^{\Lambda} e_j^S \le E_0^S \;\; and \;\; \sum_{j=1}^{\Lambda} e_j^C \le E_0^C
\end{aligned} \tag{14}$$

where $avg(VQ_j)$ denotes the average video quality over the system lifetime. Constraint (I) signifies the fact that power consumption of the server and client (i.e. energy consumption per frame) are upper-bounded due to restrictions on the maximum current output of their battery sources as well as the parasitics in the power distribution network. Constraint (II) guarantees the average video quality of the system whereas constraint (III) corresponds to the total energy bound. Note that this constraint is implicitly taken into account for an online solution to the optimization problem, since as soon as (III) is violated the system's life time will end. In the remainder of this paper subscript *j* is used to indicate the time slot *j* in which the corresponding parameters are evaluated.

At the beginning of each timeslot *j*, the server chooses an average power consumption value, e_{opt}^{S} / τ, for itself based on the current estimates of the remaining battery lifetimes of itself and the client and the predicted channel state for timeslot *j*. In our approach, the channel state for timeslot *j* is taken to be the same as the channel state *j-1* received from the client.[1] From the chosen energy consumption rate, the server will then put into effect the respective encoding (computation) and transmit (communication) parameters by looking up these values from a pre-computed and locally-stored *policy parameter table*. Since the actual channel state and/or the remaining lifetime of the client may be different from the one predicted on the server side, the client will have to solve another optimization problem. This time the client knows the actual channel state and the adopted parameters of encoding and transmitting on the server side and has more accurate and up-to-date information about its own remaining lifetime; therefore, the client can determine its average power consumption value, e_{opt}^{C} / τ, more effectively and thus obtain and enforce the reception and decoding parameters which are again looked up from the policy parameter table.

The aforementioned policy optimization problem, which involves a hierarchical variable determination process, is a form of multi-level optimization problems known as *Stackelberg game* [17] as detailed next.

4 A Game-theoretic Formulation

4.1 Background

In his monograph about market economy [18], H. V. Stackelberg used a hierarchical model to describe real market conditions. His model captured the scenario in which different decision makers attempt to make the best decisions in a market with respect to their own, generally different, utility functions. Generally speaking, these decision makers cannot determine their course of action independently of each other; rather, they are forced to act according to a certain hierarchy. Consider a simple case of such a problem where there are only two active decision makers. The hierarchy classifies these two decision makers into a leader, who acts independently of the market, and a follower, who has to act in a dependent manner. The leader is able to dictate the selling prices or to overstock the market with his products, but in making his decisions, he has to anticipate the possible reactions of the follower since his profit strongly depends not only on his own actions but also on the response of the follower. On the other hand, the choice of the leader influences the set of possible decisions as well as the objectives of the follower who in turn must react to the selections of the leader.

The aforementioned problem can mathematically be formulated as follows: Let *X* and *Y* denote the set of admissible strategies *x* and *y* of the follower and of the leader, respectively. Assume that the values of the choices are measured by the means of the functions $f_L(x,y)$ and $f_F(x,y)$, denoting the utility functions of the leader and follower, respectively. Then, with the knowledge of the selection *y* of the leader, the follower can select his best strategy $x(y)$ so that his utility function is minimized on *X*:

$$x(y) \in \Psi_L(y) = \underset{x}{Argmin}\left\{ f_F(x,y) \middle| x \in X \right\} \tag{15}$$

Being aware of this selection, the leader solves the Stackelberg game [18] for computing his best selection:

$$\underset{y}{Argmin}\left\{ f_L(x,y) \middle| y \in Y, x \in \Psi_L(y) \right\} \tag{16}$$

It is worth noting that the solutions to the Stackelberg game are different from the *Nash equilibrium points*, due to the special hierarchy that is imposed on the players. In Nash equilibrium solution all players have the same level of hierarchy and make decisions simultaneously, but in a Stackelberg game the decisions are made one after the other, following certain rules. In general, in an *n*-player Stackelberg game all players in same hierarchy level achieve the Nash's equilibrium point, but this is not true for players from different levels of hierarchy.

4.2 Application to Streaming Video

In our context, the follower and the leader become the client and the server, respectively. Strategy *x* for the client is the adoption of a specific vector of truncation lengths (*TL's* denoted by a_i*'s*) for the sub-carriers and an operating frequency for the decoding image processing core, f^C, and therefore, $X = \left\{ (a_1, a_2, \cdots, a_n, f^C) \middle| \forall i : a_i \in TLS,\ f^C \in FS \right\}$, where *n* is the number of sub-carriers in the Orthogonal Frequency Division Multiplexing (OFDM) signal, *TLS* denotes the set of all (feasible) *TL*'s for the adaptive Viterbi decoder, and *FS* is the set of feasible frequencies for the image processing core. Strategy *y* for the transmitter is a choice of specific overall transmission power level, p_{amp}, a set of modulation levels for the different sub-carriers, b_i's, and operating frequency for the encoding image processing core, f^S. Therefore, $Y = \left\{ (p_{amp}, b_1, b_2, \cdots, b_n, f^S) \middle| p_{amp} \in PLS, \forall i : b_i \in MLS,\ f^S \in FS \right\}$ where *MLS* and *PLS* denotes the sets of (feasible) modulation levels for each sub-carrier and available power levels for signal transmission. These sets are known from chipset specification or the standard protocol supported by the chipset. Note that this formulation can easily be extended to take into account different transmit power levels for each sub carrier. This case is not explored here because it would require multiple output amplifiers (one per sub-

[1] Obviously, more elaborate channel estimation techniques may be employed to improve the selection process, but this simple channel prediction scheme serves our purpose of illustrating the general approach.

carrier) in order to support independently controlled different power level per sub-carrier. This is quite expensive from implementation point of view.

The overall objective of the client-server game is to ensure that we achieve an acceptable level of performance while maximizing the overall video service time. Notice that the video service is terminated as soon as any one of the server or the client exhausts its energy source. For solving this optimization problem, the server and the client take turn at the beginning and end of each timeslot. The server's goal is to minimize the overall energy consumption of the client-server system whereas the client's objective is to make sure that it will not exhaust its energy source any sooner than the server does. In this way, this two-player game results in extending the overall system lifetime by first minimizing the energy consumption and then by ensuring that no one dies earlier than the other. Details are explained below.

The client (follower) uses the absolute value of the difference between its remaining lifetime at timeslot *j*, denoted by Λ_j^C, and the remaining lifetime of the server at timeslot *j-1*, denoted by Λ_{j-1}^S, as the cost function, i.e., $f_F(x,y) = \left|\Lambda_{j-1}^S - \Lambda_j^C\right|$. Notice that the client knows Λ_{j-1}^S as a result of the last data transmission. The client must therefore determine client parameters (i.e., *TL's* and f^C) so as to make Λ_j^C as close as possible to Λ_{j-1}^S.

The client must do this optimization under appropriate maximum power consumption and BER constraints (cf. Eqn. (14)). Maximum power constraint for the client can be written as:

$$e_{comm,j}^C + e_{comp,j}^C \le p_{\max}^C.\tau \Leftrightarrow \theta.\sum_{i=1}^{n} a_{i,j} + \eta^C.f_j^C + \varepsilon^C \le p_{\max}^C \tag{17}$$

where $a_{i,j}$ denotes the *TL* value used for the ith sub-carrier in timeslot *j*. Notice that the left hand side inequality in Eqn. (17) corresponds to the inequality constraint (I) of Eqn. (14). The right hand side inequality in Eqn. (17) is derived from Eqn's (7), (8), (10), and (12). θ is a constant coefficient that captures the effect of a_i*'s* on the power consumption of the Viterbi decoder (cf. Eqn. (12)). η^C is a constant coefficient that captures the effect of f^C on the power consumption of the decoding image processing core. ε^C is the constant that sums the constant values of Eqn's (7), (8), (10), and (12).

The client must also maintain a stable channel condition, i.e., it must satisfy a maximum BER constraint. This constraint however involves non-linear equations, which would make the optimization problem hard to solve. Therefore, a linear estimation of BER based on the modulation-level and truncation length for each sub-carrier is used as follows:

$$\alpha_{BER_i}.a_{i,j} + \beta_{BER_i}.b_{i,j} \le BER \tag{18}$$

where α_{BER_i} and β_{BER_i} are empirical coefficients for linear estimation of BER for the i^{th} sub-carrier in terms of the modulation level and the truncation length of the decoder. *BER* denotes the global BER requirement to receive image data correctly. Note that Eqn. (18) *implicitly* captures inequality constraint (II) of Eqn. (14). This is due to the fact that by upper bounding the BER in the channel, client makes sure that it can receive all of video data generated in the server side.

The optimization problem on the client side may be formally stated as follows:

$$\arg\min_{\hat{X}} \left\{ \left|\Lambda_{j-1}^S - \Lambda_j^C\right| : \Gamma\hat{X} + \Phi\hat{Y} \le R\hat{E}Q^C, \hat{X} \in TLS^n \times FS \right\} \tag{19}$$

Note that $\Gamma\hat{X} + \Phi\hat{Y} \le R\hat{E}Q^C$ represents the matrix-vector form of inequality constraints (17) and (18). Here, Γ and Φ denote the coefficient matrices that account for the channel conditions and per-frame energy consumptions of the basic building blocks of the client. $R\hat{E}Q^C$ is a vector consisting of an upper bound on maximum power consumption, $p_{\max}^C$, and the required *BER* value for all sub-carriers as shown below:

$$\Gamma = \overbrace{\begin{pmatrix} \theta & \theta & \cdots & \theta & \eta^C \\ \alpha_{BER_1} & 0 & \cdots & 0 & 0 \\ 0 & \alpha_{BER_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha_{BER_n} & 0 \end{pmatrix}}^{(n+1)\times(n+1)}$$
$$\Phi = \overbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \beta_{BER_1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \beta_{BER_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{BER_n} & 0 \end{pmatrix}}^{(n+1)\times(n+2)} \quad R\hat{E}Q^C = \begin{bmatrix} p_{\max}^C - \varepsilon^C \\ BER \\ BER \\ \vdots \\ BER \end{bmatrix} \tag{20}$$

Since the power consumption of the client can easily be derived from Eqn's (7), (8), (10), and (12), the client actually solves the following equivalent optimization problem :

$$\arg\min_{\hat{X}} \left\{ \left| \frac{E_j^C}{\Lambda_{j-1}^S} - \frac{e_j^C}{\tau} \right| : \Gamma\hat{X} + \Phi\hat{Y} \le R\hat{E}Q^C, \hat{X} \in TLS^n \times FS \right\} \tag{21}$$

The ratio of E_j^C to Λ_{j-1}^S signifies the power dissipation target for the client in order for it to survive until the end of the server's expected lifetime. e_j^C denotes the power consumption (i.e., per-frame energy consumption) of the client, which is calculated from the *TL's* and the f^C value by equation: $e_j^C = <\rho, \hat{X}> + \varepsilon^C$ where $\rho^T = \left[2^K P_{SMU} \ \ 2^K P_{SMU} \ \cdots \ 2^K P_{SMU} \ \ \eta^C \right]$ is a constant coefficient vector and $<a,b>$ is used to represent the inner product of vectors *a* and *b*. The client's objective in this optimization step is to find $\hat{X}^*$ such that its actual power consumption rate becomes as close as possible to its target power dissipation.

The server (leader), on the other hand, attempts to minimize the overall energy consumption of the client-server system, given the channel conditions provided by the client, hence $f_L(x,y) = e_j^C + e_j^S$. This optimization problem is solved with constraints similar to that of the client's. The maximum energy consumption rate is calculated following Eqn's (2), (3), and (4) as follows,

$$e_{comm,j}^S + e_{comp,j}^S \le p_{\max}^S.\tau \Leftrightarrow p_{amp,j} + \eta^S.f_j^S + \varepsilon^S \le p_{\max}^S \tag{22}$$

As in the client case, the bit error rate in the channel is upper-bound by using a linear equation as shown below:

$$\omega_i.p_{amp,j} + \delta_{BER_i}.b_{i,j} \le BER \tag{23}$$

where ω_i and δ_{BER_i} denote the BER estimation coefficient for sub-carrier *I*, respectively

The server also uses the minimum video quality constraint, $VQ_{\min}$, to find a lower bound on the channel rate for this timeslot. To do so, first the server adopts a target video quality for the frame at timeslot *j*, based on the total achieved video quality until then as reported by the client, and its own expected life time. This calculation is done as follows:

$$avg(VQ) = \frac{VQ_{j-1}^C + \Lambda_j^S.VQ_{tar,j}}{t + \Lambda_j^S} \ge VQ_{\min} \tag{24}$$

where VQ_{j-1}^C is the total video quality achieved at the client and reported to the server at timeslot *j-1*, and Λ_j^S is the expected remaining life time of the server at timeslot *j*. Given the requirement for average video quality $VQ_{\min}$ (cf. inequality constraint (II) of Eqn. (14)), the server solves Eqn. (24) to find the target average video quality, $VQ_{tar,j}$, for time slot *j*. Next, it uses this value to constrain the bandwidth and operating frequency of the encoding image processor using Eqn's (1) as follows:

$$VQ_{tar,j} \le k.R_s.\sum_{i=1}^{n} b_{i,j} \le \kappa_f.f_j^S + \varepsilon_{VQ} \tag{25}$$

The first term in Eqn. (25) describes the total bandwidth of the communication whereas the second term represents the encoding aptitude of the image processor.

Notice that the server must estimate its remaining battery lifetime at each timeslot *j*. A simple way to calculate Λ_j^S is to divide the remaining energy level of the server by its energy depletion rate. A key challenge is to accurately estimate the expected depletion rate of the server. Simply using the energy depletion rate of the previous timeslot, *j-1*, is not appropriate because it will not account for the long-term behavior of the node and may thus result in erroneous estimates. The approach we have taken is to calculate the (history-based) *aggregate energy depletion rate* of the server as a moving exponentially-weighted average so that the recent past has more influence, but the distant past is not completely ignored.

The optimization problem for the server can now be mathematically formulated as:

$$\arg\min_{\hat{Y}} \left\{ \langle \rho, \hat{X} \rangle + \langle \vartheta, \hat{Y} \rangle : \Xi\hat{Y} \le R\hat{E}Q^S, \hat{Y} \in PLS \times MLS^n \times FS, \hat{X} \in \Psi_L(\hat{Y}) \right\} \tag{26}$$

Similar to the client case, writing Eqn's (22), (23), and (25) in matrix-vector form would result in the coefficient matrix Ξ for linear estimation of the $p_{\max}^S$, *VQ*, and BER in terms of the SNR and the modulation level. Here ϑ is a constant coefficient vector $\vartheta^T = \left[1 \ 0 \ \cdots \ 0 \ \eta^S \right]$. $<\vartheta, \hat{Y}>$ signifies the power consumption (i.e., per-frame energy consumption, e_j^S) of the server. $R\hat{E}Q^S$ is a vector representing the maximum power consumption $p_{\max}^S$ and the minimum requirements for the *VQ* and BER.

$$R\hat{E}Q^S = \begin{bmatrix} p^S_{\max} - \varepsilon^S \\ -VQ_{tar} \\ -VQ_{tar} + \varepsilon_{VQ} \\ BER \\ BER \\ \vdots \\ BER \end{bmatrix} \quad \Xi = \overbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & \eta^S \\ 0 & -k.R_s & -k.R_s & \cdots & -k.R_s & 0 \\ 0 & 0 & 0 & \cdots & 0 & \kappa_f \\ \omega_1 & \delta_{BER_1} & 0 & \cdots & 0 & 0 \\ \omega_2 & 0 & \delta_{BER_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \omega_n & 0 & 0 & \cdots & \delta_{BER_n} & 0 \end{pmatrix}}^{(n+3)\times(n+2)} \tag{27}$$

Notice that when solving Eqn. (26), the optimization variables are $\hat{X}$ and $\hat{Y}$, that is, the server estimates the best strategy that the client may put in practice in response to the server's strategy and then based on this estimate of the client's policy, the server goes on to determine both the optimum values of the server and the client parameters. It will then implement the server policy, but will not report or in any way attempt to enforce the client policy. The client in its own turn will determine its optimum policy parameters as explained previously.

5 Experimental Results

We implemented an MPEG-4 FGS streaming system on a high performance testbed [20]. The processing core in this testbed is the Intel's Xscale processor, which supports nine different frequencies from 200MHz to 733MHz. A D/A converter was used as a variable operating voltage generator to control the reference input voltage to a DC-DC converter that supplies operating voltage to the CPU. Inputs to the D/A converter were generated using customized CPLD logic. When the CPU clock speed is changed, a minimum operating voltage level should be applied at each frequency to avoid a system crash due to increased gate delays. In our implementation, these minimum voltages are measured and stored in a table so that these values are automatically sent to the variable voltage generator when the clock speed changes. Voltage levels mapped to each frequency are distributed from 0.9V @200MHz to 1.5V @733MHz. For the software work, Microsoft reference MPEG-4 FGS encoder/decoder was modified to fit our purpose. Two generated bit-streams of QCIF video sequence with 150 frames, a base layer and a FGS enhancement layer with 5 bit-planes (bp0~bp4), are split into packets with size of 256-byte. RTP/RTCP on UDP was used as a network protocol between the server and the client. Both the server and the client were equipped with an IEEE 802.11b wireless LAN (WLAN) card. Energy consumption of the WLAN interface was measured by using a data acquisition (DAQ) system.

In order to simulate the system, the Simulink 5.0 environment from Matlab 6.5 Release 13, was used. To model a multi-path fading channel, a parallel combination of Rayleigh and Rician fading propagation channels was used (see [16] for details about these channels.) The maximum Doppler shift and the spreading factor of the Rician fading channel were set to 40Hz and 1, respectively.

To account for the effect of multi-path fading, three different paths with delays of 2us, 3us, and 5us and gains of –3, 1, and 2 were considered in Rayleigh propagation channel. The characteristics of this channel were simulated and recorded for duration of 4800 frames, i.e., five minutes @15 frames/second. To produce the channel probability distribution, $h(\omega_i)$, this data is used. Then these probability values are fed into the Stackelberg game for policy design. To show the effectiveness of our approach three different scenarios were simulated and compared with each other. In scenario number one, for each timeslot *i*, after the detection of channel conditions we assigned enough energy to the server and the client to support the specified average video quality.

In scenario number two, average channel condition was used to determine the required energy in each timeslot, and finally in scenario three our adaptive algorithm was used to calculate the required energy for each timeslot.

Figure 3 shows the lifetime versus required average video quality graphs for scenarios 1 and 3. Total initial energy for this experiment is set to 600 J. According to this graph, our approach increases the system lifetime by as much as 20% for high values of the required video quality. On average, the proposed method increases the system lifetime by more than 15% over the whole range of video qualities.

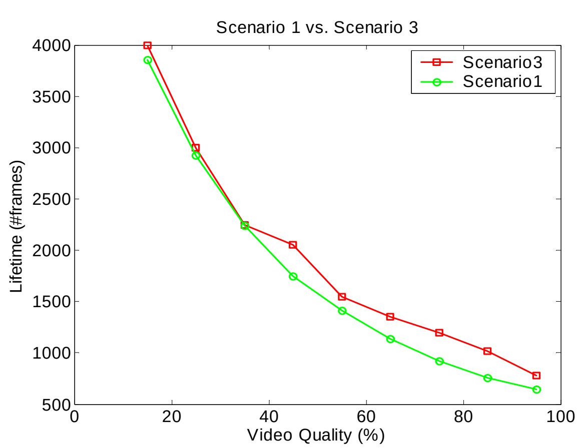
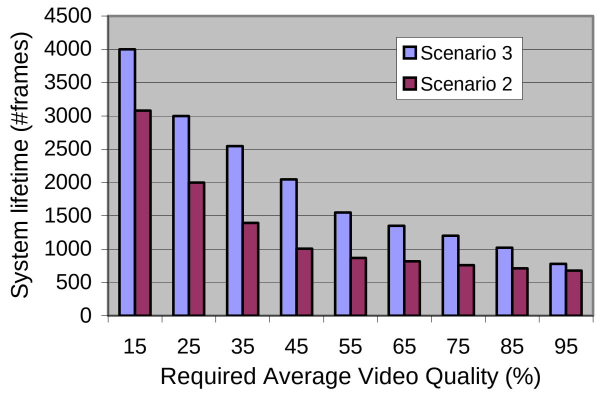
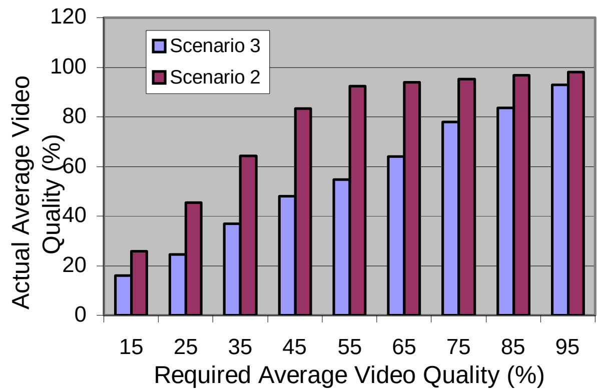


Figure 3. Lifetime comparison between Scenario 1 and 3



(a) System lifetime



(b) Video quality

Figure 4. Comparisons between Scenarios 2 and 3

Figure 4 shows the comparison between scenarios 2 and 3. Since in scenario 2, the assigned energy to each frame is fixed, and is selected according to the average channel behavior, the system consumes extra energy to produce higher video quality, and hence, has a lower system lifetime. Figure 3(a) shows comparison between the system lifetimes for scenarios 2 and 3. It is clear that system lifetime is significantly increased for scenario 3 where we employed the proposed dynamic policy approach. Notice that the average video quality was maintained above the required value. However, for scenario 2, the average video quality is unnecessarily improved, which may be OK if there no was energy dissipation overhead (cf. Figure 3(b)).

6 Conclusion

In this paper, the energy consumption of a mobile video streaming system was modeled. Using this model, an adaptive approach for energy assignment to each frame was developed. The proposed approach guarantees the minimum video quality for all frames and meets a required average video quality over the system lifetime. Actual experimental data was used to extract parameters of the proposed model. Based on these parameters, simulations were setup to demonstrate the effectiveness of the proposed approach. It is rather straightforward to extend the proposed approach to multi-hop routing networks, where the video information is received from a mobile/stationary host and is relayed through some intermediate nodes in a mobile ad-hoc network to reach the target node. More precisely, the multi-hop scenario can be easily handled by extending the two-level hierarchical game to a multi-level hierarchical game.

References

[1] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," IEEE Trans. On Circuits and Systems for Video Technology, Vol.11, No. 3, Mar. 2001.

[2] M. van der Schaar, H. Radha, and C. Dufour, "Scalable MPEG-4 Video Coding with Graceful Packet-loss Resilience over Bandwidth-varying Networks," Proc. of the ICME, vol.3, pp.1487-1490, 2000.

[3] R. Cohen and H. Radha, "Streaming Fine-Grained Scalable Video over Packet-Based Networks," Proc. of the GLOBECOM. IEEE, pp.288-292, 2000.

[4] R. Yan, F. Wu, S. Li, and R. Tao, "Error resilience methods for FGS video enhancement bitstream," The First IEEE Pacific-Rim Conference on Multimedia, Dec. 13-15, 2000 Sydney, Australia.

[5] T. Pering, T. Burd, and R. Broderson, "The simulation and evaluation of dynamic voltage scaling algorithms," Proc. of Int'l Symp. on Low Power Electronics and Design, pp.76-81, 1998.

[6] J. Proakis, Digital Communications, McGraw-Hill, 3rd Edition, 1995.

[7] ------, "A Game Theoretic Approach to Dynamic Energy Minimization in Wireless Transcievers," to appear in Proc. of Int. Conf. on Computer Aided Design, 2003.

[8] http://www.transmeta.com/

[9] K. Choi, K. Kim, and M. Pedram, "Energy-aware MPEG-4 FGS Streaming," Proc. of the Design Automation Conference, pp.912-915, 2003.

[10] S. Appadwedula, M. Goel, N.R. Shanbhag, D.L. Jones, and K. Ramchandra, "Total System Energy Minimization for Image Transmission," Journal of VLSI Signal Processing Systems Feb. 2001.

[11] Vetro, A., Christopoulos, C., Huifang Sun, "Video transcoding architectures and techniques: an overview," Signal Processing Magazine, IEEE , Volume: 20 , Issue: 2 , March 2003, Pages:18 – 29.

[12] R. Henning and C. Chakrabarti, "Low-power approach to decoding convolutional codes with adaptive viterbi algorithm Approximations," in Proc. of Proc. of Int'l Symp. on Low Power Electronics and Design, pp. 68 –71, Aug. 2002.

[13] F. Chan and D. Haccoun, "Adaptive Viterbi Decoding of Convolutional Codes over Memory less Channels," IEEE Trans. on Comm., Vol. 45, No. 11, pp. 1389-1400, Nov. 1997.

[14] S. Swaminathan, R. Tessier, D Geockel, and W. Burleson, "A dynamically Reconfigurable Adaptive Viterbi Decoder," in Proc. of the FPGA Conf., Monterey, California, Feb. 2002.

[15] C. F. Lin and J. B. Anderson, "M-Algorithm Decoding of channel convolutional Codes," in Proc. of Princeton Conference of Information Sci. and System, pp 362-366, Princeton, NJ, Mar. 1986.

[16] Wireless propagation bibliography, http://w3.antd.nist.gov/wctg/manet/wirelesspropagation_bibliog.html

[17] Stephan Dempe, Foundations of Bilevel Programming, Kluwer Academic Publishers, Boston, 2002.

[18] H.v. Stackelberg, "Marktfrom und Gleichgewicht", Springer-Verlag, Berlin 1934. engl. Transl. The theory of the Market Economy, Oxford University Press, 1952.

[19] T.M. Cover, J.A. Thomas, Information Theory, 2nd Edition , John Wiley & Sons Inc., New York, 1991.

[20] http://apollo.usc.edu/testbed/