Cross-layer Quality Assessment of Wireless Video Transmission over Mobile Broadcast Networks

Kyungtae Kang, Won J. Jeon, Kyung-Joon Park
Roy H. Campbell, and Klara Nahrstedt

***Abstract*—The recent development of high-speed data transmission over wireless cellular networks has enabled the delivery of multimedia broadcasting services to mobile users. These services involve a range of interactions among different system components, including the wireless channel, the network, and mobile devices, making it crucial for the service provider to verify the model, design, and behavior of a new service before it is deployed. However, previous studies have largely relied on network simulations or scaled experiments, and there has been little work on the sort of unified framework for quality-of-service (QoS) assessment, which considers the interactions between components, that we propose in this paper. Accurate models of the wireless channel, the network, and the data processing that takes place on a client device, are integrated within our framework, and allow us to predict several key system metrics and the quality of the video stream as it is perceived by users. Furthermore, different models of system components can be easily plugged in to extend this framework. As an example application, we analyze the performance of the process of decoding scalable videos on mobile devices in CDMA2000 wireless cellular networks.**

***Index Terms*—Video broadcasting, quality assessment, cross-layer, wireless cellular networks.**

I. Introduction

The recent development of high-speed data transmission over wireless cellular networks has enabled a range of video broadcasting services, but these require a higher data-rate than earlier services such as web browsing and email. As the network data-rate increases, the timely processing of video streams by mobile client devices becomes more important than preserving the full integrity of the data streams.

Multimedia broadcasting services over wireless cellular networks involve many interactions among different system components, including the wireless channel, the network, and client devices. Therefore it is crucial for the service provider to verify the system model, the design, and its behavior before a new type of service is deployed. However, it is not easy to predict how all the system components will interact. Different wireless operators and device manufacturers have their own specifications for production and maintenance, and it is common to observe unexpected malfunctions and loss of performance. Due to the complexity and size of these systems, the prototyping or temporary deployment of a real system is not usually considered to be feasible.

A preliminary version of this paper will appear in Proceedings of the 29th International Conference on Distributed Computing Systems (ICDCS 2009).

The authors are with the Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801, USA. (email: {ktkang,wonjeon,kjp,rhc,klara}@illinois.edu).

The difficulties of ensuring that system components are fully compatible before deployment has been addressed in various contexts. The use of network simulation tools such as OPNET [1] and ns-2 [2], simulated network models [3], [4], or mimicking the network system with scaled emulations [5], can provide an approximate assessment of a particular network. However, these approaches are not the same as an analysis of the entire system, including data processing and the quality delivered by applications running on mobile devices.

In this paper, we propose a holistic cross-layer quality-of-service (QoS) assessment framework for video broadcasting services over wireless cellular networks. Taking this approach is of fundamental importance since it should not only improve the accuracy of performance evaluation, but may also provide valuable insights leading to the design of new protocols. Our framework involves a precise model of a real system architecture and can be used to verify models. By simulating the physical wireless channel, the network, and the data processing that takes place on a client device, the system throughput and so-called *goodput* (which is a user-oriented metric especially for video streams) can be predicted.

Our main contributions are as follows: First, we provide a flexible QoS assessment framework for video broadcast services in wireless networks, which can easily be extended to other scenarios with different network and device models. This framework covers the whole system stack, including the wireless physical channel and network, the processing power of the device, and the quality assessment of applications. Second, we analyze the system and provide a guideline for the performance required from a client device in terms of error correction (i.e., the error-recovery rate, the delay jitter, and the perceived quality of video streams). This analysis relates specifically to CDMA2000 networks, and is based on a range of channel parameters, such as the signal-to-noise ratio (SNR) and the mobile speed, and as well as on the service parameters, such as the type of error-correction scheme.

The rest of our paper is organized as follows. In Section II, we abstract the requisite network architecture and protocols, and introduce our framework for the assessment of quality in video broadcasting services. In Section III, we describe the different network and system modules in the framework in detail. In Section IV, we analyze key QoS parameters and their simulation. In Section V, we show how this framework can be used to analyze scalable video decoding on mobile devices in CDMA2000 wireless networks. We discuss other aspects of our framework in Section VI, and finally conclude this paper in Section VII.

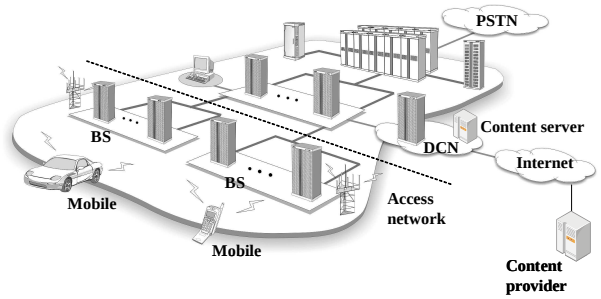


Fig. 1. Network architecture for a video broadcasting service.

II. Overview

We will review the network architecture and protocols used to provide a video broadcasting service over wireless cellular networks. We will also introduce the structure of our simulation and analysis framework.

A. Video transmission over mobile broadcast networks

Fig. 1 shows a generic cellular network architecture to support video broadcast services, which is composed of a RAN (radio access network) and a DCN (data core network). The broadcast content provider, who can be located in the serving or home network, or anywhere in an IP network such as the Internet, makes broadcast content available within an IP stream. The content server may store and forward content from a provider, or merge content from multiple content providers. Finally the base station (BS) transmits a video stream to a number of mobile stations over the wireless channel. We will focus on this forward link and assume the forward traffic channel is time-multiplexed between the different access mobiles.

A wireless mobile device is equipped with a radio modem and a data interface, allowing it to access time-division multiplexed channels. Fig. 2 shows the generic protocol layers and the interactions between them that are required for reliable data communication. The broadcast security protocol encrypts content to cope with the threat of a user obtaining unauthorized access to a particular content stream. The broadcast data-link protocol encapsulates the physical layer to create a link responsible for node-to-node communications, and provides

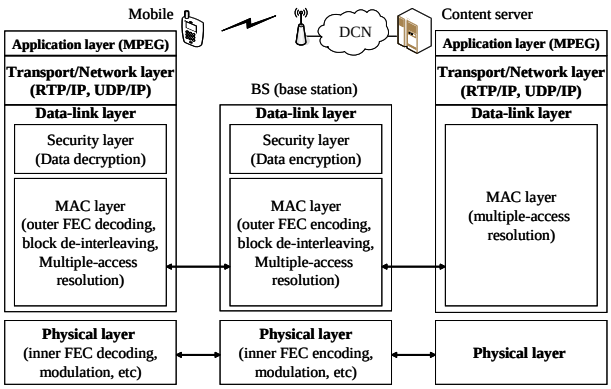


Fig. 2. Generic network protocol layers used in a video broadcasting service.

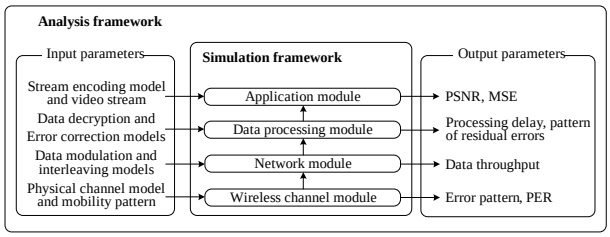


Fig. 3. The simulation and analysis framework.

error control and medium access control (MAC) for reliable communications. In general, ARQ (automatic repeat request) cannot be used to correct damaged packets in a broadcasting service, because (1) there is no reverse channel for ARQ, (2) a storm of NAKs (negative acknowledgments) will occur when a channel condition is bad, and (3) it does not meet the real-time constraints of video applications. Reliability is increased by using the outer FEC (forward error correction) code and the block interleaver to rearrange code symbols so as to spread bursts of errors over multiple codewords to make error correction codes more effective. The broadcast physical layer provides the structure of the broadcast channel.

B. Simulation and analysis framework

As shown in Fig. 3, each module in the simulation framework has a corresponding model which allows it conceptually to mimic the operations of the appropriate components of a wireless broadcasting system. The lower-level system modules feed information to the upper-level modules, which allows the individual analysis of each module, as well as an analysis of the whole system. The following modules are incorporated in the simulation framework:

- Wireless channel module: The wireless channel model proposed by Zorzi *et al.* [6]–[8] uses Markov approximations for the block error process that occurs in fading channels, with a range of modulation schemes. This represents the behavior of block errors at different mobile speeds more precisely than other additive white Gaussian noise (AWGN) models [9]. It also provides a good approximation to the relationship between the probability of successful transmission, the steady-state packet error-rate (PER), and the mobile speed. The PER is strongly affected by the SNR of the channel and the implementation of inner FEC coding in the physical layer.
- Network module: The wireless network model is based on TDM (time-division multiplexing) technology, which enables the deterministic delivery of video data to multiple mobile devices.
- Data processing module: Data processing at the mobile device mainly takes place in the data-link layer, and involves correction of the errors that occur during the transmission of data across a fading channel, and content protection to restrict the reception of broadcast services to authorized users. The error correction scheme that we model is the Reed-Solomon (RS) FEC scheme, and the encoded data is aligned by block interleaving [12]. Data

encryption and decryption are performed by an AES (Advanced Encryption Standard [13]) block cipher. (If link-layer encryption is used, the content-encryption function is typically located in the RAN. If higher-layer encryption is used, content encryption takes place in the content server.) This module simulates RS decoding for different levels of block interleaving and coding parameters, and AES block decryption.

- Application module: A model of scalable video decoding is used to determine the quality of video streams perceived by users, for a given system throughput and pattern of packet loss in the underlying network layer.

While the whole system is being monitored, probes can also be inserted into individual network and service components. This analysis framework allows detailed monitoring throughout the simulation period. The system also checks the compatibility of parameters across simulation modules, and translates them if necessary.

III. Proposed system model for video broadcasting

We will now provide more detail about the system models introduced in the previous section, represent the physical channels, the wireless networks, data processing operations, and the decoding of scalable video streams on mobile devices.

A. Modeling wireless channels

Zorzi *et al.* investigated the behavior of the block errors which arise in data transmission over fading channels, and showed that a Markov chain is a good approximation to a range of modulation schemes, block lengths, and error correction capabilities.

They also showed that the transition probabilities in Markov models are largely insensitive to parameters such as the rate of inner FEC coding and the modulation format, and depend only on the PER and the Doppler frequency normalized to the block transmission rate, V, which can be derived as follows:

$$V(f_c, v, L_{phy}, \mu_p) = f_D L_{phy}/\mu_p, \tag{1}$$

where L_{phy} is a size of a physical-layer packet, μ_p is the reference data-rate of the physical channel, and f_D is the Doppler frequency of the system, which equals $f_c v/c$, where v is the mobile speed, c is the speed of the electromagnetic wave, and f_c is the carrier frequency [14]. This relationship is, in fact, almost the same as that in the simple threshold model, for which closed-form analytical expressions are available [6], [8] for the Rayleigh fading case. This observation allows each transition probability to be obtained as a function of PER and V.

B. Wireless network model

Fig. 4 is a high-level abstraction of a forward link channel based on the TDM structure with η sub-channels, which we have generalized from competing wireless technologies. This multiple-access structure enables the timely delivery of video data by assigning a dedicated sub-channel to each mobile. If

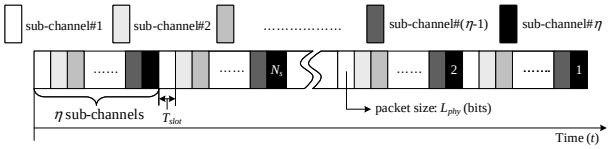


Fig. 4. Abstraction of the TDM structure.

the length of each time-slot is T_{slot} and that the transmission of one encoded physical-layer packet takes exactly one slot, the size L_{phy} of each physical-layer packet of a particular sub-channel k can be determined as follows:

$$L_{phy} = \mu_{p_k} T_{slot}, \tag{2}$$

where μ_{p_k} is the data-rate of that sub-channel. When there are η sub-channels, the total data-rate of the forward data channel μ_p is given as follows:

$$\mu_p = \sum_{k=1}^{\eta} \mu_{p_k}. \tag{3}$$

This data-rate is determined by the modulation, the coding rate of the inner FEC code in the physical layer, and the spreading used in each sub-channel. For example, a CDMA2000 1xEV-DO system supports a range of channel data-rates between 38.4kb/s and 2457.6kb/s. The rate depends on: the modulation, which may be QPSK (quadrature phase-shift keying), 8-PSK (phase-shift keying) or 16-QAM (quadrature amplitude modulation); the code-rate (1/5 or 1/3); and the number of slots scheduled for each physical-layer packet [10], [11].

C. The model of data processing on the device

1) Physical layer: Channel coding and modulation in the physical layer are performed on the information-bearing digital signal, with the aim of optimizing the performance of the communications system. While modulation is concerned with packing bits in a chip which is the fundamental unit of transmission in a CDMA system, channel coding is the packing of data in a slot. Channel coding can cope with bits that are flipped due to noise and interference by employing *smart redundancy*. The most significant coding techniques for implementing redundancy are error control coding, using inner FEC codes such as convolutional and turbo codes, and interleaving. Typically, turbo coding is more powerful than convolutional coding when the physical-layer packets are large (i.e., several hundred bits or more). In this case, turbo coding significantly improves performance by allowing the use of a lower RF (radio frequency) power while still achieving the same error-rate. This is so effective that the communication channel is able to perform close to the Shannon limit [10], [15], [16].

Additionally, concatenated error-correcting codes, in which an inner code is combined with an outer FEC code, improve the performance of error correction. To optimize the performance of this form of coding, we need to allow for the characteristics of the channel through which transmission takes place, so as to select the most suitable channel coding and modulation techniques for a given application.

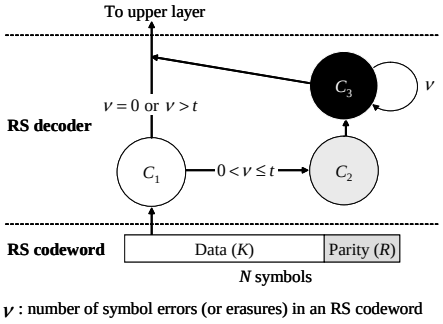


Fig. 5. Computational components of the RS decoding process.

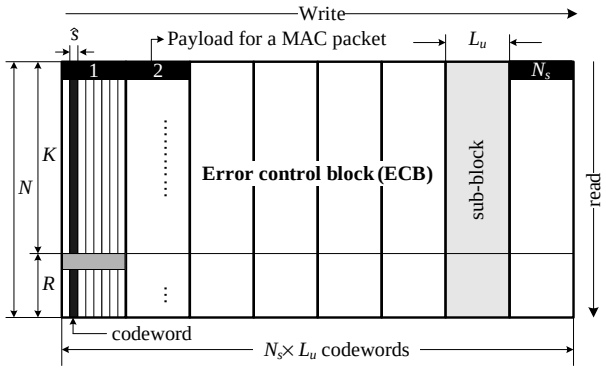


Fig. 6. The block interleaving scheme used in an ECB.

2) MAC layer: The MAC protocol provides error control, as well as a method of medium access, by adding an outer FEC code that forms a concatenated code in conjunction with the physical-layer inner FEC code. This outer code provides additional data protection, and RS codes are a good choice for the outer code because of their superior performance at lower error-rates [17], [18].

An RS code is specified as a tuple (N, K, R) with $\hat{s}$-bit symbols. This means that the encoder takes K data symbols of $\hat{s}$ bits each and adds R parity symbols of $\hat{s}$ bits each to make an N-symbol codeword. Reed-Solomon is therefore known as a systematic code because the appended parity symbols leave the data unchanged. The algebraic RS decoder can correct up to $t = R$ *erasures* when the positions of the erroneous symbols are known or up to $t = R/2$ *errors* when these positions are unknown. Erasure coding outperforms error correction coding because more information is available. Large values of R mean that more erasures or errors can be corrected, but more processing power will be required because the amount of computation needed to encode and decode RS codes increases with the number of parity symbols in each codeword.

The RS decoding algorithm, which we have analyzed in detail elsewhere [19], consists of three computational components, C_1, C_2, and C_3, as shown in Fig. 5. C_1 operates on every codeword received, C_2 is called once for each codeword which contains any errors, and C_3 is invoked to deal with each error in that codeword. The total execution time T_{cw} required for the decoding of an RS codeword, encoded by an (N, K, R) code with $\hat{s}$-bit symbols and containing ν erroneous symbols, can be estimated by summing the time required by the components C_1, C_2 and C_3:

$$T_{cw}^{(\hat{s},N,K)}(\nu) = \begin{cases} T_{C_1}^{(\hat{s},N,K)} + T_{C_2}^{(\hat{s},N,K)} + \nu T_{C_3}^{(\hat{s},N,K)} & (\text{if } 0 < \nu \le t) \\ T_{C_1}^{(\hat{s},N,K)} & (\text{otherwise}), \end{cases} \quad (4)$$

where $T_{C_i}^{(\hat{s},N,K)}$ is the execution time of component C_i.

This RS coding scheme is usually combined with block interleaving. In a wireless mobile channel environment, error bursts occur frequently, and interleaving allows an error-correcting code to function more effectively, although interleaving itself has no intrinsic error-correcting capability. The basic idea is that data block is entered into a two-dimensional array from the top to the bottom, with each row written from left to right. The block is then read out in columns from left to right and from top to bottom, as shown in Fig. 6.

In order for block interleaving to work properly, a special region of memory, which is called an error control block (ECB), is required to buffer data at both the sender and receiving receiver. Fig. 6 shows the structure of the ECB, which is a matrix with N rows and $N_s \times L_u$ columns, where N_s is the number of MAC-layer packets in each ECB row, and L_u is the size of the payload in each MAC-layer packet. A sequence of RS codewords, each $\hat{s}$ bits wide and encoded with the outer RS code (N, K, R), spans the columns of N_s sub-blocks, each of which is L_u bits wide. An ECB sub-block therefore contains $N_{cw} = (L_u/\hat{s})$ codewords. The value of N_s determines the level of block interleaving. As that increases, the error bursts which inevitably occur during data transmission are interleaved more effectively, so as to favor RS decoding and reliable communication.

Each of the N rows is then individually encoded by the outer code and sent over the air to multiple mobiles. The data is transmitted using MAC-layer packets, and the access network concatenates trailer bits to the payload of each MAC-layer packet, to form a complete MAC packet of size L_{mac}. The resulting packet is then forwarded to the physical-layer protocol for transmission. A single physical-layer packet of size L_{phy} contains $z = \lfloor L_{phy}/L_{mac} \rfloor$ MAC-layer packets ($\lfloor\ \rfloor$ is a floor function).

3) Security layer: Encryption of the broadcast content and distribution of the decryption key to authorized subscribers to the channel is used to counter the threat of pirate users obtaining free access to the content. We focus on the use of AES encryption procedures, operating in the security layer within the link layer. The AES algorithm is a symmetric block cipher that can both encrypt and decrypt information. We are only concerned with deciphering, because it is the operation that needs to be performed in a mobile that is receiving a broadcast service.

The AES algorithm operates on a two-dimensional array of bytes called the *state*. This consists of four rows, each containing N_b bytes, where N_b is the block length divided by 32. The length of the cipher key can be 128, 192, or 256 bits. The key length N_k is 4, 6, or 8, which corresponds to the number of 32-bit words (the number of columns) in the cipher key. The number of rounds N_r that have to be performed

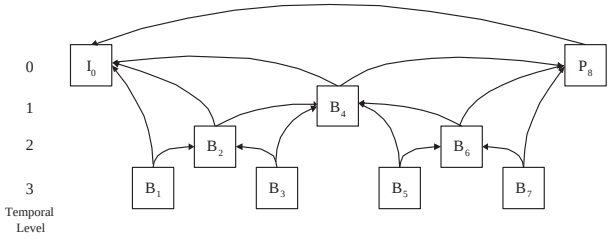


Fig. 7. Hierarchical structure of H.264/SVC frames in a video stream.

during the execution of the AES algorithm depends on the key size, so that $N_r = N_k + 6$.

At the start of the decipher, the input is copied to the state array. After an initial round-key (RK) addition, this array is transformed by performing the round function 10, 12, or 14 times, depending on the key length, with the final round differing slightly from the first $N_r - 1$ rounds, as explained elsewhere [13]. A round function is composed of four different byte-oriented transformations, as follows:

- Sub-bytes (SB): a non-linear byte substitution that operates independently on each byte of the state using a substitution table (S-box).
- Shift-rows (SR): the bytes in the last three rows of the state are cyclically shifted over different numbers of bytes (offsets).
- Mix-columns (MC): data is exchanged between columns of the state array.
- Add-round-key (ARK): a round key is added to the state by a bitwise XOR operation.

The AES decryption algorithm also includes a key expansion (KE) routine which generates a key schedule for each new value of the cipher key. Thus the execution time $T_{IC}^{N_b,N_k}$ required to encrypt a data block containing $4 \times N_b$ bytes with a cipher key of $4 \times N_k$ bits can be determined as follows:

$$\begin{aligned} T_{IC}^{N_b,N_k} =& T_{KE}^{N_k} + N_r(T_{ISR}^{N_b} + T_{ISB}^{N_b} + T_{ARK}^{N_b}) \\ &+ (N_r - 1)T_{IMC}^{N_b} + T_{ARK}^{N_b}, \end{aligned} \quad (5)$$

where T_A^B is the execution time of transformation A with the parameters B.

D. Application model for scalable videos

Typically, video stream is encoded for efficiency in such a way that successive frames share data and are therefore mutually dependent. This means that the quality of a video perceived by a user is degraded more than one might expect by the loss of a single frame during transmission over a wireless network, because dependent frames will also be damaged. Several different methods of error concealment have been suggested to deal with this so-called *cascading effect*. The dependencies among video frames and error concealment methods in use must be considered in evaluating the perceived quality of video streams for given wireless channel and network conditions.

H.264/SVC [20] was suggested and developed as an extension of H.264/AVC, and is currently recognized to be the most efficient video encoding scheme. SVC streams have a hierarchical B-frame structure which enables higher coding efficiency than non-hierarchical coding schemes such as AVC. H.264/SVC allows temporal scalability, because B-frames at lower levels can be discarded to generate lower-resolution video streams. Fig. 7 shows a typical configuration of GoPs (group of pictures) in the base layer of H.264/SVC, with a GoP size (G) of 8. The quality of decoded frames at the receiver is naturally affected by this hierarchical structure and also by the picture copy concealment technique, which is used to hide errors.

Mansour *et al.* [21] analyzed the hierarchical structure and error concealment scheme in H.264/SVC video streams and suggested the following model[1] to relate quality degradation (D) to frame losses:

$$D(p_{net}) = p_{net} \times \frac{E[D_{I/P}] + (G-1)E[D_B]}{G}, \quad (6)$$

where

$$\begin{aligned} E[D_{I/P}] =& [(1-p)\triangle_G + 2p\triangle_G] \times G \\ &+ \frac{1}{\gamma}\sum_{j=0}^{M-1}\sum_{i=1}^{M-j-1}\triangle_G(1 - \frac{i}{M-j}) \times G, \end{aligned} \quad (7)$$

and

$$E[D_B] = \sum_{k=1}^{\log_2(G)} \frac{2^{k-1}}{G-1} D_{B_k}. \quad (8)$$

Here, p_{net} is the overall probability of frame loss, p is the probability that the previous key frame (i.e., the I or P frame) is lost, $E[D_{I/P}]$ and $E[D_B]$ respectively are the expected quality degradation, expressed as an MSE (mean square error), when key frames and B-frames are lost. G is the size of a GoP, $\triangle_G$ is the quality degradation due to concealment mismatch between two frames that are G pictures apart, γ is the relative frequency of the occurrence of I frames and P frames in a GoP, and D_{B_k} is the distortion due to the loss of frame B_k, where k is the temporal level of that frame.

IV. Cross-layer simulation and performance analysis of wireless video transmission

We will now describe how our framework is implemented and used to analyze the QoS parameters of wireless video transmission.

A. Simulation framework

Our analysis tool is written in Java, and its structure is shown in Fig. 8. The tool runs with parameters that describes the physical channel, the network, and the video broadcasting service. Once this input has been received, the input control module passes the SNR measure of the forward data channel, the code-rate r, and the physical-layer packet size to a simulator of the inner decoding process[2], which derives the

[1]Note that we consider packet losses to occur in the base layer only. In reality, there could be a drift effect due to losses in the enhancement layers. This could be modeled in a similar manner to the base layer losses, by scaled quality compensation.

[2]We will not consider the timing aspect of inner FEC coding because it is implemented typically in hardware and introduces little overhead.

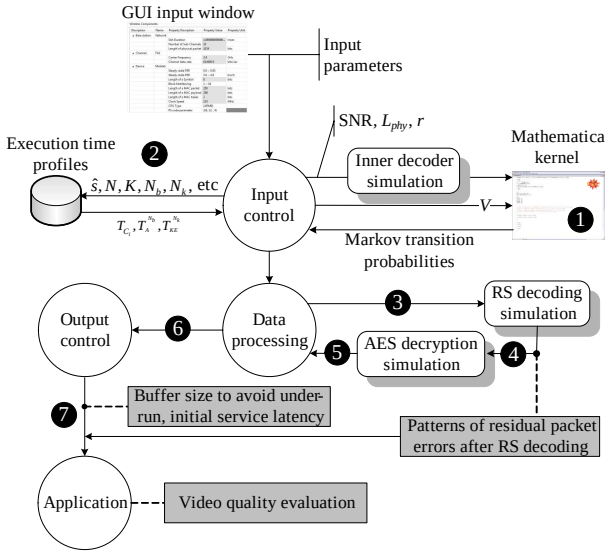


Fig. 8. Structure of the simulation analysis.

corresponding PER for the channel. This PER, together with the correlation properties of the fading process, representing the dependency between the transmission of consecutive blocks, is passed to the Mathematica [22] kernel through J/Link [23], which is a toolkit that connects Mathematica and Java. Mathematica code is used to calculate the Markov state-transition probabilities of consecutive packets by applying model of the wireless fading channel, which is described in Section III-A. These Markov state-transition probabilities can then be used to approximate the behavior of the packet errors in the physical layer which arise in data transmission over fading channels. Thus the input control module describes the functionalities of the wireless channel module and the network module, which were introduced in Section II-B.

The analysis tool then obtains profile data for the execution time $T_{C_i}^{(\hat{s},N,K)}$ required by the three computational components that are involved in decoding a codeword. This execution time depends on the symbol length $\hat{s}$ of the RS codeword and the RS code in use. Then it obtains profile data for the execution time $T_A^{N_b}$ required by the transformation A performed by the AES decryption algorithm, and the time $T_{KE}^{N_k}$ required by the key expansion routine of the example software implementation of AES block decryption. $T_A^{N_b}$ depends on the length of the AES cipher block, while $T_{KE}^{N_k}$ depends on the length of the cipher key. This profile also dependent on the computational power of the hardware platform that is executing the AES decryption code and the RS decoder. We created several profiles by measuring the time required by each computational component of the RS decoder for different RS codes, and the time required by all the transformations involved in AES decryption for different values of N_b and N_k, in execution environments based on different processors.

Next, the data processing module uses Monte-Carlo simulation to model the consecutive stages of MAC-layer RS decoding and security-layer AES decryption, which interact with each other across the layers. This simulation uses the timing profile data described above and a stochastically generated pattern of errors in the physical-layer packets which matches the expected occurrence of errors during transmission over a fading channel, which is obtained using the state-transition probabilities. A Monte-Carlo approach using repeated random sampling can obtain reliable value for the decoding time of each ECB, as well as the pattern and number of the residual errors in each ECB after RS decoding, for a given channel condition, provided that the number of iterations is sufficiently large. The time required for AES decryption can then also be determined.

The results of the simulation decided above are forwarded to the output control module, which estimates the size of buffer needed to avoid service dropout due to buffer underrun, which would cause significant interruption in video streaming. The buffer must be large enough to contain the amount of data needed to accommodate the worst-case execution-time (WCET) for the consecutive operations of RS decoding and AES decryption in processing a single ECB. The resulting initial service latency can now also be determined, and finally the application module is invoked to determine how the perceived quality of the final video is affected by the end-to-end delivery of data from the service provider to the application running on the mobile device.

Physical-layer packets are processed by a mobile device, and the resulting payload is fed into the application buffer for display. If some packets do not arrive at the application buffer in time, so that they do not meet the timing constraint imposed by the display rate, they are regarded as missed. An error concealment algorithm within the application minimizes the effect of such misses on the video as it appears on the screen. The quality perceived by a user must be determined in the context of an appropriate distortion model (6) for scalable videos.

B. Estimation of the buffer size required to avoid service interruption

Estimating the WCET required by the module that performs RS decoding and AES decryption at the mobile is critical in determining an adequate size of buffer to absorb jitter and provide a seamless video service. Recall that our timing model (5) predicts that the execution time required for RS decoding will increase with the number of erasures ν in an RS codeword, up to R, because C_3 must run more frequently than the other components of the decoding process. But the time required for AES decryption is not greatly affected by the number of erasures, provided that the RS decoder is able to correct them. This means that the WCET for processing an ECB at the mobile increases as the channel condition deteriorates. We have also shown [24] that the WCET is relatively long in a slow fading environment. Let us also assume that the value of SNR within the service area covered by a particular BS, with a code-rate of r, is larger than or equal to ξ_{min}, and that the mobile speed v is in the range v_{min} to v_{max}. In this case the WCET occurs when the channel condition, which can be expressed as a pair made up of SNR and V, is Φ_{WC}, which corresponds to the tuple $(\xi_{min}, V(f_c, v_{min}, L_{phy}, \mu_p))$. We can determine the WCET for a particular level of block

interleaving that satisfies a $\Upsilon\%$ confidence limit by Monte-Carlo simulation.

Providing a seamless service is critically important for video applications. We therefore need to absorb the jitter caused by the variation in data processing required at the mobile by buffering the data for the period equal to the WCET. The total size of buffer required at the data-link layer L_{buf}, and the service latency T_L, when the level of block interleaving is N_s, can then be determined as follows:

$$L^N_{buf}(\eta, N_s|\Phi_{WC}, \Upsilon) = \overline{\mu_u} T^N_L(\eta, N_s|\Phi_{WC}, \Upsilon), \quad (9)$$

$$T^N_L(\eta, N_s|\Phi_{WC}, \Upsilon) = T^N_p(\eta, N_s) + T_{WC}(N_s|\Phi_{WC}, \Upsilon), \quad (10)$$

where $\overline{\mu_u}$ is the data-rate of the MAC payload in a particular sub-channel (the effective data-rate of the MAC payload excluding parity information $\overline{\mu_e}$ is $\overline{\mu_u} \times \frac{K}{N}$), T_p is the period required to fill each ECB with its MAC payload when there are η sub-channels, T_{WC} is the WCET, and Υ is the confidence limit. These quantities can be determined as follows:

$$\overline{\mu_u} = \frac{zL_u}{\eta T_{slot}}, \quad (11)$$

$$T^N_p(\eta, N_s) = \frac{NN_sL_u}{\overline{\mu_u}} = \frac{\eta N N_s T_{slot}}{z}. \quad (12)$$

V. Example QoS analysis of scalable video broadcasting over a CDMA2000 network

We now present our experimental analysis of scalable video broadcasting over a CDMA2000 network. The 3GPP2 recently baselined the specification for a CDMA2000 high-rate broadcast packet data air interface [25], [26], which allows the high-speed delivery of packets to multiple-access terminals [27]. The air interface of this packet data system comprises a group of protocols collectively called the broadcast protocol suite [26]. The form of a transmission begins with framing protocol, which is used to fragment higher-layer packets at the access network, and specifies how the access terminal determine higher-layer packet boundaries. Next, the broadcast security protocol specifies how framing packets are encrypted using the AES block cipher. The broadcast MAC protocol defines the procedures used to transmit over the broadcast channel. It also provides FEC using RS coding, and multiplexing to reduce the radio-link error-rate seen by the higher layers. Finally the broadcast physical layer provides the channel structure for the broadcast channel. We will now go on to the detailed implementation of each layer specified in the standard, in the order in which they are processed by a mobile device.

A. Simulation of the inner turbo coding

A turbo code is known to be one of the best channel coding schemes for coping with bit flipping in communication systems. In the CDMA2000 forward channel, a coding-rate r of either $1/3$ or $1/5$ is used for the turbo encoder, which provides increased redundancy to help ensure that the packets are decoded by the mobile even when the channel conditions are poor. A turbo code of rate $1/3$ is a punctured version of the $1/5$ code. Details of the turbo code specified in CDMA2000 are given elsewhere [10], [11].

To evaluate the performance of turbo coding, it is necessary to know the size of a physical-layer packet and to determine the number of decoding chips that will be required when a particular code-rate is used. When the rate of the turbo code is r and QPSK modulation is used, $L_{phy}/2r$ chips are required for an L_{phy}-bit packet because one chip can transport two bits using QPSK. A further constraint is that the the number of chips for processing each physical-layer packet must be a multiple of 1536. As a result, 1536 are required when the code-rate is $1/3$ and 3072 are needed for a code-rate of $1/5$. The performance of the CDMA2000 turbo code for a given number of chips can be determined using the Coded Modulation Library from Iterative Solutions[3], which runs on MATLAB.

B. Implementation of the RS decoder in the MAC layer

We used a software implementation of the RS *erasure* decoder, which is highly optimized for speed and memory, especially as regards the Berlekamp algorithm (i.e., the modified Berlekamp-Massey algorithm [28], [29]). This algorithm is used in Minsky's version of the RS decoder[4].

TABLE I
Time required by each component of the RS algorithm to decode one codeword.

	$T^{(8,N,K)}_{c_i}$		
RS code	C_1	C_2	C_3
(32,24,8)	20.3μs	237.5μs	13.2μs
(32,26,6)	15.7μs	175.1μs	12.1μs
(32,28,4)	10.2μs	118.9μs	11.3μs

This version of the RS decoder runs in an ARM11-based mobile platform[5], operating at 400MHz in pipeline mode. Table I shows the time required to decode an 8-bit wide RS codeword ($\hat{s} = 8$), which is measured on the IAR® embedded workbench[6] for the ARM processor. IAR® embedded workbench is a cycle-accurate simulator that supports instruction-level or functional-level execution-time profiling. In the simulations, the erasures were randomly located in one of the 8 bits of the codeword, and the results presented in Table I are average values; but the position of an erasure within a codeword appears to have negligible effect.

C. Implementation of AES block decryption in the security layer

The broadcast security framework in CDMA2000 is described in detail elsewhere [30]. The broadcast security protocol in the access network encrypts framing packets to form broadcast security packets using the AES encryption

[3]Iterative Solutions, http://www.iterativesolutions.com.

[4]RSCODE project, http://sourceforge.net/projects/rscode.

[5]The ARM processor family is widely used in contemporary 3G phones for the GSM and CDMA2000 networks. For example, QUALCOMM's MSM7500™ convergence platform for CDMA2000 1xEV-DO high-speed wireless multimedia has an integrated 400MHz ARM11 processor.

[6]Embedded development tools, http://www.iar.com.

TABLE II
TIME REQUIRED BY EACH TRANSFORMATION TO DECRYPT A 128-BIT DATA BLOCK ($N_b = 4$), AND BY THE KEY EXPANSION ROUTINE FOR DIFFERENT VALUES OF N_k.

A	ARK	ISR	ISB	IMC
T_A^4	0.727μs	0.097μs	7.373μs	20.633μs
KE				
N_k		4	6	8
$T_{KE}^{N_k}$		30.273μs	28.271μs	41.074μs

procedures [31], and the AES decryption code deciphers those security packets at the mobile which is receiving a video broadcast service. The AES algorithm can be implemented in any combination of software, firmware, and hardware. The appropriate choice depends on several factors such as the application, the environment, and the technology used. There are a number of examples of efficient implementations of this AES algorithm [32], [33] on a variety of platforms.

In this analysis, we used Gladman's reference implementation[7] of the AES algorithm, which is heavily optimized. This code allows blocks of 128, 196, and 258 bits, and this size is set during compilation. Both the code and data for AES decryption are cache-locked, preventing this time-critical process being ejected from from cache. This version of AES block decryption will also run in a mobile device equipped with an ARM11 processor, and the execution time of the AES block decipher is also measured using the IAR® embedded workbench.

The times required by each transformation routine of the AES algorithm to decrypt a 128-bit ($N_b = 4$) data block are presented in Table II, which illustrates how the time required by the key expansion routine depends on the length of the cipher key. The pseudo-code for key expansion [13] shows how the first N_k words of the expanded key are filled with the cipher key. Every subsequent word is the result of an exclusive OR operation between the previous word and the word N_k positions earlier. When a word occurs in a position that is a multiples of N_k, an additional transformation, consisting of a cyclic shift of the four bytes of the word, followed by the application of a table lookup to all four bytes, is applied to the preceding word and the result is subject to a further exclusive OR with the round constant. This transformation is therefore executed more frequently when the value of N_k is small, so that more time is required for key expansion when $N_k = 4$ than when $N_k = 6$, as shown in Table II. However, as N_k increases from 6 to 8, a slightly different key expansion routine is applied and the time required for key expansion increases again because the transformation is applied even more frequently.

D. Parameter settings in the proposed service architecture

The parameters used in our case study are summarized in Table III. The value of each parameter is extracted from the CDMA2000 1xEV-DO [10], [11], [26] standard. We assume the use of QPSK modulation with a reference data-rate of 614.4kb/s for a forward data channel, which is divided into 1.67ms time-slots, because this is known [26] to provide sufficient coverage even with an RS code of (16,12,4). In this paper, QPSK is supposed to be used as a modulation scheme. Additionally, we assume that the forward channel contains five sub-channels.

[7]Code for AES and combined encryption/authentication modes, http://www.gladman.me.uk.

TABLE III
SYSTEM PARAMETERS USED IN OUR ANALYSIS.

Symbol	Value(s)	Description
Physical channel parameters		
f_c	1.8GHz	Carrier frequency
μ_p	614.4kb/s	Reference channel data-rate
$\overline{\mu_u}$	120kb/s	Data-rate of the MAC payload
$\overline{\mu_e}$	90kb/s	Effective data-rate of the MAC payload
ξ_{min}	-1.414dB	Minimum SNR within the service area ($r = 1/3$)
v_{min}	4km/h	$V = 0.0111$
v_{max}	40km/h	$V = 0.111$
Network parameters		
T_{slot}	1.67ms	Slot duration
η	5	Number of sub-channels
r	1/3	Inner turbo-code code-rate
L_{phy}	1024 bits	Length of a physical-layer packet
L_{mac}	1002 bits	Length of a MAC-layer packet
L_u	1000 bits	Length of the payload in a MAC-layer packet
Data processing parameters for the mobile device		
$\hat{s}$	8 bits	Length of a symbol
(N, K, R)	(32,24,8)	Example RS code
N_s	$1 \sim 16$	Number of MAC-layer packets in an ECB row
N_b	4	Number of 32-bit words comprising the state
N_k	4	Number of 32-bit words comprising the cipher key

In CDMA2000 1xEV-DO, the modulation parameter and data-rate used for the forward data channel [10], [11] determine the size of each physical-layer packet, which is 1024 bits under the assumptions that we have made; and thus the payload in every physical-layer packet is a single 1002-bit MAC-layer packet. Each MAC-layer packet has a 1000-bit security-layer packet as its payload. The CDMA2000 1xEV-DO standard also specifies that the physical-layer packets transmitted over the forward data channel are encoded with a code-rate of 1/3 by the turbo encoder when the data-rate of that channel is 614.4kb/s.

For our QoS analysis of scalable video in the CDMA2000 broadcasting environment, we used an example RS code of (32,24,8). Therefore, each sub-channel has a data-rate for its MAC payload of 120kb/s, because we are assuming that there are five sub-channels, which means that the effective data-rate is 90kb/s.

Regarding the physical channel and data processing parameters, we used values of V between 0.0111 and 0.111, which correspond to mobile speeds between 4km/h and 40km/h, with a reference channel data-rate of 614.4kb/s, a carrier frequency of 1.8GHz, and a physical-layer packet length of 1024 bits. We also set the minimum value of SNR experienced by a mobile within a region to -1.414dB. Then we analyzed the service reliability and execution delay achieved by the RS decoder and AES decryption for different amounts of interleaving (i.e., $1 \leq N_s \leq 16$). The unit size of an AES cipher block and the size of the cipher key are 128 bits, and thus the values of N_b and N_k are both 4.

TABLE IV
STATISTICS OF THE SAMPLE VIDEO USED IN OUR ANALYSIS.

Description	Value(s)
Title	*Star Wars IV*, *Silence of the Lambs*
Frames per second	30
Frame size	CIF (352×288)
Scalability scheme	Temporal
Stream type	VBR (variable bit-rate)
GoP size	8
Number of B frames in a GoP	4 or 7
Quantization parameter	10 or 24

E. Characteristics of the sample video and video broadcasting applications

The sample video streams used for this analysis were *Star Wars IV* and *Silence of the Lambs*. The characteristics of this video stream [34] are summarized in Table IV. The frames are packetized into 125 bytes, which is the maximum payload size of physical-layer packet in CDMA2000 1xEV-DO network.

A specified packet loss behavior by both the underlying network module and the data processing module on the mobile is translated by the application module into frame losses in the sample video streams. The quality degradation perceived by a user can then be estimated from (6).

F. Results of QoS analysis

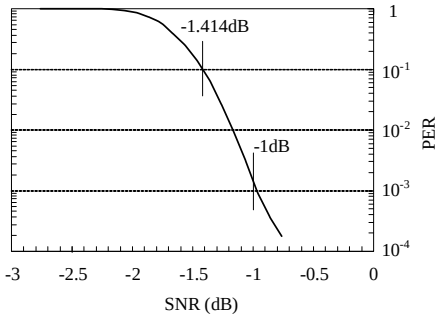


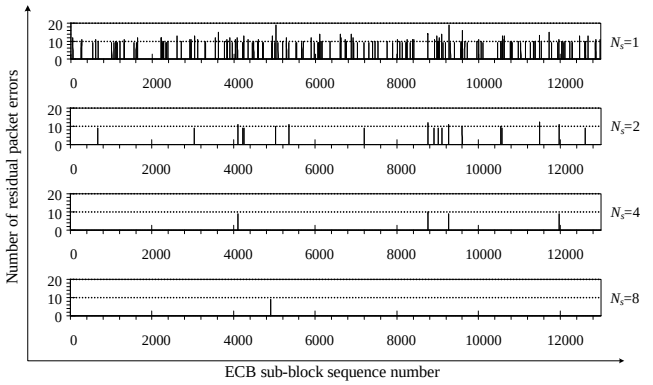
Fig. 9. The effect of turbo coding on the PER for varying values of SNR.

Fig. 9 presents the PER for varying values of SNR when the inner code-rate r is $1/3$, obtained from the turbo coding simulator. These results illustrate the performance of inner coding in CDMA2000 1xEV-DO. For example, we can see that the resulting PER varies between 0.001 and 0.1 when the value of SNR varies between -1.414dB and -1dB.

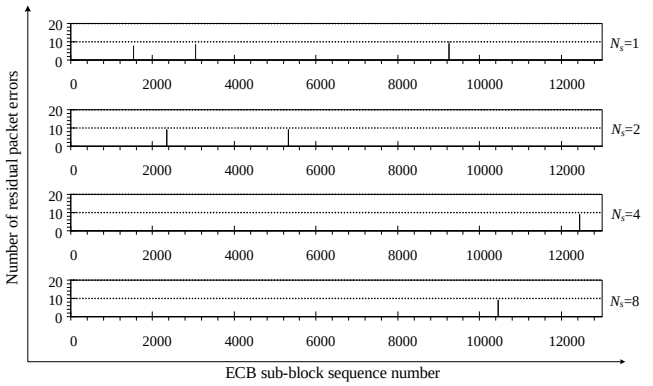
TABLE V
WCET, LATENCY, AND BUFFER SIZE FOR DIFFERENT BLOCK INTERLEAVING PARAMETERS.

N_s	$\overline{\text{PER}}_r$	WCET (ms)	Buffer size (bits)	Latency (ms)
1	2.918E-2	67	39720	331
2	1.178E-2	133	80040	667
4	3.294E-3	263	159600	1330
8	1.201E-3	517	318000	2650
16	9.589E-4	1009	633720	5281

We now present the simulation results obtained from our cross-layer framework. Table V shows the upper bound on the residual rate of packet errors after RS decoding, which is $\overline{\text{PER}}_r$, when the worst SNR is -1.414dB (equivalent to a PER of about 0.1 when $r = 1/3$), and the speed of the mobile device ranges from 4km/h to 40km/h. Additionally, the table shows the WCET required for RS decoding and AES decryption of the data in an ECB, the buffer size required for the initial buffering of a video stream arriving at a mobile device, and the corresponding service latency, for a given block interleaving parameter N_s. These numbers are simulation results with a 99.9% confidence level ($\Upsilon = 99.9$).



(a) 4km/h



(b) 40km/h

Fig. 10. Fluctuations in the number of residual packet errors in each ECB sub-block as its sequence number increases, at different interleaving levels, with a SNR of -1.329dB and at mobile speeds of 4km/h and 40km/h.

Fig. 10 shows traces of the number of residual packet errors in each ECB sub-block for different levels of block interleaving, when the instantaneous SNR experienced by a mobile is -1.329dB (equivalent to a PER of about 0.05 when $r = 1/3$). As shown in Fig. 10(a), more interleaving reduces the number of residual packet errors, because a lower density of errors in the ECB sub-blocks make it more predictable that those errors will be recovered by the RS decoder. However, at the higher mobile speed, error bursts are almost perfectly randomized by the block interleaving, as shown in Fig. 10(b), so the interleaving has no effect on the error-correction performance of RS decoding.

Fig. 11 shows time traces for the RS decoding of consecutive ECB sub-blocks with different levels of block interleaving, measured at mobiles moving at two different speeds, when the instantaneous SNR experienced by a mobile is -1.414dB. With higher levels of block interleaving, error bursts in the physical layer are distributed across more ECB sub-blocks,

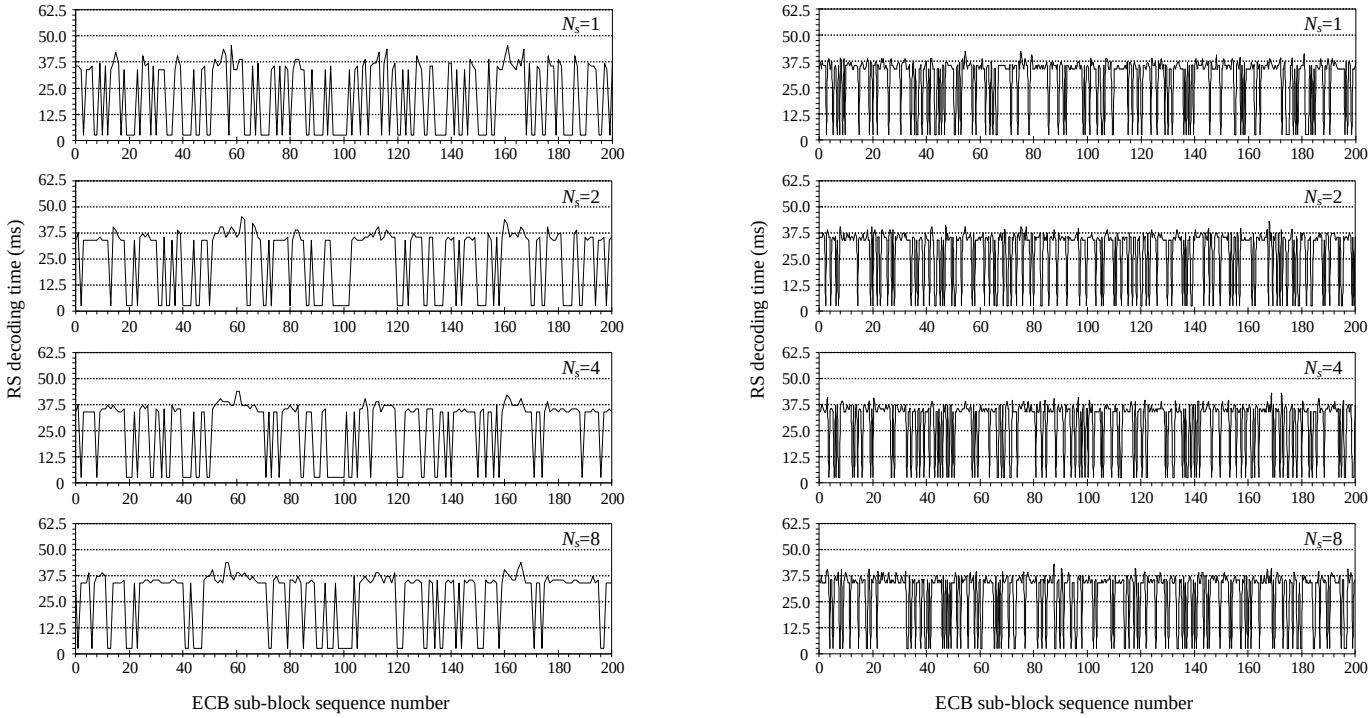


Fig. 11. Fluctuations of the time required for RS decoding of an ECB sub-block as its sequence number increases, with a SNR of -1.414dB and at mobile speeds of 4km/h and 40km/h.

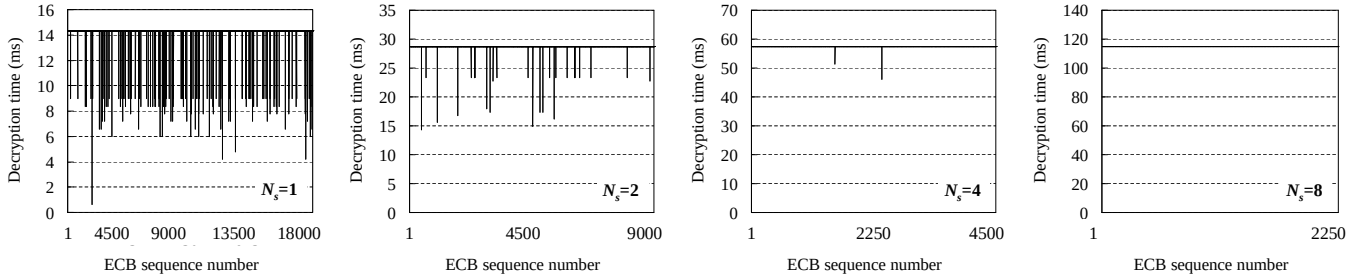
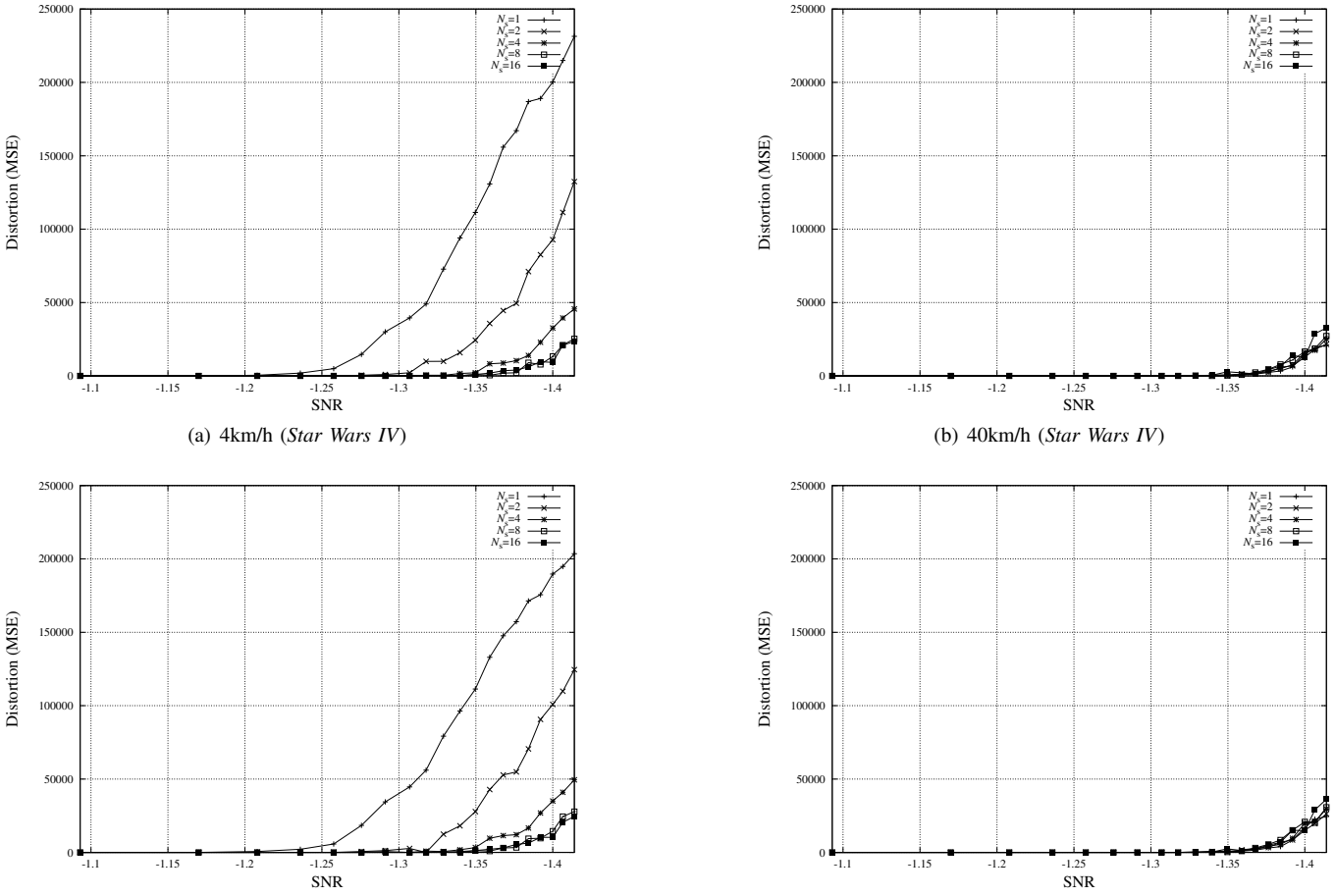


Fig. 12. Fluctuations of the time required for the AES decryption of an ECB as its sequence number increases, with a SNR of -1.329dB and at a mobile speed of 4km/h.

reducing the average number of erased packets in a particular sub-block but few sub-blocks contain no erased symbols, when error burst occurs. This explains the lower and more consistent RS decoding times. Conversely, low levels of interleaving concentrate errors into a smaller number of ECB sub-blocks and the number of error-free sub-blocks increases. This raises the density of errors in each sub-block when the error burst occurs and the WCET increases, resulting in longer, and also more variable, RS decoding times, as shown in Fig. 11. In the faster mobile, these fluctuations are so pronounced that the results almost look random. In this case, the interleaving has no influence on the performance of RS decoding because the pattern of packet errors is already so highly irregular.

Fig. 12 shows traces of AES decryption time for successive ECBs, with different levels of block interleaving at a mobile speed of 4km/h, when the instantaneous SNR experienced by a mobile is -1.329dB. As the level of block interleaving increases, the variability of the decryption time is reduced because more errors are recovered and thus fewer errors remain after RS decoding, which is very clear from Fig. 10(a). We only present the result for the mobile moving at 4km/h because the time required to decrypt an ECB is almost constant for the faster mobile moving at around 40km/h. This is because few residual packet errors remain after RS error correction when the value of SNR is -1.329dB, as shown in Fig. 10(b).

Finally, the perceived quality degradation of the sample video streams, as mean-square errors, are shown in Figs. 13 to 15. At both 4km/h and 40km/h, distortion of the stream quality increases as the SNR decreases. At higher levels of block interleaving, the distortion decreases dramatically at both speeds. However the distortion shows no clear relationship with the level of interleaving for the faster-moving mobile. That is because the bursty errors caused by the fading channel are being randomized almost perfectly by the block



(a) 4km/h (*Star Wars IV*)

(b) 40km/h (*Star Wars IV*)

(c) 4km/h (*Silence of the Lambs*)

(d) 40km/h (*Silence of the Lambs*)

Fig. 13. Perceived quality of a scalable video stream for different levels of interleaving and mobile speeds (number of B frames = 7, quantization parameter = 10).

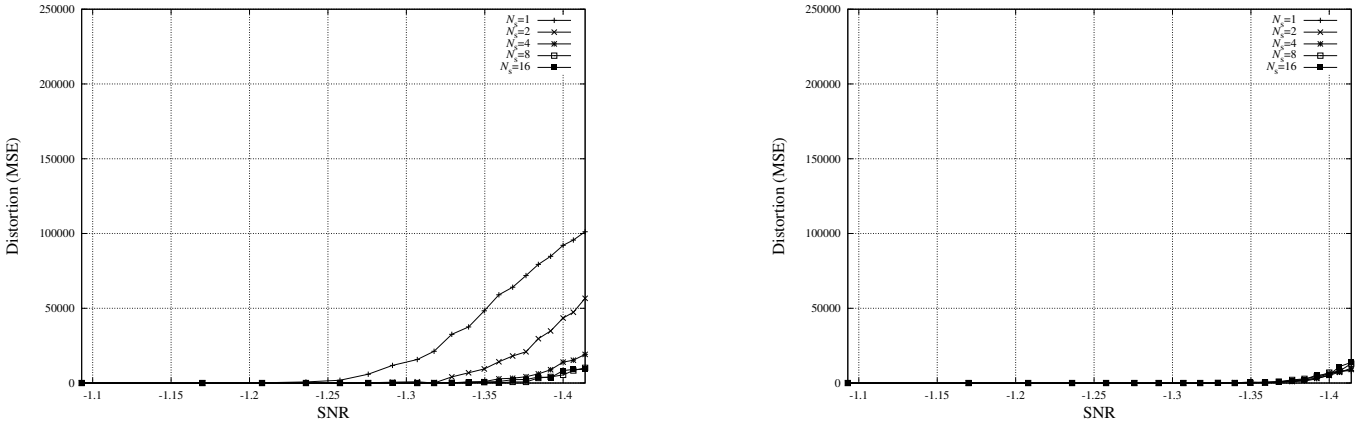
interleaving, which can have no further effect on the perceived quality of the video stream.

Fig. 13 shows the perceived quality of the two video streams, *Star Wars IV* and *Silence of the Lambs*, with the same number of B frames and the same quantization parameter. We see that the way in which quality declines is similar for both streams. Fig. 14 shows the perceived quality of *Star Wars IV* with different numbers of B frames in a GoP. If there are fewer B frames in a GoP then there must be more P frames, which allow increased error-resilience. Finally, Fig. 15 shows the perceived quality of the *Star Wars IV* video stream with different quantization parameters. The use of AVC video compression with higher quantization parameters results in more compression and smaller frames, which are more tolerant of packet errors, and this reduces the perceived distortion of video streams over lossy wireless networks.

VI. Discussion

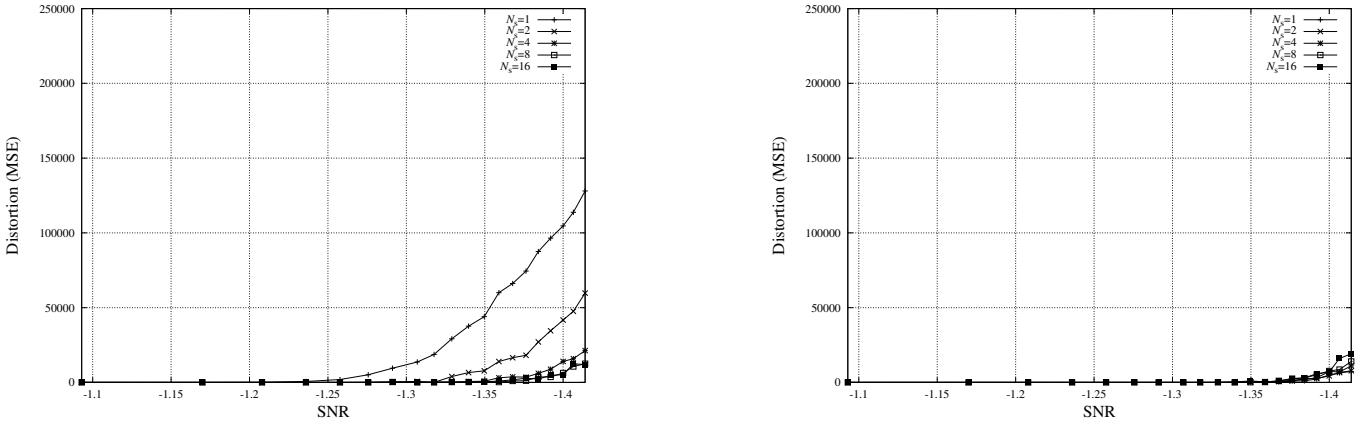
Our framework could be useful in other scenarios if it were improved in a few ways:

- First, we have assumed that network operations are executed on a separate hardware chipset from the main processor (e.g., a modem processor), and this separate chipset is mainly present in order to correct errors in data packets. If some network operations have to be performed by the main processor, then an accurate assessment of computation time must allow for the scheduling of that processor.
- Second, our current approach to the validation and translation of input parameters is primitive. In order to provide more reliable guidelines for different system configurations, we need to improve the verification and translation of system configurations using a formal language such as the Architecture Analysis & Design Language (AADL) [35]. Using the AADL to describe a wireless system for broadcast services would allows an unfamiliar designer to check the performance of a configuration with different parameters. The AADL detects the use of invalid configurations, and the model itself constitutes a record of a set of acceptable system parameters. Many parameters are required for an analysis, so this arrangement would significantly reduce the possibility of parameters being misinterpreted and the scope for communication problems among system designers and engineers.
- Finally, it is desirable that feedback from users regarding perceived quality should be forwarded to the components



(a) 4km/h (b) 40km/h

Fig. 14. Perceived quality of a scalable video stream for different levels of interleaving and mobile speeds (*Star Wars IV*, number of B frames = 4, quantization parameter = 10).



(a) 4km/h (b) 40km/h

Fig. 15. Perceived quality of a scalable video stream for different levels of interleaving and mobile speeds (*Star Wars IV*, number of B frames = 7, quantization parameter = 24).

in the underlying layers on the mobile device, and also sent back to the service provider, so that they can adjust either the application or network parameters accordingly. This is part of our program of work to increase the extent to which the user's perspective is considered in the cross-layer design of multimedia broadcasting services.

VII. Conclusion

In this paper, we have proposed a QoS assessment framework for video broadcasting services in wireless cellular networks. The proposed framework is flexible in nature and can easily be extended to different system and network scenarios. By using a layered architecture of simulation modules, our analysis framework enables the systematic validation and evaluation of each module and the entire system before the deployment of a new type of system component or service. Using this framework, we analyzed the system and user performance of a mobile device in scalable video broadcasting in CDMA2000 1xEV-DO networks. By looking at different parameters for the system configurations such as wireless physical channel, network, and video streams, we were able to provide a baseline for client performance in terms of video quality in wireless broadcast networks.

References

[1] Opnet technologies - making networks and applications perform, http://www.opnet.com.

[2] The network simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[3] C. Atici and M. O. Sunay, "High data-rate video broadcasting over 3G wireless systems," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 212–223, Mar. 2007.

[4] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall, "Delivery of broadcast services in 3G networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 188–199, Mar. 2007.

[5] N. H. Vaidya, J. Bernhard, V. V. Veeravalli, P. R. Kumar, and R. K. Iyer, "Illinois wireless wind tunnel: a testbed for experimental evaluation of wireless networks," In *Proc. ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, Aug. 2005, pp. 64–69.

[6] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control on fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445–455, May 1997.

[7] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE Transactions on Communications*, vol. 45, no. 6, pp. 660–667, Jun. 1997.

[8] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468–1477, Nov. 1998.

[9] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," IEEE Transactions on Communications, vol. 48, no. 10, pp. 1681–1691, Oct. 2000.

[10] R. Parry, "CDMA2000 1xEV-DO [for 3G communications]," *IEEE Potentials*, vol. 21, no. 4, pp. 10–13, Oct./Nov. 2002.

[11] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, Jul. 2000.

[12] Y. Q. Shi, X. M. Zhang, Z.-C. Ni, and M. Ansari, "Interleaving for combating bursts of errors," *IEEE Wireless Communicaitons*, vol. 4, no. 1, pp. 29–42, 2004.

[13] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - the Advanced Encryption Standard*, Springer-Verlag, 2002.

[14] W. C. Jakes, *Microwave Mobile Communications*, Wiley-IEEE Press, May 1994.

[15] C. Berrou, A. Glavieux "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[16] S. Benedetto and G. Montorsi, "Unveiling turbo-codes: some results on parallel concatenated coding schemes", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–429, Mar. 1996.

[17] W. J. Ebel and W. H. Tranter, " The performance of Reed-Solomon codes on a bursty-noise channel," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 298–306, Feb./Mar/Apr. 1995.

[18] K. Kang, "Probabilistic Analysis of Data Interleaving for Reed-Solomon Coding in BCMCS," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3878–3888, Oct. 2008.

[19] K. Kang, Y. Cho, and H. Shin, "Energy-efficient MAC-layer error recovery for mobile multimedia applications in 3GPP2 BCMCS," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 338–349, Mar. 2007.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[21] H. Mansour, P. Nasiopoulos, and V. Krishnamurthy, "Modelling of loss distortion in hierarchical prediction codecs," In *Proc. IEEE International Symposium on Signal Processing and Information Technology*, Aug. 2006, pp. 536–540.

[22] P. Wellin, R. Gaylord, and S. Kamin, *Introduction to Programming with Mathematica*, Cambridge University Press, 2005.

[23] T. Gayley, "Building user interfaces using J/Link," *The Mathematica Journal*, vol. 9, no. 1, pp. 189–215, 2003.

[24] K. Kang and L. Sha, "QoS-guaranteed reed-solomon coding with block interleaving for real-time broadcast services in 3G wireless networks," submitted for publication in *IEEE Transactions on Computers*.

[25] CDMA2000 high rate broadcast-multicast packet data air interface specification, 3GPP2 Std. C.S0054-A Rev. 1.0, Mar. 2006.

[26] P. Agashe, R. Rezaiifar, and P. Bender, "CDMA2000 high rate broadcast packet data air interface design," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 83–89, Feb. 2004.

[27] J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, and E. Tiedemann, "Broadcast and multicast services in CDMA2000," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 76–82, Feb. 2004.

[28] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[29] N. B. Atti, G. M. Diaz-Toca, and H. Lombardi, "The Berlekamp-Massey algorithm revisited," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 1, pp. 75–82, Apr. 2006.

[30] Broadcast-multicast service security framework, 3GPP2 Std. S.R0083 Rev. 1.0, Oct. 2003.

[31] Enhanced Cryptographic Algorithms, 3GPP2 Std. S.S0055 Rev. 2.0, Jan. 2005.

[32] X. Zhang and K. K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," *IEEE Circuits and Systems Magazine*, vol. 2, no. 4, pp. 24–46, 2002.

[33] G. Bertoni, L. Breveglieri, P. Fragneto, M. Macchetti, and S. Marchesin, "Efficient software implementation of AES on 32-bit platforms," *Lecture Notes in Computer Science*, vol. 2523, pp. 159–171, Aug. 2002.

[34] Video traces for network performance evaluation, http://trace.eas.asu.edu/tracemain.html.

[35] P. H. Feiler, D. P. Gluch, and J. J. Hudak, "The architecture analysis & design language (AADL):an introduction," The Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2006-TN-011, Feb. 2006.