Toward an Improvement of H.264 Video Transmission over IEEE 802.11e through a Cross-Layer Architecture

Adlen Ksentini and Mohamed Naimi, LICP Lab
Abdelhak Guéroui, PRiSM Lab

ABSTRACT

The recently developed H.264 video standard achieves efficient encoding over a bandwidth ranging from a few kilobits per second to several megabits per second. Hence, transporting H.264 video is expected to be an important component of many wireless multimedia services, such as video conferencing, real-time network gaming, and TV broadcasting. However, due to wireless channel characteristics and lack of QoS support, the basic 802.11-based channel access procedure is merely sufficient to deliver non-real-time traffic. The delivery should be augmented by appropriate mechanisms to better consider different QoS requirements and ultimately adjust the medium access parameters to the video data content characteristics. In this article we address H.264 wireless video transmission over IEEE 802.11 WLAN by proposing a robust cross-layer architecture that leverages the inherent H.264 error resilience tools (i.e., data partitioning); and the existing QoS-based IEEE 802.11e MAC protocol possibilities. The performances of the proposed architecture are extensively investigated by simulations. Results obtained indicate that compared to 802.11 and 802.11e, our cross-layer architecture allows graceful video degradation while minimizing the mean packet loss and end-to-end delays.

INTRODUCTION

One of the driving forces of the next wireless LAN (WLAN) generation is the promise of high-speed multimedia service. Providing multimedia services to mobiles and fixed users through wireless access can be a reality with the development of:

- Two high-speed physical (PHY) layers, IEEE 802.11g (54 Mb/s) [1] and IEEE 802.11n (100 Mb/s) [2]
- The new IEEE 802.11e quality of service (QoS)-based medium access control (MAC) layer [3]

However, wireless channel characteristics such as shadowing, multipath, fading, and interferences still limit the available bandwidth for the deployed applications. Consequently, video compression techniques are a crucial part of multimedia applications over WLAN.

Recently, the H.264/AVC [4] video coding standard, proposed by both the Joint Video Team (JVT) of the International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) and the Moving Picture Experts Group (MPEG), achieved a significant improvement in compression efficiency over the existing standards. For instance, digital satellite TV quality was reported to be achievable at 1.5 Mb/s, compared to the current operation point of MPEG-4 part 2 video codec [5] at around 3 Mb/s. Additionally, H.264 standard introduces a set of error resiliency techniques such as slice structure, data partitioning, and flexible macroblock ordering (FMO). However, these techniques are insufficient because the resource management and protection strategies available in the lower layers (PHY and MAC) are not optimized explicitly considering the specific characteristics of multimedia applications.

This article focuses on the transmission of H.264 video streams over WLAN (IEEE 802.11e-based) by proposing a QoS cross-layer architecture based on both Application layer and MAC layer features. The proposed cross-layer architecture relies on a data partitioning (DP) technique at the application layer and an appropriate QoS mapping at the 802.11e-based MAC layer. Through employing DP, the H.264 encoder partitions the compressed data in separate units of different importance (called *partitions*). Based on the QoS requirements of those different partitions, we specify a marking algorithm at the MAC layer that associates each partition with an access category (AC) provided by 802.11e enhanced distributed channel access (EDCA). Thus, we allow the application layer to pass its streams along with their requirements in order to protect the most important H.264 information, which guarantees low degradation of received H.264 stream.

In order to be able to change picture parameters without necessarily retransmitting PSC updates, the video codec can continuously maintain a list of parameter set combinations to switch on. In this case each slice header would contain a codeword that indicates the PSC to be used.

The remainder of this article is organized as follows. We introduce the H.264 video standard. We give a brief overview of 802.11 and 802.11e MAC protocols. Related work on video over WLAN is presented. We describe the proposed cross-layer architecture. We also devote a section to the performance evaluations. Finally, we conclude this article.

H.264 Standard Overview

H.264 consists of two conceptually different layers. First, the video coding layer (VCL) contains the specification of the core video compression engines that achieve basic functions such as motion compensation, transform coding of coefficients, and entropy coding. This layer is transport-unaware, and its highest data structure is the video slice — a collection of coded macroblocks (MBs) in scan order. Second, the network abstraction layer (NAL) is responsible for the encapsulation of the coded slices into transport entities of the network. In this H.264 overview, we particularly focus on the NAL layer features and transport possibilities. The reader can refer to [4] for more details of VCL layer characteristics.

Network Abstraction Layer

The NAL defines an interface between the video codec itself and the transport world. It operates on NAL units (NALUs) that improve transport abilities over almost all existing networks. An NALU consists of a one-byte header and a bit string that represents, in fact, the bits constituting the MBs of a slice. The header byte itself consists of an error flag, a disposable NALU flag, and the NALU type. Finally, the NAL provides a means to transport high-level syntax (i.e., syntax assigned to more than one slice, e.g., to a picture or group of pictures) to an entire sequence.

Parameter Set Concept

One very fundamental design concept of the H.264 codec resides in its ability to generate self-contained packets, making mechanisms such as header duplication and MPEG-4's header extension code (HEC) unnecessary. The way this is achieved is to decouple information relevant to more than one slice from the media stream. This higher-layer meta information should be sent reliably, asynchronously, and before transmitting video slices. Here, provisions for sending this information in-band are also available for applications that do not have an out-of-band transport channel appropriate for the purpose. The combination of higher-level parameters is called the parameter set concept (PSC). The PSC contains information such as picture size, display window, optional coding modes employed, MB allocation map, and so on. In order to be able to change picture parameters without necessarily retransmitting PSC updates, the video codec can continuously maintain a list of parameter set combinations to switch on. In this case each slice header would contain a codeword that indicates the PSC to be used.

Error Resilience Tools: Data Partitioning

The H.264 standard includes a number of error resilience techniques. Among these techniques, DP is an effective application-level framing technique that divides the compressed data into separate units of different importance. Generally, all symbols of MBs are coded together in a single bit string that forms a slice. However, DP creates more than one bit string (partition) per slice, and allocates all symbols of a slice into an individual partition with a close semantic relationship. In H.264 three different partition types are used:

- Partition A, containing header information such as MB types, quantization parameters, and motion vectors. This information is the most important because without it, symbols of the other partitions cannot be used.
- Partition B (intra partition), carrying intra coded block pattern (CBP) and intra coefficients. The type B partition requires the availability of the type A partition in order to be useful at the decoding level. In contrast to the inter information partition, intra information can stop further drift and hence is more important than the inter partition.
- Partition C (inter partition), containing only inter CBPs and inter coefficients. Inter partitions are the least important because their information does not resynchronize the encoder and decoder. In order to be used it requires the availability of the type A partition, but not of the type B partition.

Usually, if the inter or intra partitions (B or C) are missing, the available header information can still be used to improve the efficiency of error concealment. More specifically, due to the availability of the MB types and motion vectors, a comparatively high reproduction quality can be achieved as only texture information is missing.

IEEE 802.11 Wireless LAN

Distributed Coordination Function

The distributed coordination function (DCF) is the basic mechanism for IEEE 802.11. It employs carrier sense multiple access with collision avoidance (CSMA/CA) as the access method. Before initiating a transmission, each station is required to sense the medium. If the medium is busy, the station defers its transmission and initiates a backoff timer. The backoff timer is randomly selected between 0 and contention window (CW). Once the station detects that the medium has been free for a duration of DCF interframe spaces (DIFS), it begins to decrement the backoff counter as long as the channel is idle. As the backoff timer expires and the medium is still free, the station begins to transmit. In case of a collision, indicated by the lack of an acknowledgment, the size of the CW is doubled following Eq. 1 until it reaches the CW_{max} value. Furthermore, after each successful transmission, the CW is initialized with CW_{min},

$$CW = (CW_{min} \times 2^i) - 1, \qquad (1)$$

where i is the number of transmission attempts.

However, under DCF, all stations compete for channel access with the same priority. There is no differentiation mechanism to provide better service for real-time and multimedia applications [6].

EDCA: ENHANCED DISTRIBUTED ACCESS CHANNEL

The need for a better access mechanism to support service differentiation has led Task Group e of IEEE 802.11 to propose an extension of the actual IEEE 802.11 standard. The 802.11e draft introduces the hybrid coordination function (HCF) that concurrently uses a contention-based mechanism and a pooling-based mechanism, EDCA, and HCF controlled channel access (HCCA), respectively. Like DCF, EDCA is very likely to be the dominant channel access mechanism in WLANs because it features a distributed and easily deployed mechanism. In the following we focus on EDCA; for more details on HCCA refer to [3].

QoS support in EDCA is realized with the introduction of access categories (ACs). Each AC has its own transmission queue and its own set of channel access parameters. Service differentiation between ACs is achieved by setting different $CW_{\min}$, $CW_{\max}$, arbitrary interframe space (AIFS), and transmission opportunity duration limit ($TXOP_{\text{limit}}$) (optional). If one AC has a smaller AIFS or $CW_{\min}$ or $CW_{\max}$, the AC's traffic has a better chance of accessing the wireless medium earlier. Generally, AC3 and AC2 are reserved for real-time applications (e.g., voice or video transmission), and the others (AC1, AC0) for best effort and background traffic.

RELIABLE VIDEO COMMUNICATION OVER WLAN

Existing work [7] on wireless video transmission focuses particularly on application-level QoS control in order to combat wireless transmission errors. Error control is one of the most popular application-level approaches dealing with packet loss and delay in multimedia communication over bandwidth-limited fading wireless channels. Two classes of communication protocols are used in practice to reliably communicate data over packet networks: synchronous and asynchronous. Asynchronous communication protocols such as automatic repeat request (ARQ) operate by dividing the data into packets and appending a special error check sequence to each packet for error detection purposes. The receiver decides whether a transmission error occurred by calculating the check sequence. For each intact data packet received in the forward channel, the receiver sends back an acknowledgment. While this model works very well for data communication, it is not suitable for multimedia streams with hard latency constraints. The maximum delay of the ARQ mechanism is unbounded, and in the case of live streaming it is necessary to interpolate late arriving or missing data rather than insert a delay in the stream playback.

In synchronous protocols (i.e., FEC-based protocols), data are transmitted with a bounded delay but generally not in a channel adaptive manner. Forward error correction (FEC) codes are designed to protect data against channel erasures by introducing parity packets. No feedback channel is required. If the number of erased packets is less than the decoding threshold for the FEC code, the original data can be recovered perfectly. Note that Reed-Solomon (RS) codes are usually used to generate packet-level FEC blocks. The aim of RS codes is to produce at the sender n blocks of encoded data from k blocks of source data in such a way that any subset of k encoded blocks suffices at the receiver to reconstruct the source data. Nonetheless, the FEC mechanism represents a lack of efficiency since FEC does not adapt to variable error channel conditions: either a waste of bandwidth may occur when the radio channel is in good condition, or insufficient error protection may exist when it gets bad.

The FEC mechanism represents a lack of efficiency since FEC does not adapt to variable error channel conditions: either a waste of bandwidth may occur when the radio channel is in a good condition, or insufficient error protection may exist when it gets bad.

It should be pointed out that approaches based on these two mechanisms (FEC, ARQ) are implemented and supervised at the application layer, and consequently do not have access to lower layers' transmission parameters.

Cross-layer architecture is an interesting alternative to the above mentioned mechanisms [8] for robust H.264 video transmission over WLAN. In [9] the authors give an excellent review of the existing solutions for combing techniques deployed at the application layer, and techniques available at either the PHY or the MAC layer. In fact, the authors classify cross-layer architectures for video transport over wireless networks into five categories:

- Top-down: The higher layer optimizes their parameters and the strategies at the next lower layer.
- Bottom-up approach: In this architecture the lower layer isolates the higher layers from losses and bandwidth variations.
- Application-centric approach: The application layer optimizes the lower-layer parameters one at a time in either bottom-up (starting from the PHY layer) or top-down manner, based on its requirements.
- MAC-centric approach: In this cross-layer technique the application layer passes its traffic information and requirements to the MAC, which decides which application layer packets/flows should be transmitted and at what QoS level.
- Integrated approach: The strategies to design a cross-layer architecture are determined jointly by all the open system interconnection (OSI) layers.

In our previous work [10] we have proposed to use a hierarchical H.264 coding scheme. However, unlike MPEG-4, where the fine granularity scalability (FGS) mechanism is specified to use hierarchical coding, the H.264 standard (current version) does not contain any specification for this purpose.

As classified in [9], our work in this article (cross-layer architecture) falls into the MAC-centric category. Indeed, rather than using hier-

The NAL layer facilitates the delivery of the H.264 VCL data to the underlying transport layers. Each NALU could be considered a packet that contains an integer number of bytes, including a header and a payload.

archical coding, we favor more interaction between, on one hand, the H.264's VCL layer that divides the original streams through DP and, on the other hand, the MAC layer that treats video streams differently. Thus, the application layer passes its traffic information (the priority of the streams) with their QoS requirements to the MAC layer, which maps these partitions to different traffic categories to improve the perceived video quality.

The Proposed Cross-Layer Architecture

VCL Layer

The VCL layer provides the core high compression representation of the video picture content. Besides the three partitions (A, B, and C) obtained when DP is enabled, the VCL layer generates an additional type of slice that represents instantaneous decoding refresh (IDR) pictures. The IDR access unit contains information that cannot be packed into the three partitions. That is, they contain only the intra picture (coded picture that can be decoded without needing information from previous pictures) where no data partitioning can be applied. Thus, the generated slices (A, B, C, PSC, and IDR) are directed to the NAL layer with an associated header using an implementation-dependent data structure. Structure elements include the type of data conveyed (slice type) and additional information such as picture ID, MB address, and entropy coding. The order in which the slice units are sent is constant. The first slice units transmitted contain the PSC information, which includes some parameters set related to the encoder configuration and relevant for all pictures in the video sequence. The next slice units transmitted contain the IDR picture. Since IDR frames may contain only *I* slices without data partitioning, they are usually sent at the start of video sequences (just after the PSC). The slice units following the IDR frames contain one of the three partitions (A, B, or C).

NAL Layer

The NAL layer facilitates the delivery of the H.264 VCL data to the underlying transport layers. Each NALU could be considered as a packet that contains an integer number of bytes, including a header and a payload. The header specifies the NALU type, and the payload contains the related data. At this point, the NAL header contains three fields; we focus particularly on the Nal_Ref_Idc (NRI) field. The NRI contains two bits that indicate the priority of the NALU payload, where 11 is the highest transport priority, followed by 10, then by 01, and finally, 00 is the lowest. Accordingly, the incoming VCL layer's slices are differentiated and encapsulated into NALUs by enabling the NRI field in the NAL header. Thus, it is obvious that PSC packets obtain the highest priority. Furthermore, as information carried in both partition A and IDR are essential for decoding an entire video frame, it is important to give these slices more priority than partition B and C. Based on these rules, the NAL layer marks the different NALUs, and hence, the MAC IEEE 802.11e layer takes over in order to protect the more important NALUs.

The IEEE 802.11e MAC Layer

The current 802.11e draft defines four access categories; AC3 corresponds to the highest access priority, and AC0 to the lowest. Based on this traffic specification it is possible to differentiate the H.264 partitions at the MAC layer. In this context we propose a marking algorithm that uses the NRI field in order to map the H.264 stream to a suitable traffic class and thus allows for QoS continuity between the different OSI layers.

Thus, each NALU arrives at the MAC layer along with a specific priority value (Fig. 1). According to the marking algorithm, the NALU is encapsulated into a QoS data frame, where the traffic category identifier (TID) field (in the MAC header) is used in order to differentiate between AC[*i*]'s frames. Here, the TID field is 4 bits, and can carry values between 0 and 15. TID values from 8 to 15 represent traffic streams as specified in [3].

The Marking Algorithm

Case of NRI:
11 then QoS_TID = t_1 *// Insert this packet in the AC3 queue.*
10 then QoS_TID = t_2 *// Insert this packet in the AC2 queue.*
01 then QoS_TID = t_3 *// Insert this packet in the AC1 queue.*
$t_1 < t_2 < t_3$ *and* $t_1, t_2, t_3 \in$ *[8.15]*

The choice of AC is based on QoS metrics such as one-way loss rate and one-way delay. Thus, the PSC is mapped to the highest-priority access category (AC3). We argue this by the fact that the stream is very sensitive to packet loss because a missing parameter set leads to delay of the whole video transmission. Since IDR pictures contain costly information (I frames) for the encoder procedure and partition A carries vital information for the encoded frame, both need bounded delays and minimum loss rates. Accordingly, the two streams are mapped to the same access category (AC2). Finally, partitions B and C do not require any QoS metrics, and are mapped to lower-priority access category AC1. This way, we ensure that partitions B and C are differentiated from best effort traffic (AC0).

As previously stated, each AC contends for the channel access by using AC-specified MAC parameters from the EDCA parameters set (AIFS[AC], CW_{max}[AC], CW_{min}[AC], and $TXOP_{limit}$ if enabled). In addition to the above contention parameters, we use the maximum retry limit. In fact, the 802.11e MAC layer uses a retry count variable, which is incremented after each transmission fails. Thus, when the retry count exceeds the maximum retry limit, the failing frame is dropped. We use this parameter (maximum retry limit) to unequally protect the high-priority information. In fact, one solution is to increase the retry of an important packet, at the expense of losing less important packets (as long as the receiver can accommodate this extra

Cross-layer Architecture	AIFS (μs)	CW_{min}	CW_{max}	Queue length	Max retry limit
Parameter set information (AC3)	50	7	15	50	8
IDR and partition A (AC2)	50	15	31	50	8
Partitions B and C (AC1)	50	31	1023	50	4
Background traffic (AC0)	70	31	1023	50	4
EDCA					
H.264 streams (AC2)	50	15	31	50	8
Background traffic (AC0)	70	31	1023	50	8
DCF	DIFS(μs)				
All traffic	50	31	1023	50	8

Table 1. *802.11 (IBSS) MAC parameters.*

A high retry limit's values decrease the frame drop-rate, but may throttle the data rate and throughput because of longer backoff time, while a smaller retry limit value increases frame drop-rate but shorten backoff time.

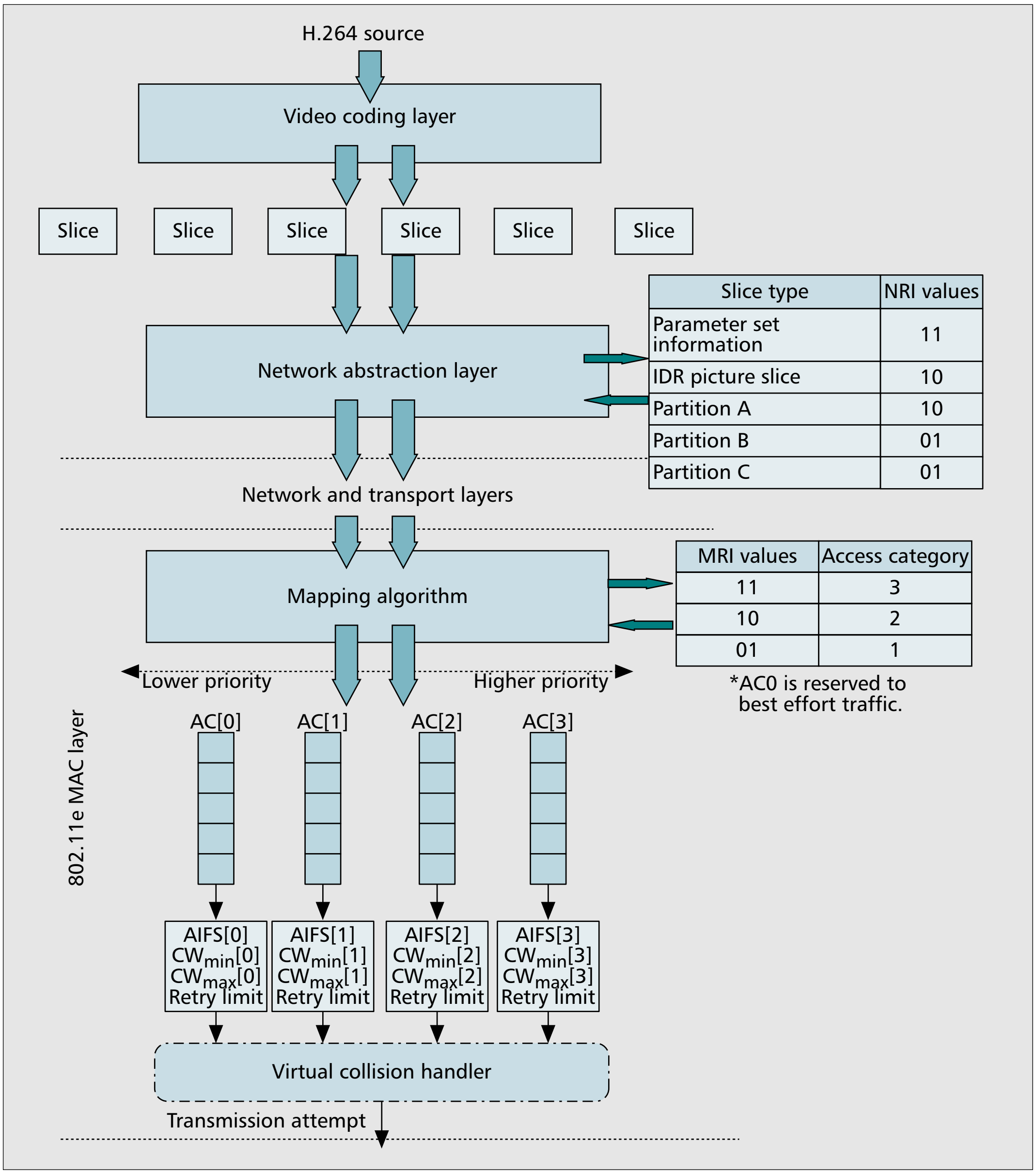
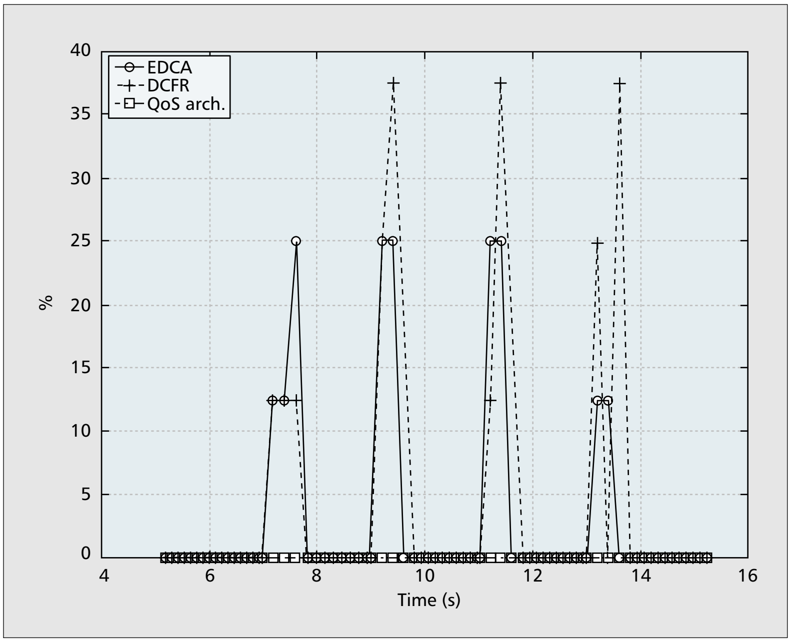
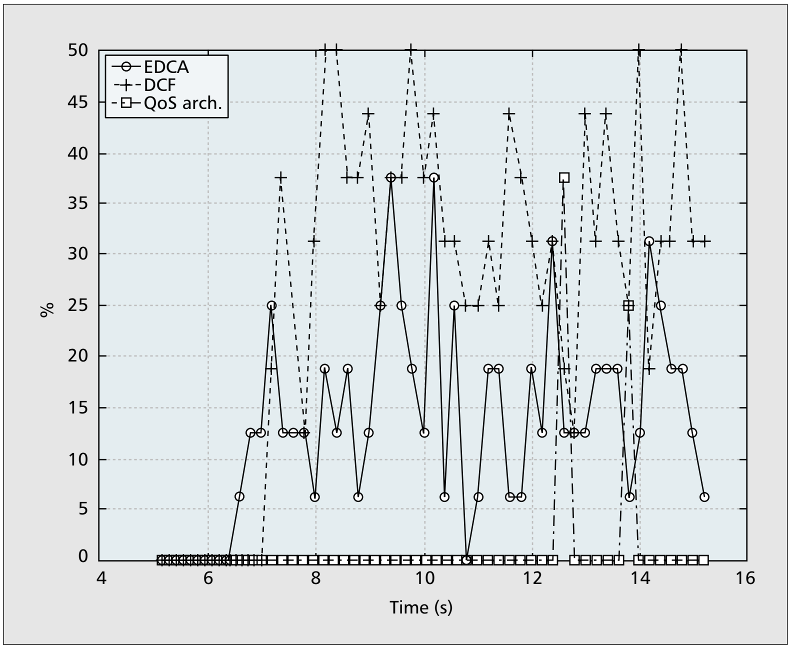


Figure 1. *Cross-layer QoS architecture.*



■ **Figure 2.** *IDR loss rate.*



■ **Figure 3.** *Partition A loss rate.*

retry latency). However, we should keep in mind that a high retry limit's values decrease the frame drop rate, but may throttle the data rate and throughput because of longer backoff time, while a smaller retry limit value increases frame drop rate but shortens backoff time. Performing the precise analysis to find the appropriate values that satisfy these constraints is beyond the scope of this article, and we would like to leave it for future studies. In this article we consider the following assumption to fix the retry limit for each AC:

- Since AC3's and AC2's CW range is small enough (small $CW_{\min}$ and $CW_{\max}$), we use the maximum value allowed by the 802.11 MAC layer.
- For AC1 and AC0 we choose a smaller maximum retry count. This limits the MAC's retransmission of both AC1's and AC0's packets, and hence discards packets that are too late to be sent (AC0's and AC1's packets wait more time in the MAC's queue than AC3's and AC2's packets).

Simulation and Results

In order to evaluate the advantage of the proposed QoS cross-layer architecture, we have constructed simulations using Network Simulator NS2 [11]. The QoS architecture (also called cross-layer architecture) is compared to EDCA (all H.264 slices share the same AC) and DCF.

Simulation Model

For the simulations, we used the H.264 Foreman CIF sequence (10 s). This sequence is coded at 25 frame/s with an intra period of 50 (i.e., after each 50 video pictures we transmit an intra-coded picture refreshment to prevent eventual error propagation). We simulate unicast H.264 video transmission (one video server and one video client) utilizing an independent basic service set (IBSS) architecture at 2 Mb/s. Besides the H.264 stream, the server station generates background traffic (300 kb/s) using constant bit rate (CBR) traffic over User Datagram Protocol (UDP). This allows us to increase the virtual collisions (if the backoff timer of two or more ACs collocated in one station elapse at the same time) at the server's MAC layer. Furthermore, we include four wireless stations where each station generates 300 kb/s of data using CBR traffic in order to overload the wireless network.

Each simulation run consists of 15.1 s of simulated network lifetime. From $t = 0$ s to $t = 5$ s, the channel is empty. Beginning at $t = 5$ s, H.264 and CBR flows are started and begin competing for the channel. Here CBR flows are started at 0.5 s intervals. Between $t = 5$ s and $t = 5.1$ s, only the PSC stream is sent. Finally, at $t = 5.1$ s, the other H.264 flows are started. At this point, it is important to note that NALU packets are encapsulated into Real-Time Protocol (RTP) packets [12] according to a simple packetization scheme. Table 1 shows the MAC parameters used for the simulations.

Results Analysis

Figure 2 depicts IDR's packet loss rates[1] when using QoS architecture vs. EDCA and DCF. The QoS architecture achieves 0 percent loss, despite the increase in the channel's load. In contrast, in DCF and EDCA the mean loss rate is 30 and 16 percent, respectively. This is mainly due to the fact that our QoS architecture associates IDR's packets with an access category (AC2) that gives more channel access opportunities (transmission). Additionally, within our QoS architecture IDR's packets share the same queue with partition A packets. At the same time, within DCF and EDCA, IDR's packets share the queue with all flows and H.264's flows, respectively. This leads to filling the queue very quickly, causing increased probability of dropping incoming packets.

[1] In loss rate measurement, we take into account the packets dropped at the AC's queue, after exceeding the maximum retry limit, and due to collisions.

Figure 3 illustrates the instantaneous loss rates affecting partition A packets. It is interesting to note that not only does the QoS architecture provide better service to IDR's packets, but it also outperforms EDCA and DCF when serving partition A packets. The mean loss rate in our QoS architecture is 2 percent, while it is 13.75 and 27 percent in EDCA and DCF, respectively.

Figures 4 and 5 give the loss rates experienced by partition B's and C's packets, respectively. From these measurements, it appears that AC1 (B and C partitions) experiences higher loss rates in our architecture than in DCF and EDCA. This is to be expected since in our architecture:

- The lower priority of AC1 reduces the transmission opportunities.
- A reduced AC1 maximum retry limit involves a high drop rate at the MAC layer.

Moreover, we notice that EDCA achieves the best performances, since partitions B and C are transmitted with AC2 (AC1 in the cross-layer architecture).

Note that although partitions B and C undergo high packe loss rates in the cross-layer architecture, the results on the final decoded frame are less harmful compared to the case when partition A or IDR packets are lost. In fact, a loss of partition C's or B's packet belonging to a frame leads to degradation of only the quality of the decoded frame (only texture information is lost). Nevertheless, when either partition A's or IDR's packet belonging to a frame is lost, this frame is automatically dropped.

Figures 6 and 7 show the delays experienced by IDR's and partition A's packets, respectively. Our architecture reduces the delay to the minimum level, indicating that packets are transmitted almost immediately. However, in DCF partition A's and IDR's packets access the medium without priority, resulting in greatly increased packet delays. Added to that, in EDCA all H.264 partitions share the same AC, so partition A's and IDR's packets have to compete with partition B's and C's packets to access the medium, which leads to a significant increase in queuing delays.

Figures 8a, 8b, and 8c represent a final decoded frame (n°76) when using DCF, EDCA, and QoS architecture, respectively. It is readily realized that QoS mapping outperforms both DCF and EDCA. At this point, it is interesting to note that we experienced 87 and 41 dropped frames that cannot be decoded when transmitting video in DCF and EDCA, respectively. In contrast, in our cross-layer architecture we were able to decode the whole video sequence (250 frames).

Conclusion

In this article we introduce a new cross-layer architecture that ensures robust H.264 video transmission over IEEE 802.11-based wireless networks. The proposed cross-layer architecture is based on two main interactions. First, a top-down cross-layer interaction allows the H.264 NAL video delivery module to transmit QoS information related to video fragment priority to the network layer. Second, a second top-down

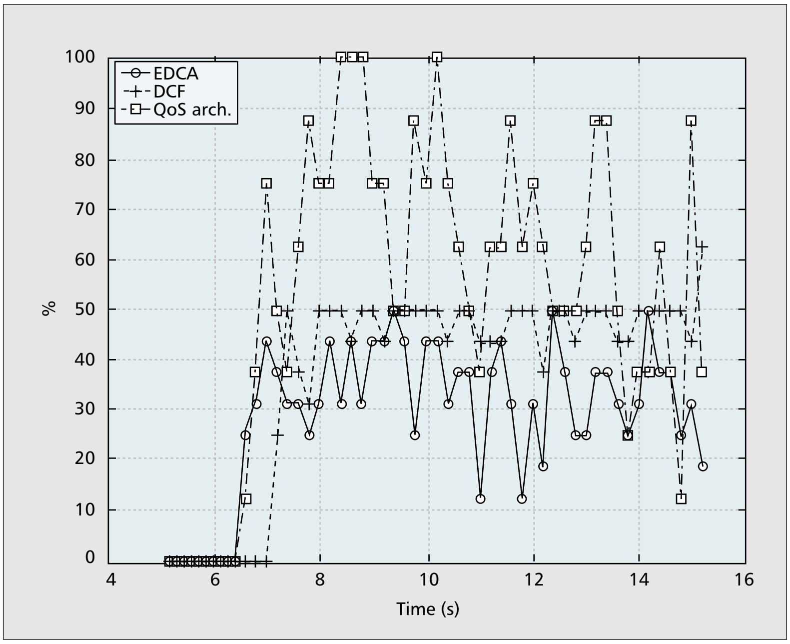


Figure 4. *Partition B loss rate.*

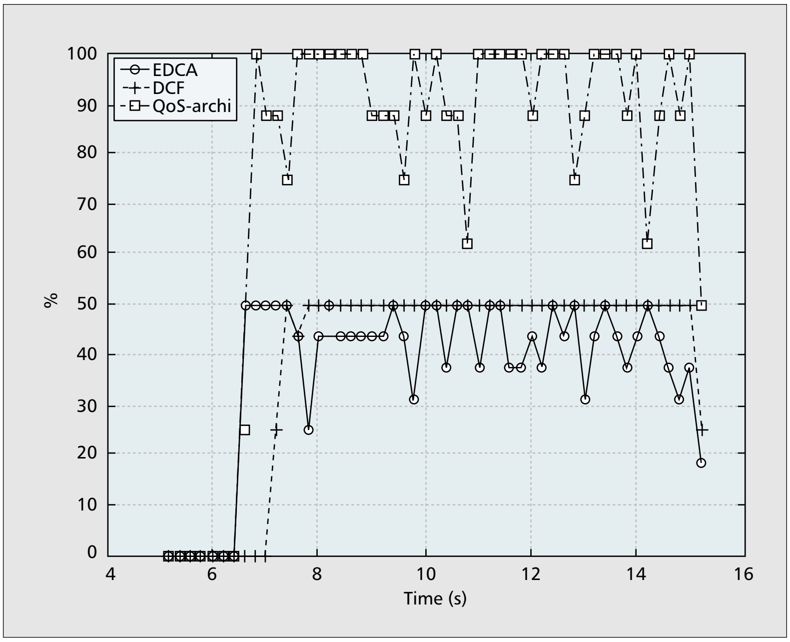


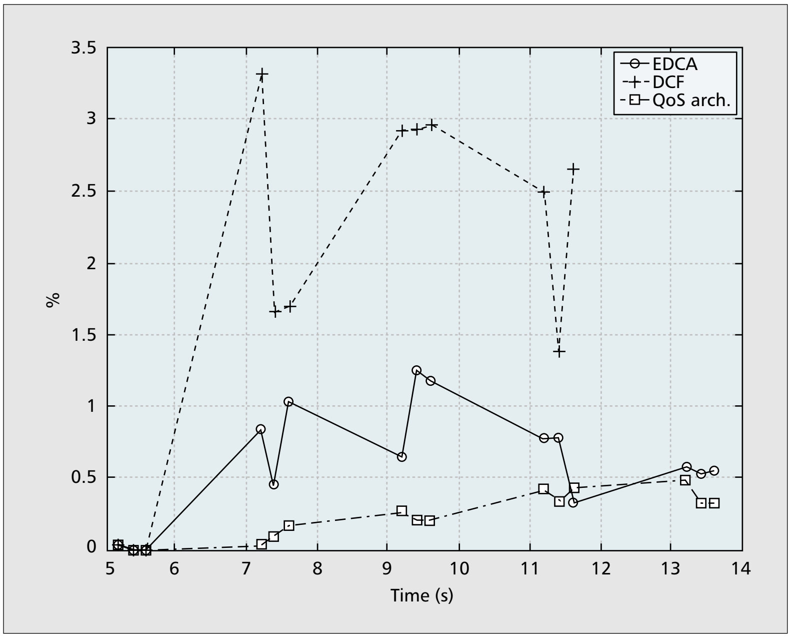
Figure 5. *Partition C loss rate.*

cross-layer interaction allows the network layer, in turn, to express the same QoS exigencies to an EDCA-based MAC layer.

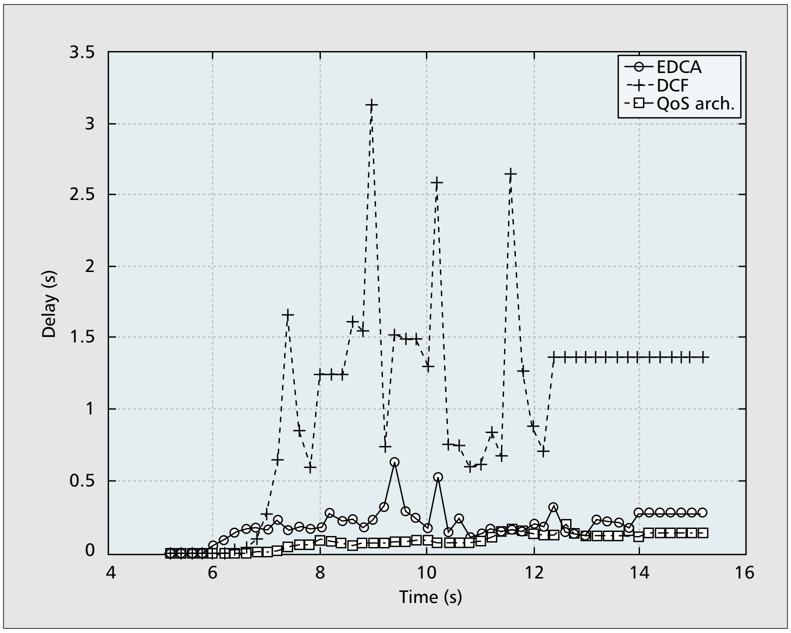
Experimental results show that the proposed architecture achieves better performances in terms of delays and loss rate than the actual WLAN standard and its QoS enhancement mechanism. Through these performance improvements, the cross-layer architecture considerably increases the perceived video quality over that obtained by both DCF and EDCA.

References

[1] IEEE 802.11g 2003, "Part11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification Band," Supp. IEEE 802.11, 2003.



■ **Figure 6.** *IDR packet delays.*



■ **Figure 7.** *Partition A packet delays.*

[2] TGn Sync Technical Specification, http://www.tgnsync.org/techdocs/tgn-sync-proposal-technical-specification.pdf.

[3] IEEE 802.11e, "Wireless LAN Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," 802.11e draft 8.0, 2004.

[4] ITU-T Rec. H.264/ISO/IEC 14496-10 AVC, JVT-G050, Joint Video Team (JVT) ISO/IEC MPEG, ITU-T VCEG, "Final Draft, International Standard on Joint Video Specification," 2003.

[5] ISO/IEC 14496-2, 01 "Coding of Audio-Visual Objects — Part 2: Video — Final Committee Draft," v. 3, Jan. 2001.

[6] A. Ksentini *et al.*, "A New MAC Protocol in IEEE 802.11e Networks for Sensitive Multimedia Applications," *Proc. IEEE GLOBECOM 2005*, St. Louis, MO, Nov. 2005.

[7] A. Majumdar *et al.*, "Multicast and Unicast Real-time Video Streaming over Wireless LANs," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 12, June 2002.

[8] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks," *Proc. IEEE ICME 2002*, Aug. 2002.

[9] M. Van Der Schaar *et al.*, "Cross-Layer Wireless Multimedia Transmission: Challenges, Principles, And New Paradigms." *IEEE Wireless Commun.*, Aug. 2005.

[10] A. Ksentini *et al.*, "Novel Architecture for Reliable H.26L Video Transmission over IEEE 802.11e," *Proc. IEEE PIMRC '04*, Barcelona, Spain, Sept. 2004.

[11] Network Simulator v. 2 EDCA extension, http://www.tkn.tu-berlin.de/research/802.11e_ns2.

[12] H. Schulzrinne *et al.*, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, 1996.

Biographies

ADLEN KSENTINI (Adlen.ksentini@dept-info.u-cergy.fr) received a B.S. degree in computer engineering from USTHB, Algiers, in 2000. He received an M.S in telecommunication and multimedia networking from the University of Versailles in 2002. He obtained his Ph.D. degree in computer science from the University of Cergy-Pontoise in December 2005, on QoS provisioning in IEEE 802.11-based wieless networks. His other interests are in wireless sensors networks and multimedia over wireless networks.

ABDELHAK GUÉROUI (mogue@prism.uvsq.fr) is an associate professor in computer engineering at the University of Versailles, France. His research interests are in the area of wireless networks (WATM, WLAN, and sensors), particularly performance evaluation and QoS provisioning. He received his M.Sc. degree (DEA) from the University of Paris 6 and his Ph.D. in networking and computer engineering from the University of Versailles.

MOHAMED NAIMI (Mohamed.naimi@dept-info.u-cergy.fr) received a Ph.D. degree in computer science from the University of Besançon, France, in 1987. He was an associate professor of computer science at the University of Lille I and Besançon, and a professor at the Polytechnic Institute of Sevenans, France. Since 2000 he has been a professor of computer science at the University of Cergy-Pontoise, France. His research interests are in the areas of distributed systems, networks, and wireless ad hoc and sensor networks. He is the author of a book, *Introduction to Computing* (Hermes, 1992).

■ **Figure 8.** *a) DCF; b) EDCA; c) QoS architecture.*