AUTHOR QUERY FORM

ELSEVIER	**Journal: ADHOC** **Article Number: 1259**	**Please e-mail your responses and any corrections to:** **E-mail: correctionsaptara@elsevier.com**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

Your article is registered as a regular item and is being processed for inclusion in a regular issue of the journal. If this is NOT correct and your article belongs to a Special Issue/Collection please contact s.selvi@elsevier.com immediately prior to returning your corrections.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the '**Q**' link to go to the location in the proof.

Location in article	**Query / Remark: click on the Q link to go** **Please insert your reply or correction at the corresponding line in the proof**
Q1	AU: Please confirm that given names and surnames have been identified correctly.
Q2	AU: Please provide page number in Ref. [26].

Please check this box or indicate your approval if you have no corrections to make to the PDF file	□

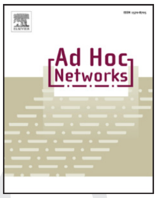
Thank you for your assistance.

Ad Hoc Networks xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Fast retry limit adaptation for video distortion/delay control in IEEE 802.11e distributed networks

Q1 F. Babich, M. Comisso*, R. Corrado

Department of Engineering and Architecture, University of Trieste, Via A. Valerio 10, Trieste, Italy

ARTICLE INFO

Article history:
Received 27 January 2015
Revised 16 June 2015
Accepted 3 July 2015
Available online xxx

Keywords:
Wireless video streaming
802.11e network
Retry limit adaptation

ABSTRACT

This paper presents a fast retry limit adaptation method for video streaming applications over IEEE 802.11e distributed networks. The method enables each source to adapt the number of retransmissions associated to each video packet by relating the perceived distortion to the drop probability and the acceptable delay to the expiration time, without asking the destination for feedback distortion/delay information. The resulting framework, which is based on a simplified but accurate evaluation of the network statistics and of the distortion introduced by the loss of a specific packet, provides a closed-form, and hence computationally cheap, estimation of the retry limit. Furthermore, with respect to most of the existing solutions, the proposed strategy accounts for the impact of the higher priority voice access category (AC), in order to improve the reliability of the retry limit adaptation in the presence of contending ACs. The method is validated by a simulation platform including the physical communication chain and the 802.11e medium access control layer, and its performance is compared to that obtained from an existing solution and from the optimum theoretical settings.

1. Introduction

The possibility to manage video traffic in 802.11 distributed wireless networks represents a challenging task due to the unreliability of the wireless medium and the presence of contention-based mechanisms [1]. These aspects are particularly relevant for streaming applications, which have often to fulfill stringent quality of service (QoS) requirements for satisfying the user's demand [2,3]. Therefore, to introduce QoS control at the medium access control (MAC) layer of an 802.11 network, task group e (TGe) developed the 802.11e amendment, which extends the functionalities of the legacy distributed coordination function (DCF) by adopting the enhanced distributed channel access (EDCA) [4]. The EDCA enables a prioritization of the traffic during the contention period by defining four access categories (ACs): voice (VO), video (VI), best effort (BE), and background (BK), whose differentiation is based on four parameters: transmission opportunity, arbitration inter-frame space (AIFS), minimum and maximum contention windows. The 802.11e extension establishes the values of these parameters according to the AC and the physical (PHY) layer technology available among those adopted in the 802.11a/b/g/n amendments.

Even if the EDCA settings are specified to provide a higher priority to the VO and VI ACs, collisions involving audio/video packets may still occur, thus making necessary a proper policy of management of the retransmissions. Accordingly, several studies have investigated this issue, by proposing useful methods for optimizing the retry limit associated to each video packet [5–16]. The main objective of these methods is the adaptation of the number of retransmissions to the acceptable delay and the perceived distortion. This leads to the derivation of elaborate optimization strategies, able to provide significant performance improvements with respect to those achievable using the 802.11e default settings. However, two relevant aspects are often neglected in these proposals.

* Corresponding author. Tel.: +390405583451.
E-mail address: mcomisso@units.it (M. Comisso).

http://dx.doi.org/10.1016/j.adhoc.2015.07.008
1570-8705/© 2015 Published by Elsevier B.V.

First, the presence of the higher priority VO AC, which can considerably reduce the access opportunities for the video packets. Second, the complexity of the conceived solution, which may be difficult to implement on the commercially available 802.11 network interface cards, that are characterized by low computational resources. Therefore, an alternative approach may be developed by moving from a more realistic model in which both the VO and VI ACs can be active, and considering the satisfaction of the QoS requirements together with a minimization of the necessary calculations.

The retry limit adaptation method proposed in this paper deals with these two issues. The method, which is derived by relating the drop probability to the video distortion, and the packet delay to the expiration time, explicitly accounts for the impact of higher priority VO AC on the evolution of the video transmission. Additionally, both the distortion estimation algorithm and the retransmission strategy are developed with the purposes of limiting the computational cost and of not requiring feedback distortion/delay information from the destination. A theoretical evaluation of the optimum retry limit and an existing retransmission strategy are used as benchmarks for validating the performance of the presented method, which is implemented in a network simulation platform including the physical communication chain and the 802.11e EDCA.

The paper is organized as follows. Section 2 presents the literature overview. Section 3 introduces the analyzed system. Section 4 describes the adopted theoretical model and the conceived adaptation algorithm. Section 5 discusses the numerical results. Section 6 summarizes the most relevant conclusions.

2. Related work

The interest in the development of an optimization strategy for the retry limit derives, firstly, from the influence of this parameter on the performance figures of the network (throughput, successful packet delay, drop probability), and, secondly, from the absence of mandatory specifications for its setting. In particular, this second aspect guarantees a certain flexibility to the designer, which instead is not guaranteed for the other EDCA parameters (transmission opportunity, AIFS, minimum and maximum contention windows), whose values are specified by the 802.11e standard according to the adopted PHY layer extension [4]. Moreover, this flexibility becomes more relevant when streaming applications are involved, since the possibility to associate a different number of retransmissions to a different packet allows the designer to better match the QoS requirements in the presence of video traffic flows.

Accordingly, several retry limit adaptation methods have been proposed in the research literature [5–16]. The QoS strategy presented in [5] adopts a priority queueing also at the network layer, in order to relate the adaptation to the tradeoff between drop probability and buffer overflow rate. The optimal retry limit estimation in [6] derives from a minimization of the total expected distortion relying on classification and machine learning techniques. In [7] the conventional count-based retransmission scheme is replaced by a time-based one, in which the deadline is determined by the expiration time and the importance of the inter-coded frames. A retry limit adaptation method for scalable videos is developed in [8] by considering the collision probability as a load indicator. The reciprocal influence among the nodes and the ACs on the selection of the retry limit is analyzed in [9], where an adaptive algorithm is derived from the numerical solution of a nonlinear system. A cross-layer content-aware scheme for scheduling the retransmissions is proposed in [10] by considering the estimated backoff time and the macroblock-level loss impact. Closed-form estimations of the retry limit in the presence of collisions and buffer overflows are obtained in [11] by modeling the 802.11 MAC layer as an M/G/1 queueing system. The concept of virtual buffer size is introduced in [12] to develop an adaptation strategy for delay-critical video transmissions in lossy networks. A video-coding aware MAC layer is proposed in [13], with the purpose of delivering a video stream in which the retry limit is adjusted to guarantee a delay reduction and a satisfactory peak signal-to-noise ratio (PSNR). A fragment-based retransmission scheme suitable for video traffic is developed in [14], where the aim is to decrease the duration of the retransmission attempts in the presence of channel errors. In [16] the mean square error and the structural similarity are compared as video quality assessments for developing adaptive retransmission strategies. A tradeoff between energy efficiency and satisfaction of the QoS requirements for centralized operations is obtained in [15] by a joint dynamic adjustment of the contention window and of the retry limit.

This overview shows that the available adaptation policies for the retry limit of the VI AC in distributed environment are developed with the aim of satisfying two main objectives: management of the distortion and control of the delay. Except for [9], the proposals are conceived assuming the absence of the VO AC, and are not focused on the limitation of the computational complexity. The aim of the strategy presented in this paper is to provide an adaptive algorithm able to account for the distortion/delay requirements of the transmitted video sequence, considering, as additional purposes, limitation of the processing time and possibility to operate in the presence of higher priority traffic.

3. System description

Consider the MAC layer of an 802.11e distributed network, and hence a single-hop scenario involving N sources and the corresponding N destinations. All the $2N$ nodes operate using the EDCA basic access mechanism combined with an 802.11g PHY layer. Each source S contends with the other sources for gaining access to the wireless medium in order to deliver its packets to the intended destination D. Except for the mandatory ACKnowledgement (ACK) packet, the destination does not provide any feedback information concerning the distortion and the delay, which hence must be estimated by the source on its own. In particular, S can support four ACs, which are numbered according to $q = 1$ (VO), $q = 2$ (VI), $q = 3$ (BE), $q = 4$ (BK), thus indicating that a lower q value identifies a higher priority. Assume that each AC of each source remains nonempty once a packet is successfully transmitted, hence considering, as in [8,10], saturated traffic conditions. For the BE and BK ACs, the saturation assumption is widely accepted, since it derives from usual file transfer applications. For the VO and VI ACs, the saturation

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

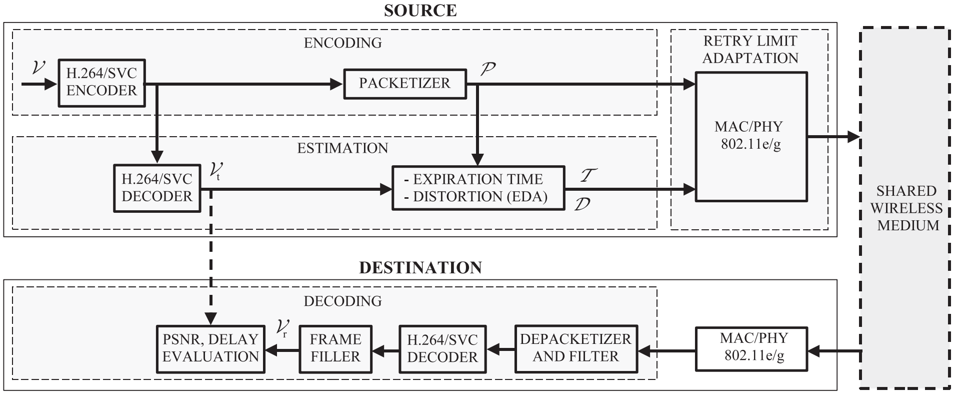


Fig. 1. Model for the generic source–destination pair.

hypothesis is justified by the transmission policy usually adopted by many common streaming services, such as YouTube, according to which the packets corresponding to the first 40 s of a requested stream are immediately sent, while the sending rate adopted for the rest of the stream must be slightly higher than the playback rate, since the objective is to avoid the interruption of the reproduction [17]. This policy implies that a large amount of packets may be considered already present also at the transmission queues corresponding to the VO and VI ACs, thus allowing to assume a saturated scenario.

Among the entire load that must be delivered to D, S has to transmit a video sequence $\mathcal{V} = \{\nu_l : l = 1, \dots, L\}$, which includes L frames $\nu_1, \dots, \nu_L$. The source–destination model is reported in Fig. 1, where four operations are considered: the encoding of $\mathcal{V}$, the estimation of the distortion and of the expiration time, the adaptation of the retry limit, and the decoding of the received video. The first three operations are carried out by the source, while the latter one by the destination. The three following subsections describe the encoding, the estimation, and the decoding operations, while the retry limit adaptation is presented in Section 4.

3.1. Encoding

The video $\mathcal{V}$ is encoded using the H.264 scalable video coding (SVC) standard developed by the joint video team (JVT) [18]. Thus, $\mathcal{V}$ is subdivided into groups of pictures (GOPs) of size α, and encoded to obtain a set of network abstraction layer units (NALUs). Each NALU, which is created considering the dependencies within a GOP, is classified according to the type of the corresponding frame: Intra-coded (I), Predictively-coded (P), and Bipredictively-coded (B), and is generated to encode the video as an independently decodable base layer and a certain number of enhancement layers [19]. The set of NALUs, which are of different size, is then packetized to obtain a set $\mathcal{P}$ of K packets $\pi_1, \dots, \pi_K$ of equal size that are transmitted over the network. Thus, at the end of the packetization process, the encoded version of a generic frame ν_l may be fragmented in a certain number of 802.11 packets. For calculation purposes, it may be then useful to define, for each $\nu_l \in \mathcal{V}$, the set $\mathcal{P}_l (\subset \mathcal{P})$ of the packets containing the NALUs of ν_l. In particular, $\mathcal{P}_l$ has k_l elements and hence the overall number of packets K that derives from the encoding of the original video sequence $\mathcal{V}$ can be expressed as $K = \sum_{l=1}^{L} k_l$. By consequence, the set $\mathcal{P}_l$ contains the packets having indexes k between $K_{l-1} + 1$ and K_l, where $K_l = \sum_{l'=1}^{l} k_{l'}$, thus $\mathcal{P}_l = \{\pi_k : k = K_{l-1} + 1, \dots, K_l\}$.

3.2. Estimation

Since each NALU may have a different impact on the overall video quality, the retry limit for each packet π_k should be selected according to its expiration time T_{e_k} and to the distortion D_k produced by its possible loss. It is useful to first relate these two quantities to the frames, which represent the real video content perceived by the final user, and subsequently refer them to the packets. As a first step, the NALUs generated by the H.264/SVC encoder are decoded to obtain the sequence $\mathcal{V}_t = \{f_l : l = 1, \dots, L\}$, which may be considered as a reference sequence that is physically transmitted over the network. This sequence will be compared with the video received at the destination (Fig. 1), in order to enable a performance evaluation of the proposed framework that accounts for the losses due to the sole access procedure, and not for the lossy compression, whose effects are out of the scope of this paper.

Let us now consider the expiration time. To this aim, one may observe that usually the player at the destination awaits the reception of a certain number of frames $\bar{l}$ before starting the play of the video. From now on, $\bar{l}$ will be referred to as the expiration time index. Therefore, one can assume that the requirement on the expiration time holds for the frames successive to a frame $f_{\bar{l}}$, while the frames previous to $f_{\bar{l}}$ may be associated to an infinite expiration time. Accordingly, the expiration time for the lth frame can be evaluated as [7,12]:

$$\tilde{T}_{e_l} = \begin{cases} +\infty & l = 1, \dots, \bar{l} \\ (l + M_l) T_f & l = \bar{l} + 1, \dots, L \end{cases} \qquad (1)$$

where M_l is the number of frames inter-coded with f_l in the same GOP and T_f is the inter-frame interval. The quantity in (1) is considered to control the delay at the destination, in order to limit the interruptions of the playback of the video due to the wait of the arriving frames.

From a practical point of view, the most accurate method for estimating the impact of the loss of a packet on the corresponding GOP would require the removal of the packet

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

itself and the subsequent decoding of the entire GOP according to the adopted error concealment strategy [10]. Since this method would be computationally too expensive, alternative approaches have been derived [20,21]. In particular, the existing distortion estimation techniques for H.264 encoded videos may be classified in two families: lightweight methods and sophisticated methods [22]. The lightweight methods are computationally cheap, since they just distinguish between key and non-key frames [23]. However, they do not provide a fine estimation of the distortion effect determined by the loss of a frame, thus making preferable the sophisticated methods when more accurate estimations are necessary. Among this second family of distortion estimation techniques [24–28], which are often characterized by a high computational cost, the exponential distortion algorithm (EDA), presented in [26,29,30], is one of the few sophisticated algorithms that enables to model the distortion due to the loss of a frame f_l maintaining a low complexity. For this reason, the EDA is adopted in this paper.

The EDA assumes the adoption of a frame copy error concealment at the decoder, since the lost frame f_l is replaced by the previous received one f_{l-1}. Therefore, considering the frame f_l and a succeeding one $f_{l'}$, both belonging to the same GOP, the EDA estimates the distortion suffered by $f_{l'}$ because of the loss of f_l as the product $\mathrm{MSD}[f_l - f_{l-1}]e^{-\xi(l'-l)}$, where $\mathrm{MSD}[f_l - f_{l-1}]$ is the mean square difference between f_l and f_{l-1} that estimates the actual mean square error at the decoder, and ξ is a parameter dependent on the encoded video that accounts for the error propagation effect. Using this approach, the distortion on the entire GOP of size α due to the loss of the frame f_l can be evaluated as:

$$\tilde{D}_l = \sum_{l'=l}^{\lceil \frac{l}{\alpha} \rceil \alpha} \mathrm{MSD}[f_l - f_{l-1}]e^{-\xi(l'-l)}, \tag{2}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Further details concerning the EDA can be found in [26,29,30].

The two sequences of estimations $\tilde{T}_{e_1}, \dots, \tilde{T}_{e_L}$ and $\tilde{D}_1, \dots, \tilde{D}_L$, which are related to the frames, must be then related to the packets. To this aim, one may observe that, if a frame f_l, with $l > \bar{l}$, is characterized by the expiration time $\tilde{T}_{e_l}$ and by the set $\mathcal{P}_l$, the corresponding k_l packets would not all be associated to the same $\tilde{T}_{e_l}$ value. In fact, in this case the transmission of the first packet of $\mathcal{P}_l$ might use all the time margin, subsequently forcing the remaining packets, having the same expiration time, to adopt a retry limit equal to zero. To avoid the occurrence of this event, the interval $\tilde{T}_{e_l} - \tilde{T}_{e_{l-1}}$, theoretically available for the transmission of the k_l packets corresponding to the lth frame, is subdivided into k_l equal subintervals. Therefore, recalling (1), the expiration time associated to $\pi_k \in \mathcal{P}_l$ remains infinite for $k = K_{l-1} + 1, \dots, K_l$ and $l = 1, \dots, \bar{l}$, while it is evaluated as:

$$T_{e_k} = \frac{\tilde{T}_{e_l} - \tilde{T}_{e_{l-1}}}{k_l}(k - K_{l-1}) + \tilde{T}_{e_{l-1}}, \tag{3}$$

for $k = K_{l-1} + 1, \dots, K_l$ and $l = \bar{l} + 1, \dots, L$. The linearized approach in (3) aims to fairly subdivide the time available to transmit a frame among all packets containing NALUs that belong to that frame.

The distortion corresponding to $\pi_k \in \mathcal{P}_l$ can be calculated by considering that associated to the frame f_l normalized to the maximum, thus:

$$D_k = \frac{\tilde{D}_l}{\max\limits_{l\in\{1,\dots,L\}} \tilde{D}_l}, \tag{4}$$

for $k = K_{l-1} + 1, \dots, K_l$ and $l = 1, \dots, L$. Observe that, since the indexing in k is related to the indexing in l, all packets $\pi_k \in \mathcal{P}_l$ associated to a frame f_l have an identical normalized distortion. For this reason, the index k does not explicitly appear in the right hand side of (4). The motivation for the normalization in (4) can be explained observing that D_k will be related to the drop probability, thus it is useful to identify a measure of the distortion lying between 0 and 1. As it will be explained in Section 4.2, D_k may be further scaled according to the specific application, if required. However, the availability of a normalized quantity may represent a reasonable starting point for the subsequent exploitation of the distortion, also in the case in which other estimation techniques are adopted. Summarizing, the process of estimation carried out at the source S provides, for the set of packets $\mathcal{P}$, the two sets of estimations $\mathcal{T} = \{T_{e_1}, \dots, T_{e_K}\}$ and $\mathcal{D} = \{D_1, \dots, D_K\}$ that will be used at MAC layer to adapt the retry limit of the VI AC.

3.3. *Decoding*

At the destination, the received packets are depacketized to derive the set of the received NALUs, which are filtered to remove, firstly, all NALUs that have been at least partially lost due to the loss of the corresponding packets and, secondly, the NALUs relative to frames whose base layer has not been received and hence cannot be decoded [19] (Fig. 1). The set of remaining NALUs is passed to the H.264/SVC decoder and the result is filled with the lost frames, thus obtaining the received video $\mathcal{V}_\mathrm{r}$, which is compared to the transmitted video $\mathcal{V}_\mathrm{t}$. The filling and comparison operations, which would not be carried out in a real network, are performed just for modeling purposes, in order to enable a frame-by-frame comparison between the reference video $\mathcal{V}_\mathrm{t}$ and the received one $\mathcal{V}_\mathrm{r}$, so as to evaluate the PSNR and the delay for the decodable frames. As discussed at the beginning of Section 3.2, this comparison enables to isolate the effects due to the 802.11 dropped packets from those due to the lossy compression.

4. Retry limit adaptation

A reliable model for a distributed network may be derived adopting a Markov approach [31,32], which has been extensively used to investigate the performance of an 802.11-based uncoordinated network in several contexts, including the presence of non-saturated conditions [33,34], directional communications [35], multiple ACs [36], and heterogeneous traffic sources [37]. Accordingly, the here developed adaptation strategy for the retry limit associated to each packet $\pi_k \in \mathcal{P}$ is based on a Markov model of the 802.11e EDCA [36]. This model, which has been validated by experimental measurements realized using a real testbed, is properly re-elaborated to obtain a reduced set of simplified equations that enable to reliably estimate the network behavior with a low computational cost. The next subsection introduces this Markov model with the purpose of briefly summarizing the

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

approach presented in [36], so as to better identify the mathematical context from which the proposed algorithm is derived. Subsequently, Section 4.2 presents, as the main contribution, the developed retry limit adaptation strategy.

4.1. Theoretical model

According to the 802.11e EDCA specifications and the network scenario described in Section 3, in the presence of equal transmission opportunities the qth AC of the generic source can be characterized by the AIFS AIFS$_q$, the minimum contention window W_q, the retry limit m_q, and the maximum backoff stage m'_q, which determines the maximum contention window. The function of these parameters can be explained by describing the EDCA backoff procedure.

When a packet belonging to the qth AC is ready for transmission, the source monitors the medium for a time AIFS$_q$. If the medium is sensed idle during this time, the packet is immediately transmitted, otherwise a random backoff is generated as the product between a constant slot time and a random integer n uniformly distributed in the interval $[0, W_q - 1]$. This backoff is inserted in a reverse counter, which is decreased when the medium is sensed idle, frozen when the medium is sensed busy, and reactivated when the medium is sensed idle again for an AIFS$_q$. When the counter reaches the zero value the packet is transmitted. If the transmission is successful, the source receives an ACK packet from the destination. Otherwise, a retransmission is scheduled by updating the retry counter and the contention window. In particular, at the ith retransmission attempt, the contention window is evaluated as:

$$W_q^i = 2^{\min(i, m'_q)} W_q, \tag{5}$$

and the backoff decrease process is repeated adopting a random integer n uniformly distributed in the interval $[0, W_q^i - 1]$. When the retry counter reaches the retry limit m_q, the packet is discarded. With reference to the qth AC and assuming identical AIFS values for the four ACs, this mechanism can be modeled considering the Markov chain in Fig. 2, where p_q denotes the conditional collision probability. This figure describes the backoff procedure by a two-dimensional process in which the generic state (i, n) identifies, for a generic packet, a residual backoff of n slots at the ith transmission attempt. According to the scenario introduced in Section 3, the model assumes saturated traffic conditions, since, once a packet is successfully transmitted or is discarded due to the achievement of the maximum number of retransmissions, a novel packet is immediately available. Observe that the assumption of identical AIFS values, which may seem to limit the applicability of the analysis, is acceptable in the case considered in this study. The suitability of this assumption will be justified is detail in Section 4.2.

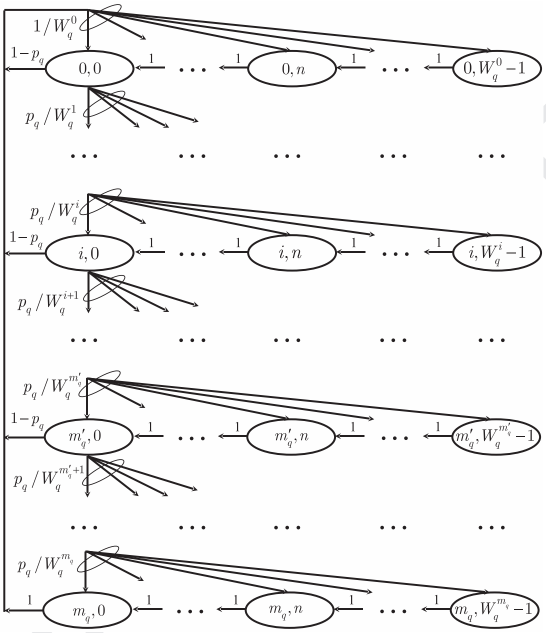


Fig. 2. Markov model for the qth AC of the generic source.

Analyzing the chain in Fig. 2, one can express the generic steady-state probability $\eta_{i,n}$ as a function of $\eta_{0,0}$, hence obtaining [36]:

$$\eta_{i,n} = \left(1 - \frac{n}{W_q^i}\right) p_q^i \eta_{0,0}, \tag{6}$$

for $n \in [0, W_q^i - 1]$ and $i \in [0, m_q]$. Therefore, using (6) and imposing the normalization condition, one obtains:

$$\eta_{0,0} = \left[\sum_{i=0}^{m_q} \sum_{n=0}^{W_q^i - 1} \left(1 - \frac{n}{W_q^i}\right) p_q^i\right]^{-1}. \tag{7}$$

The probability τ_q that the source attempts the transmission can then be evaluated by summing over all the steady-state probabilities with backoff equal to zero, thus:

$$\tau_q = \sum_{i=0}^{m_q} \eta_{i,0}. \tag{8}$$

Using (5)–(7) in (8) and performing some algebra, one can therefore obtain the first set of equations of the system:

$$\begin{cases} \tau_q = \dfrac{2(1-2p_q)\left(1 - p_q^{m_q+1}\right)}{(1-2p_q)\left[1 - p_q^{m_q+1} + p_q W_q 2^{m'_q}\left(p_q^{m'_q} - p_q^{m_q}\right)\right] + W_q(1-p_q)\left[1 - (2p_q)^{m'_q+1}\right]} \\ p_q = 1 - \prod_{q'=1}^{4} (1 - \tau_{q'})^{N-1} \prod_{q'=1}^{q-1} (1 - \tau_{q'}) \end{cases} \tag{9}$$

which is defined for $q = 1, \ldots, 4$, and hence consists of $2 \times 4 = 8$ equations. The second set of equations in (9) expresses the conditional collision probabilities p_q for $q = 1, \ldots, 4$, according to the fact that a packet belonging to the qth AC of a given source S collides in two cases. First, if S and at least another source transmit their packets at the beginning of the same slot time (external collision). Second, if, at the source S, the backoff of the elaborated packet and that of a packet belonging to an AC with a higher priority reach the zero value at the same time (internal collision). In this second case the collision is directly resolved at the source S by allowing the transmission of the packet with the higher priority and considering as collided the packet with the lower priority. Further mathematical details for the derivation of (9) can be found in [36]. The nonlinear system of eight equations in (9) represents the core of the model, since it enables the calculation of the transmission and collision probabilities for the ACs of interest. The parameters W_q and m'_q for $q = 1, \ldots, 4$ are assumed known, since they are specified in the 802.11e standard for a given PHY layer [4]. Instead, the parameter m_q and the quantities τ_q and p_q for $q = 1, \ldots, 4$ are assumed unknown, thus $3 \times 4 = 12$ unknowns are present in (9). For the case $q = 2$, corresponding to the VI AC that is of interest in this study, m_2 depends also on the specific video packet π_k. However, to simplify the notation, this dependence will be explicitly introduced afterwards, thus currently considering the network behavior for a given video packet.

Moving from (9), the proposed approach for the derivation of a retry limit adaptation algorithm first considers the collision and transmission probabilities of the active ACs, from which the drop probability and the packet delay are estimated. Subsequently, the retry limit for each packet is evaluated by relating the drop probability to the distortion, and the packet delay to the expiration time.

4.2. Adaptation algorithm

The first approximation introduced to simplify (9) relies on the practical observation that the main impact on the collision probability of a given AC is due to the ACs having a higher or equal priority, and hence a higher or equal transmission probability [36]. The 802.11e standard states that, when a PHY layer specifies a minimum contention window and a maximum backoff stage, these parameters hold for the BK AC. Thus, they must be intended as W_4 and m'_4, respectively, and must be used to obtain the corresponding parameters for all the ACs as $W_1 = W_2/2 = W_3/4 = W_4/4$, $m'_1 = m'_2 = 1$, $m'_3 = m'_4$ [4], in order to provide a higher priority to the VO and VI ACs. For the same reason, in the 802.11e extension, the AIFS values are selected as $\mathrm{AIFS}_1 = \mathrm{AIFS}_2 > \mathrm{AIFS}_3 > \mathrm{AIFS}_4$ [4]. Since the higher the AIFS, the minimum contention window, and the maximum backoff stage, the lower the transmission probability, these settings imply that $\tau_3, \tau_4 < \tau_2 < \tau_1$. This allows one to neglect, on first approximation, the impact of the BE and BK ACs on the remaining ones, thus assuming $\tau_3, \tau_4 \cong 0$, and the impact of the VI AC on the VO one. Besides, being in this study the interest focused on the VI AC, the equations expressing p_3 and p_4 as functions of the transmission probabilities in (9) can no longer be considered. Observe that, since $\mathrm{AIFS}_1 = \mathrm{AIFS}_2$, the Markov chain in Fig. 2 properly describes the behavior of both the VO and VI ACs without the need of additional states, which instead would be required if the interest had been focused also on the BE and BK ACs, in order to account for the larger AIFS_3 and AIFS_4 values [38]. Regarding the parameter settings specified in the 802.11e amendment for the VO and VI ACs, a final aspect that is worth noticing concerns the low number of possible contention windows enabled by the unity backoff stage, that is, $W_1^0 = W_1$ and $W_1^1 = 2 \cdot W_1$ for the VO AC, and $W_2^0 = W_2 = 2 \cdot W_1$ and $W_2^1 = 2 \cdot W_2 = 4 \cdot W_1$ for the VI AC, which lead to just three possible contention windows. This has a relevant consequence, since, once a given source has established its retry limit, the retry limits selected by the other sources have a limited influence on the collision probability of that source, because, after the first transmission attempt, the contention window remains identical for all the subsequent attempts. To better clarify this issue, consider, as a theoretical reference, the limiting case in which just the VI AC is active and the backoff stage is equal to zero. This is a perfectly homogeneous case, since, regardless of the retry limit value selected by each source, the backoff time is randomly selected in an identical interval. Hence, the collision probability of the single source is insensitive to the retry limits selected by the other sources. The unity backoff stage, even if not guarantees this complete insensitivity, however maintains the sensitivity very low, thus justifying the implicit assumption adopted in the formulation developed in (9), according to which each source selects its m_2 value considering that the other sources adopt the same one. The reliability of this hypothesis will be further explored in Section 5.2.4, where a network scenario involving sources transmitting different video sequences, and hence adopting different retry limits, will be considered.

Exploiting the above arguments, whose reliability will be also investigated in Section 5 considering scenarios in which all the four ACs are active, (9) can be replaced by a reduced nonlinear system of four equations:

$$\begin{cases} \tau_q = \dfrac{2\left(1 - p_q^{m_q+1}\right)}{(2W_q + 1)\left(1 - p_q^{m_q+1}\right) - W_q(1 - p_q)}, \quad q = 1, 2 \\ p_1 = 1 - (1 - \tau_1)^{N-1} \\ p_2 = 1 - (1 - \tau_1)^{N}(1 - \tau_2)^{N-1} \end{cases} \tag{10}$$

Then, in the first two equations of (10), τ_q may be approximated by a second-order polynomial passing through the points $(0, \tau_{q_1})$, $(1/2, \tau_{q_2})$, and $(1, \tau_{q_3})$, where:

$$\tau_{q_1} = \tau_q\big|_{p_q=0} = \frac{2}{W_q + 1}, \tag{11a}$$

$$\tau_{q_2} = \tau_q\big|_{p_q=1/2} = \frac{2^{m_q} \cdot 4 - 2}{2^{m_q}(2 + 3W_q) - 2W_q - 1} \xrightarrow[m_q \to \infty]{} \frac{4}{3W_q + 2}, \tag{11b}$$

$$\tau_{q_3} = \lim_{p_q \to 1} \tau_q = \frac{2m_q + 2}{m_q(2W_q + 1) + W_q + 1} \xrightarrow[m_q \to \infty]{} \frac{2}{2W_q + 1}, \tag{11c}$$

are evaluated for $m_q \to \infty$ when a dependence on m_q is present. Thus, τ_q can be approximated by:

$$\tau_q \cong a_q p_q^2 + b_q p_q + c_q, \tag{12}$$

where the coefficients:

$$a_q = \frac{4W_q^2}{6W_q^3 + 13W_q^2 + 9W_q + 2}, \tag{13a}$$

$$b_q = -\frac{2W_q(5W_q+2)}{6W_q^3+13W_q^2+9W_q+2}, \tag{13b}$$

$$c_q = \frac{2}{W_q+1}, \tag{13c}$$

depend only on the minimum contention window of the qth AC. These coefficients are obtained by fitting (12) through the three points $(0, \tau_{q_1})$, $(1/2, \tau_{q_2})$, and $(1, \tau_{q_3})$, that is, substituting the p_q and τ_q coordinates in (12) for each point, and then solving the resulting linear system of three equations in the three unknowns a_q, b_q, and c_q. Concerning the approximations in (11) and (12) for the evaluation of τ_q, it is worth remarking that the aim of the proposed strategy is to estimate, first, the drop probability and the delay for the VI AC through p_2, and, subsequently, the retry limit according to the video distortion and the expiration time. Solving directly (10) would lead to a p_2 value dependent on m_2 and hence to a higher computational cost. Instead, the solution of a polynomial equation, obtained removing the dependence on m_2, can rely on efficient root-finding techniques. Moreover, one can easily prove, by performing a simple derivative, that τ_q in (10) is a monotonically decreasing function of m_q. Hence, since (12) is obtained for $m_q \to \infty$, it provides an estimate from below of τ_q, which leads to an underestimation of p_q and, in turn, of the drop probability $p_q^{m_q+1}$. This behavior, combined with the requirement that the drop probability of a packet be inversely proportional to the distortion introduced by its loss, guarantees a small (conservative) overestimation of the distortion's effect, thus explaining the reason for the use of an approximation based on an infinite retry limit in (11b) and (11c).

Given $\tau_q < 1$, each term $(1-\tau_q)^{N-1}$ in the two latter equations of (10) can be approximated by truncating the corresponding binomial expansion to a suitable value $\bar{N}(\leq N-1)$. Using this approximation and substituting (12) for $q = 1$ in the third equation of (10), one obtains the polynomial equation:

$$\sum_{j=0}^{\bar{N}} \binom{\bar{N}}{j} (-1)^j (a_1 p_1^2 + b_1 p_1 + c_1)^j + p_1 - 1 = 0, \tag{14}$$

whose solution $\bar{p}_1 \in [0, 1]$ is the approximated conditional collision probability for the VO AC. Similarly, substituting (12) for $q = 2$ in the fourth equation of (10), and defining $\bar{\tau}_1 = a_1\bar{p}_1^2 + b_1\bar{p}_1 + c_1$, one can use the binomial approximation to derive a second polynomial equation:

$$(1-\bar{\tau}_1)^N \cdot \sum_{j=0}^{\bar{N}} \binom{\bar{N}}{j} (-1)^j (a_2 p_2^2 + b_2 p_2 + c_2)^j + p_2 - 1 = 0, \tag{15}$$

which provides, for the VI AC, the approximated conditional collision probability $\bar{p}_2 \in [0, 1]$ and subsequently the approximated transmission probability $\bar{\tau}_2 = a_2\bar{p}_2^2 + b_2\bar{p}_2 + c_2$. Thus, the original problem of evaluating the transmission and collision probabilities for the VO and VI ACs by the nonlinear system in (9) is considerably simplified by the replacement with the two polynomial equations in (14) and (15). Observe that this simplification may be maintained even if the number of nodes is high. In fact, the degree of the polynomial in (14) and (15) depends on the parameter $\bar{N}$, which may be selected lower than $N-1$ maintaining an acceptable accuracy, since $\bar{\tau}_q^j$ becomes less significant as j becomes larger. Furthermore, it is worth noticing that the current 802.11 PHY layer extensions lead to very low minimum and maximum contention windows for the VO and VI ACs. Hence, the number of collisions grows rapidly with the increase of N, thus making difficult the support of many contending video flows of acceptable quality. By consequence, the selection of the maximum value $\bar{N} = N-1$ in (14) and (15), which provides the highest accuracy, may be acceptable in practical scenarios, where, realistically, the number of contending video flows is limited.

Once the four probabilities $\bar{p}_q, \bar{\tau}_q$ for $q = 1, 2$ are estimated, one can derive the performance figures of the network as a function of the retry limit. Remembering that $m_2' = 1$ and introducing the notation $m_{2,k}$ to identify the dependence of the retry limit for the VI AC on the generic video packet $\pi_k \in \mathcal{P}$, the average packet delay can be expressed as [36]:

$$T(m_{2,k}) = E_s \cdot E_{ns}(m_{2,k}), \tag{16}$$

where E_s is the average time required for a decrease of the backoff counter and:

$$E_{ns}(m_{2,k}) = \sum_{i=0}^{m_{2,k}} \frac{W_2^i - 1}{2}\bar{p}_2^i = \frac{2W_2-1}{2} \cdot \frac{1-\bar{p}_2^{m_{2,k}+1}}{1-\bar{p}_2} - \frac{W_2}{2}, \tag{17}$$

is the average number of backoff decreases for the $m_{2,k}$ retransmissions of π_k. Recalling that, using the basic access, the transmission time $\bar{T}$ for a success and a collision is the same, E_s is given by the sum of the fractions of time wasted because of inactivity and used for transmission (successful or not), thus:

$$E_s = \varsigma + \left\{1 - [(1-\bar{\tau}_1)(1-\bar{\tau}_2)]^N\right\}(\bar{T} - \varsigma), \tag{18}$$

where ς is the slot time specified by the adopted PHY layer standard and the term $1 - [(1-\bar{\tau}_1)(1-\bar{\tau}_2)]^N$ denotes the probability that at least one packet is transmitted [36]. Furthermore, the transmission time $\bar{T}$ in (18) can be calculated as:

$$\bar{T} = \frac{\bar{\Lambda}}{R} + \frac{\mathrm{H} + \mathrm{ACK}}{R_c} + \mathrm{SIFS} + \mathrm{AIFS}_2, \tag{19}$$

where $\bar{\Lambda}$ is the length of the payload averaged over the VO and VI ACs, R is the data rate, H is the length of the MAC/PHY headers of the DATA packet, ACK is the length of the ACK packet, R_c is the control rate, SIFS is the short inter-frame space, and $\mathrm{AIFS}_2 = \mathrm{SIFS} + 2\varsigma$ [4]. The second fundamental performance figure that is required to estimate the network behavior when a video sequence has to be transmitted is the drop probability, which can be evaluated as:

$$p_{\mathrm{drop}}(m_{2,k}) = \bar{p}_2^{m_{2,k}+1}. \tag{20}$$

Now, the problem's requirements can be imposed by relating the drop probability to the distortion and the delay to the expiration time. In particular, the packets that lead to a higher distortion in the case of loss should be associated to a lower drop probability and hence to a higher retry limit, while the packets determining a lower distortion should be associated to a higher drop probability and hence to a lower retry limit. Moreover, the distortion should be not only inversely proportional to the drop probability, but should recall the exponential relationship in (20) between p_{drop} and $m_{2,k}$, in order to provide an effective adaptation [16]. Therefore, imposing as a further requirement that the sum of the delays derived by (16)–(18) for the first k packets be lower than the expiration time of the kth packet, the following minimization problem in the unknown $m_{2,k}$ can be formulated for each $\pi_k \in \mathcal{P}$:

$$\arg\min_{m_{2,k}\in\mathbb{N}} \left|p_{\text{drop}}(m_{2,k}) - 10^{-\zeta D_k}\right|, \tag{21}$$

$$\text{subject to}: \sum_{k'=1}^{k} T(m_{2,k'}) \le T_{e_k}, \tag{22}$$

where $\zeta(>0)$ is a parameter introduced to better manage the relationship between drop probability and distortion, whose impact on the estimation process will be discussed at the beginning of the next section. The objective of the problem in (21) and (22) is to find, for each $\pi_k \in \mathcal{P}$ from $k = 1$ to $k = K$, the retry limit $m_{2,k}$ that provides the drop probability closest to the reciprocal of the exponentially weighted distortion, simultaneously verifying that the reception time remains lower than the expiration time. One of the main advantages of this formulation relies on the possibility to obtain a closed-form expression for the estimated retry limits, thus considerably limiting the computational burden. To achieve this result, (21) and (22) may be separately solved. More precisely, one may first use (20) in (21), and then solve the corresponding equation in the integer retry limit recalling that a conservative overestimation of the distortion has been adopted, thus obtaining the quantity:

$$m^D_{2,k} = \left\lceil \frac{\log\left(10^{\zeta D_k}\bar p_2\right)}{\log\left(1/\bar p_2\right)} \right\rceil, \tag{23}$$

which accounts for the sole distortion. Subsequently, since the retry limits are evaluated following the order $k = 1, \ldots, K$, (22) can be usefully rewritten as:

$$T(m_{2,k}) \le T_{e_k} - T_{a_{k-1}}, \tag{24}$$

where, using (16) and (17), the delay accumulated by the $k-1$ packets previous to the kth one can be expressed as:

$$T_{a_{k-1}} = \sum_{k'=1}^{k-1} T(m_{2,k'}) = (k-1)\hat T - \left(\hat T + \frac{E_s W_2}{2}\right)\sum_{k'=1}^{k-1} \bar p_2^{\,m_{2,k'}+1}, \tag{25}$$

with:

$$\hat T = \frac{E_s}{2}\left(\frac{2W_2 - 1}{1-\bar p_2} - W_2\right). \tag{26}$$

The novel formulation in (24) for the time requirement in (22) allows the exploitation of the knowledge of the retry limits already evaluated for $k' < k$. Now, using (16) and (17) in (24), one can solve the corresponding inequality in the integer retry limit that guarantees the satisfaction of the requirement on the expiration time, thus identifying the limiting value:

$$m^T_{2,k} = \left\lfloor \log\left[\frac{\left(\hat T - T_{e_k} + T_{a_{k-1}}\right)^+}{\bar p_2\left(\hat T + E_s W_2/2\right)}\right] \cdot \frac{1}{\log \bar p_2} \right\rfloor, \tag{27}$$

where $(\,\cdot\,)^+$ is the positive part and $\lfloor\cdot\rfloor$ is the floor function. In particular, (27) selects the largest integer that allows the maintenance of the reception time of the kth packet below its expiration time. The positive part is introduced for mathematical purposes to include in a unique expression also the cases in which the term $\hat T - T_{e_k} + T_{a_{k-1}}$ is negative, and hence no requirement on the delay is present in practice. The absence of a delay requirement characterizes, for example, the packets corresponding to the frames indexed from 1 to $\bar I$, which, as explained in Section 3.2, are associated to an infinite expiration time, since the play of the video starts after the elaboration of the frame $f_{\bar I}$. Finally, $m_{2,k}$ can be evaluated by taking the minimum between the value in (23), accounting for the video distortion, and that in (27), accounting for the expiration time, hence obtaining:

$$m_{2,k} = \min\left(m^D_{2,k}, m^T_{2,k}\right). \tag{28}$$

4.3. Summary and remarks

The presented mathematical derivation allows the development of a very fast retry limit adaptation algorithm, which requires just a limited number of operations. These operations are summarized in Fig. 3. Firstly, one evaluates (in order): $\bar p_1$ by solving (14), $\bar\tau_1$ by (12) and (13) for $q = 1$, $\bar p_2$ by solving (15), $\bar\tau_2$ by (12) and (13) for $q = 2$, E_s by (18), and finally $\hat T$ by (26). Observe that all these quantities do not depend on the video packet, thus they can be calculated just once for the entire stream, and, if desired, they might be inserted in a lookup table to avoid their re-calculation when the source has to manage different videos in the same network scenario. Secondly, for each $\pi_k \in \mathcal{P}$ and using the estimations D_k and T_{e_k}, one evaluates (in order): $m^D_{2,k}$ by (23), $T_{a_{k-1}}$ by (25), $m^T_{2,k}$ by (27), and finally $m_{2,k}$ by (28). In general, the algorithm requires just the solution of two polynomial equations and the evaluation of expressions available in closed-form.

It may be useful to observe that the entire framework proposed in this paper, consisting of the estimation process for the video distortion and the subsequent retry limit adaptation, may be viewed as a modular procedure, in the sense that the retransmission strategy could be also exploited using different measures of the distortion, if desired. In fact, one may notice that (23) requires the D_k value, but is not constrained on how this normalized value is obtained. In this paper, D_k has been derived using the EDA to maintain an overall low computational cost. However, the procedure for solving the problem in (21) and (22), and hence the steps of the proposed algorithm, remain identical if a different estimation of the video distortion is adopted. This implies that the proposed retry limit adaptation may be applied not only to distortions obtained using different estimation techniques, but also in the presence of video encoders different from the H.264/SVC, provided that a normalized measure of the distortion is available in some way. Furthermore, just the delay estimation and the retry limit adaptation result necessary if a provider of video services supplies an estimation of the distortion together with the video sequence. The applications enabled by this second possibility, which are not limited to the sole MAC layer, are discussed in [26,30], and may also involve the routing and the transport layers.

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

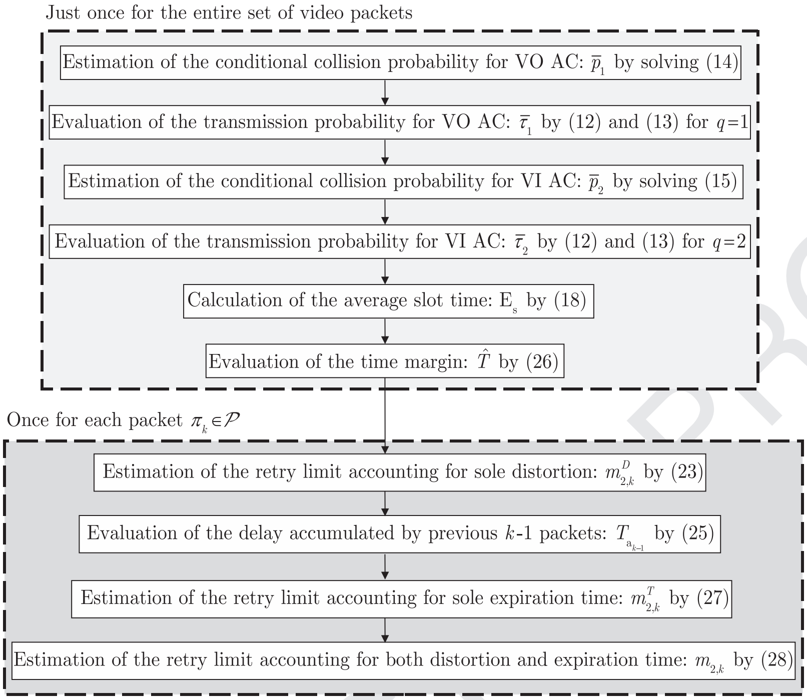

Just once for the entire set of video packets
Estimation of the conditional collision probability for VO AC: $\bar{p}_1$ by solving (14)
Evaluation of the transmission probability for VO AC: $\bar{\tau}_1$ by (12) and (13) for q=1
Estimation of the conditional collision probability for VI AC: $\bar{p}_2$ by solving (15)
Evaluation of the transmission probability for VI AC: $\bar{\tau}_2$ by (12) and (13) for q=2
Calculation of the average slot time: E_s by (18)
Evaluation of the time margin: $\hat{T}$ by (26)
Once for each packet $\pi_k \in \mathcal{P}$
Estimation of the retry limit accounting for sole distortion: $m^D_{2,k}$ by (23)
Evaluation of the delay accumulated by previous k-1 packets: $T_{a_{k-1}}$ by (25)
Estimation of the retry limit accounting for sole expiration time: $m^T_{2,k}$ by (27)
Estimation of the retry limit accounting for both distortion and expiration time: $m_{2,k}$ by (28)

Fig. 3. Retry limit adaptation algorithm.

Similar arguments may hold in the presence of traffic with time constraints much more stringent than those assumed to derive (1) and then (3). In fact, for such kind of traffic, (27) still holds and provides a value that follows the possible stringent requirement specified by T_{e_k}. As shown in [39,40], the VI AC may be selected not only to deliver video sequences, but, more in general, to manage the access of other kinds of traffic, including multimedia contents. Thus, the proposed adaptive algorithm may be applied to any traffic that a network operator decides to associate with the VI AC, provided that a measure of the normalized distortion D_k and of the time constraint T_{e_k} are available for each packet.

5. Results and discussion

This section evaluates the performance obtained from the proposed method. The results are derived using the parameters in Table 1 by assuming that the video playback begins after the elaboration at the receiver of at least the first GOP, thus $\bar{l} = 1 + \alpha = 17$ (the first frame is the frame I). Each AIFS value can be derived as $\text{AIFS}_q = \text{SIFS} + \text{AIFSN}_q\varsigma$, where AIFSN_q denotes the AIFS number for the qth AC. In the following of this section, when one of the parameters in Table 1 will be set to a different value, for example to specifically study its impact on the performance of the developed algorithm, it will be explicitly declared. The transmission buffer of the VI AC of each source is assumed to be sufficiently large to contain all the packets belonging to a given video sequence, in order to avoid losses due to queue overflow, whose modeling is out of the scope of this paper. All routines and the developed 802.11e network simulator are implemented in Matlab. The presented numerical values are obtained using one core of an Intel Core2 Quad Q9300 @2.50 GHz Sun Ultra 24 workstation.

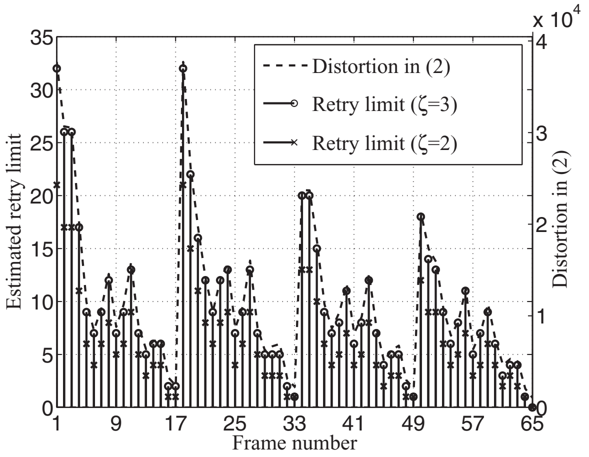


Fig. 4. Retry limits estimated for each frame according to the distortion provided by the EDA for $N = 4$ sources when the VO and VI ACs are both active.

5.1. Estimated retry limits

To provide a clarification of the behavior of the proposed adaptation algorithm, Fig. 4 reports, for each frame of the adopted video sequence (Bus), the distortion in (2) evaluated by the EDA and the corresponding retry limits for two values of the parameter ζ in the presence of $N = 4$ sources when both the VO and VI ACs are active. In this case no requirements on the expiration time are imposed in order to

Table 1
Adopted parameters.

H.264/SVC (*Bus* sequence)		Adaptation algorithm (Basic access; 802.11g)			
Number of frames	$L = 65$	Distortion parameter in (21)	$\zeta = 3$	Slot time	$\varsigma = 20\,\mu$s
				SIFS	SIFS $= 10\,\mu$s
GOP size	$\alpha = 16$	EDA parameter	$\xi = 1/6$	AIFS numbers	AIFSN$_{1,2} = 2$, AIFSN$_3 = 3$, AIFSN$_4 = 7$
Expiration time index	$\bar{l} = 17$	Polynomial degrees in (14)–(15)	$2\bar{N} = 2(N-1)$		
Inter-frame interval	$T_\mathrm{f} = 1/15$ s	Minimum contention windows	$4W_1 = 2W_2 = W_{3,4} = 16$	MAC/PHY header length	H $= 24$ bytes
Number of layers	$V = 4$	Maximum backoff stages	$m'_{1,2} = 1,\ m'_{3,4} = 6$	ACK length	ACK $= 14$ bytes
				Data rate	$R = 54$ Mbits/s
Payload sizes	$\Lambda_{1,2,3,4} = 1400$ bytes	Default retry limits	$m_{1,2,3,4} = 7$	Control rate	$R_\mathrm{c} = 2$ Mbits/s

better outline how the distortion is managed. Consider first the case $\zeta = 3$ (circle marker). For this case the figure shows that the estimated retry limit accurately follows the curve representing the distortion (dashed line), thus revealing that the approximations adopted in (4.2) to develop the algorithm by reducing the complexity of the calculations have a very limited impact on the capability of the algorithm to reliably account for the distortion. A more detailed view of this figure, involving even the case $\zeta = 2$ (cross marker), additionally reveals that also the set of retry limits obtained using $\zeta = 2$ is shaped according to the distortion, but at a different scale. This difference between the two sets of retry limits is useful to understand the impact of the parameter ζ, which in practice can be used to control the overall number of retry limits associated to the sequence in absence of requirements on the expiration time. When these requirements on the delay are instead present, they may reduce the retry limits estimated according to the sole distortion, as it can be inferred from (28).

A final aspect that may be observed from Fig. 4 concerns the high distortion values, and the corresponding high retry limits, that may be noticed for some specific indexes. These indexes identify the I frame, which contains the fundamental encoder settings, and the P frames, which represent the most important frames for the respective GOPs. In fact, all B frames of a given GOP depend on the P frame of that GOP. Thus, while the loss of a B frame has an impact on just a subset of the other B frames of the same GOP, the loss of a P frame has an impact on all the B frames of the same GOP, and hence the loss of a P frame usually results highly detrimental. This damage is reliably managed by the proposed retransmission strategy, since, for a P frame, the estimated distortion and the corresponding retry limit are both high. Of course, possible stringent requirements on the delay may reduce the retry limit also for a P frame.

5.2. Network simulations

Now that the basic behavior of the proposed algorithm has been introduced, the subsequent results aim to further test its performance in a distributed network. Each test is carried out by running 20 network simulations for each considered network scenario, namely for each combination of active ACs and number of contending sources N. Besides, the single network simulation is run for 10 s in order to complete the access procedure for all video packets of all sequences. Each simulation has been carried out at packet-level, that is, the retry limits, once obtained, are used in a packet-level 802.11e/g simulator, which is implemented in Matlab as a state machine. Then, for each simulation, the trace corresponding to the correctly received packets is used to derive the drop probability and the reception time, and is physically elaborated by the H.264/SVC decoder to derive the PSNR by comparing the transmitted video $\mathcal{V}_\mathrm{t}$ with the received one $\mathcal{V}_\mathrm{r}$.

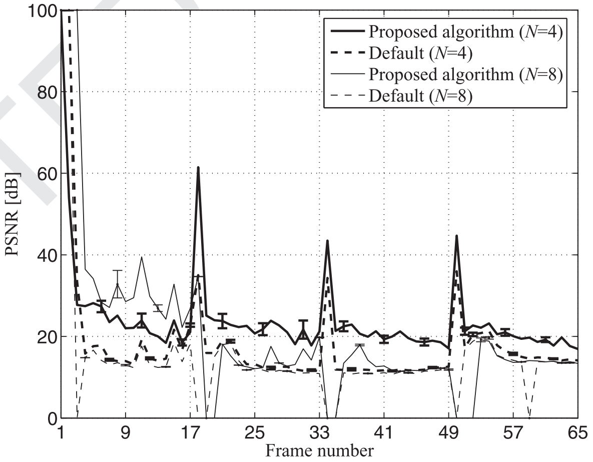


Fig. 5. PSNR for $N = 4$ and $N = 8$ when the VO and VI ACs are active using the proposed algorithm and the default settings.

5.2.1. Preliminary results: two ACs

Figs. 5 and 6, which are obtained for $N = 4$ and $N = 8$ when the VO and VI ACs are active, compare for each frame of the video sequence, the PSNR (Fig. 5) and the playback reception time (Fig. 6) obtained using the proposed algorithm with those derived using the 802.11e/g default settings. The vertical bars present on each curve represent the 95% confidence intervals, which are reported at steps of five frames to maintain the readability of the figures. The playback reception time is evaluated starting from the frame $\bar{l} + 1 = 18$, for

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

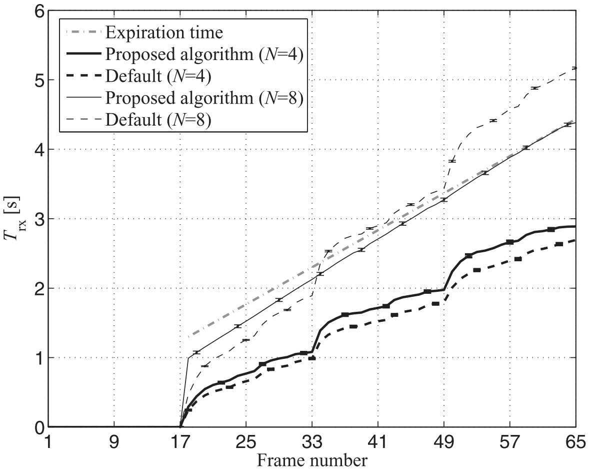


Fig. 6. Playback reception time for $N = 4$ and $N = 8$ when the VO and VI ACs are active using the proposed algorithm and the default settings.

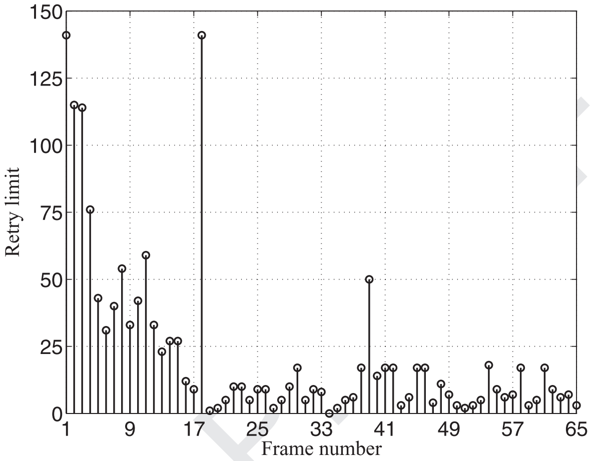


Fig. 7. Estimated retry limits for $N = 8$ sources when the VO and VI ACs are both active.

which the requirement on the expiration time becomes finite. Therefore, for a frame f_l with $l > \bar{l}$, T_{rx} is given by the difference between the reception time of the frame f_l and the reception time of the frame $f_{\bar{l}}$, both obtained from the packet-level simulation. This representation, which adopts the instant of starting of the playback as reference, is in agreement with the requirement expressed by the expiration time, and is suitable to verify if, once the playback of the video is started (after the reception of the frame number $\bar{l} = 17$), the reproduction would be interrupted or not. Each curve is derived by averaging the corresponding quantity (PSNR or playback reception time) over all the simulations and the sources.

The figure shows that the presented algorithm is preferable to the default settings for both values of N. In particular, the default settings achieve acceptable results for $N = 4$, while, when the number of contending sources increases to $N = 8$, the playback reception time largely exceeds the expiration time. The closeness between the curve corresponding to the expiration time and the curve corresponding to the playback reception time for the case $N = 8$ when the proposed algorithm is used represents an interesting confirmation that the time available for video transmission is efficiently exploited by the presented retransmission strategy. Recalling that the retry limit estimation process has been developed by operating on average quantities and that the curves are obtained by averaging the results over the simulations and the nodes, one may expect, as a proof of the correctness of the analysis, that, when the scenario becomes highly congested, the playback reception time approaches the expiration time for the last frames of the video sequence. From the provided confidence intervals, one may also observe that, in a single simulation, the playback reception time may be even slightly higher or slightly lower than this average value, exactly because the developed analysis is based on a predictive approach for estimating the evolution of the transmissions.

One may notice that, when $N = 4$, high PSNR values are achieved for the P frames. This may be again explained recalling the H.264 encoding structure. As previously discussed, all B frames of a GOP depend on the P frame of that GOP. On the other hand, a P frame does not depend on these B frames, and hence it is immune to their loss. This leads to a situation in which, once a P frame is received, the PSNR for that P frame may be very high. Differently, the achievement of high PSNR values for a B frame requires not only the correct reception of that B frame, but also the correct reception of all the B and P frames from which that B frame depends. Thus, the presence of a high PSNR for a B frame is an event less frequent than the presence of a high PSNR for a P frame, which, being more important (i.e., associated to a higher distortion), is protected by a high retry limit.

Concerning the scenario corresponding to $N = 8$, it is worth remarking, firstly, that the contention does not only involve 8 VI ACs, but also 8 higher priority VO ACs, and, secondly, that the minimum contention windows and the maximum backoff stages established by the 802.11e/g standard for the VO and VI ACs are very low (Table 1). This leads to a very constrained scenario, in which the proposed algorithm remains able to guarantee an acceptable video playing within the expiration time in the presence of a necessarily high number of collisions. In these kinds of scenarios, the PSNR referred to some frames may drop to very low values. To this purpose, it is worth to remark that the objective of the proposed algorithm is not to ensure that all frames are received, but to ensure that, in the presence of distortion/delay requirements and contention-based mechanisms, the highest possible quality level (in those network conditions) is achieved for the overall video sequence. To provide more details on the behavior of the proposed adaptation mechanism, Fig. 7 presents the estimated retry limits for the scenario with $N = 8$ sources. Considering this figure together with Fig. 4, one may observe that the retry limits of the frames having index lower than $\bar{l} + 1 = 18$ remain shaped according to the distortion, since the expiration time is infinite for $l < 18$, but they become much higher than those corresponding to the case $N = 4$, because for $N = 8$ the collision probability considerably increases and hence more retransmissions are statistically necessary. For $l \geq 18$, instead, the retry limits are no more exactly shaped according to the

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

distortion, since the requirement on the expiration time becomes dominant.

5.2.2. *General results: two and four ACs*

While the previous results have shown that the proposed algorithm is able to operate in the presence of many contending flows, a second set of simulations is carried out to deepen some further aspects, which have been fundamental during the development of the method.

These aspects concern the performance of the proposed algorithm when all the four ACs are active. The aim is to investigate if the approximation in (10) of the system in (9) is acceptable, simultaneously evaluating the computational time required to estimate the retry limit. The results of this second set of simulations are presented in Table 2, which reports the drop probability and the PSNR (both averaged over the simulations, the sources, and the frames), the maximum playback reception time $T_{\mathrm{rx}_{\max}}$, corresponding to the playback reception time of the last frame of the video sequence (averaged over the simulations and the sources), the single-node throughput $\bar{S}$ of the VI AC (averaged over the simulations and the sources), and the central processing unit (CPU) time required by Matlab to estimate all the retry limits of the video sequence. From now on, the symbol Q is used to denote the number of active ACs. In particular, when $Q = 2$, the sole VO and VI ACs are active, while, when $Q = 4$, all the four ACs are active. For a given source and a given simulation, the throughput is evaluated as the ratio between the sum of the bits correctly received by each destination and that may be used for decoding purposes, and the time instant corresponding to the end of the processing of the last packet of the video sequence. This time is the reception time, when the last packet is correctly received, while it is the discarding time, when the last packet is dropped. To have a reliable term of comparison, Table 2 includes the performance corresponding to the set of optimum retry limits, which are derived by numerically solving the problem in (21) and (22) using the complete system in (9). In particular, the eight equations in (9) and the requirement in (21) have been implemented in a unique Matlab function. Then, the resulting system has been solved using the Matlab function `fsolve`, reducing, if necessary, the derived retry limit until the constraint in (22) is satisfied.

The results in the table confirm that both the optimum retry limit setting and the proposed algorithm provide a better performance with respect to the default settings. As expected, when the number of sources increases, the drop probability increases. However, in these cases, both the optimum setting and the algorithm are able to provide satisfactory PSNR values. It is furthermore interesting to observe that, for each scenario, the drop probability, the PSNR, the maximum playback reception time, and the throughput obtained from the optimum setting and the developed method are very close to each other. In particular, this closeness holds also when all the four ACs are active, thus confirming the reliability of the approximations adopted during the development of the algorithm. This element becomes more relevant when considered together with the CPU time, since the results in Table 2 reveal that less than 2 s are sufficient to evaluate all the retry limits using the proposed algorithm, while some minutes are required by the policy relying on the optimum setting. Concerning this latter aspect, a further noticeable advantage of the presented solution may be signaled. As explained in Section 4.3, the set of operations performed by the algorithm may be subdivided in two subsets: the subset of the operations required to estimate the network evolution, such as the evaluation of the collision probabilities and of the average slot duration E_s, which can be performed just once for the entire stream, and the subset of the operations required to subsequently estimate the retry limit, which must be performed packet by packet (Fig. 3). The CPU times reported in the last column of Table 2 are mainly due to the first subset of operations, while the second subset requires considerably lower computational times.

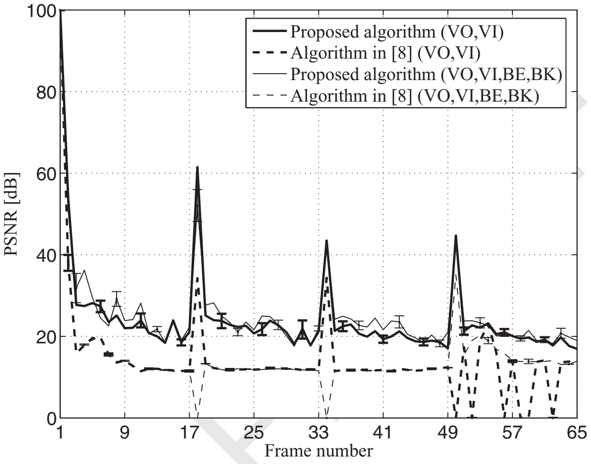


Fig. 8. PSNR for $N = 4$ using the proposed algorithm and the method presented in [8].

5.2.3. *Comparison*

To further test the proposed solution, the obtained results are compared to those achievable using the retry limit adaptation algorithm presented in [8]. This algorithm relies on an unequal loss protection approach developed according to the collision probability and to a strategy of differentiation of the packets into groups, in which the most important packets are associated to higher retry limits. The solution in [8] has been selected for its very low computational cost, thus making significant the comparison with the algorithm proposed in this paper. Since [8] assumes that the collision probability is available and does not consider the problems relative to its estimation, the quantity $\bar{p}_2$, estimated by the proposed algorithm, is used as the collision probability in [8], in order to guarantee a fair comparison between the two algorithms. Furthermore, since [8] does not account for possible time constraints, the results concerning the delay are not considered.

Fig. 8 shows the PSNR as a function of the frame number obtained for $N = 4$ when two (VO and VI) and four ACs are active, while Table 3 reports the average drop probability, the PSNR, the single-node throughput for the VI AC, and the CPU time also for the network scenarios corresponding to larger values of N. The results confirm the satisfactory performance of the proposed algorithm, which remains capable of sustaining the video traffic even in a highly congested

Please cite this article as: F. Babich et al., Fast Retry Limit Adaptation for Video Distortion/Delay Controlretry limit adaptation for video distortion/delay control in IEEE 802.11e Distributed Networksdistributed networks, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.008

Table 2
Drop probability, PSNR, maximum playback reception time, single-node throughput of the VI AC, and processing time for different retry limit setting policies: default (Def), optimum (Opt), and proposed algorithm (Alg).

N	Q	$\bar{p}_{\text{drop}}$ (%)			$\overline{\text{PSNR}}$ (dB)			$T_{\text{rx}_{\max}}$ (s)			$\bar{S}$ (Mbits/s)			CPU time (s)		
		Def	Opt	Alg	Def	Opt	Alg	Def	Opt	Alg	Def	Opt	Alg	Def	Opt	Alg
4	2	40.8	30.9	29.7	18.5	24.2	24.4	2.7	2.9	2.9	0.5	0.8	0.8	–	65.1	1.1
	4	39.0	29.6	30.3	18.6	24.2	25.4	2.7	2.9	2.9	0.5	0.8	0.8	–	110.9	1.2
6	2	72.0	41.9	42.4	18.6	22.6	22.5	4.1	4.6	4.4	0.2	0.3	0.3	–	136.4	1.4
	4	71.8	42.5	43.0	17.1	22.0	21.9	4.1	4.3	4.4	0.2	0.3	0.3	–	228.9	1.4
8	2	88.8	65.9	65.2	16.5	21.7	22.7	5.2	4.3	4.4	0.1	0.2	0.2	–	226.7	1.3
	4	88.7	65.5	65.9	16.4	22.9	21.4	5.2	4.2	4.4	0.1	0.2	0.2	–	393.9	1.4
10	2	94.3	74.9	73.9	16.4	27.0	27.8	5.9	4.2	4.3	0.0	0.1	0.1	–	406.7	1.4
	4	94.1	73.9	74.5	16.7	29.6	25.9	5.9	4.1	4.3	0.0	0.1	0.1	–	792.6	1.5

Table 3
Drop probability, PSNR, single-node throughput of the VI AC, and processing time obtained using the proposed algorithm (Alg) and that presented in [8] (Alg [8]).

N	Q	$\bar{p}_{\text{drop}}$ (%)		$\overline{\text{PSNR}}$ (dB)		$\bar{S}$ (Mbits/s)		CPU time (s)	
		Alg	Alg [8]	Alg	Alg [8]	Alg	Alg [8]	Alg	Alg [8]
4	2	29.7	76.0	24.4	15.8	0.8	0.2	1.1	1.1
	4	30.3	76.7	25.4	15.4	0.8	0.2	1.2	1.2
6	2	42.4	99.5	22.5	21.5	0.3	0.0	1.4	1.4
	4	43.0	99.5	21.9	18.1	0.3	0.0	1.4	1.4
8	2	65.2	99.7	22.7	–	0.2	0.0	1.3	1.3
	4	65.9	99.7	21.4	–	0.2	0.0	1.4	1.4

Table 4
Drop probability, PSNR, maximum playback reception time, single-node throughput of the VI AC, and processing time when $N = 4$ sources transmit different video sequences (source 1: *Bus*, source 2: *Container*, source 3: *Foreman*, source 4: *News*).

Source	Optimum retry limit setting				Proposed algorithm				Algorithm in [8]			
	1	2	3	4	1	2	3	4	1	2	3	4
$\bar{p}_{\text{drop}}$ (%)	21.2	11.2	27.2	10.4	19.5	12.4	26.4	9.2	64.2	30.3	34.7	32.0
$\overline{\text{PSNR}}$ (dB)	40.4	86.9	33.2	72.8	36.9	81.9	33.7	76.8	17.6	32.8	24.6	42.6
$T_{\text{rx}_{\max}}$ (s)	2.1	0.4	0.7	0.5	2.0	0.4	0.7	0.4	1.8	0.4	0.6	0.4
$\bar{S}$ (Mbits/s)	1.1	0.9	0.9	0.9	1.1	0.9	0.9	0.9	0.4	0.7	0.8	0.8
CPU time (s)	114.5	33.2	37.1	28.9	1.3	0.2	0.3	0.3	1.3	0.2	0.3	0.3

environment. In particular, one may observe from the last two columns of Table 3, that the CPU times for the two solutions appear as identical. In practice, differences are present just in the not reported less significant decimals. The similarity is due to the use of the same procedure of estimation for the collision probability, which, as previously discussed, has the larger impact on the computational cost. Thus, once this estimation is available, the remaining calculations have a negligible cost for both compared algorithms. Combining this characteristic with the satisfactory performance achievable by the proposed method, one may conclude that the here presented algorithm is able to provide a really satisfactory tradeoff between performance and complexity.

5.2.4. Different video sequences

As a final set of results, Table 4 reports, for each source, the performance obtained when $N = 4$ sources transmit different video sequences. The table considers the optimum retry limit setting, the proposed algorithm, and the solution presented in [8]. The values are derived allowing that, at each source, all the four ACs are active. Observe that, in some cases, similar throughput values may appear in conjunction with different drop probabilities. In particular, a direct comparison between the proposed algorithm and that presented in [8] for the fourth video sequence (*News*) shows that the two throughput values are close, but the drop probabilities are considerably different.

The reason of this behavior may be explained, firstly, remembering that the drop probability is referred to the frames, while the throughput is referred to the packets usable for decodable frames, and, secondly, recalling the characteristics of the two retransmission strategies. In this specific case, the two algorithms allow the reception of a similar number of packets, but the packets received adopting the proposed algorithm enable the decoding of a number of frames that is larger than that enabled by the packets received adopting the algorithm in [8]. This is confirmed by the different PSNR values, and further outlines the importance of adopting a sophisticated estimation of the distortion, and, in turn, of the retry limits.

One may notice from the table that also in this case the developed algorithm provides values for the drop probability, the PSNR, the maximum playback reception time, and the throughput that are very close to those achievable using the optimum setting, simultaneously maintaining a very low CPU time, which remains identical to that required by the method conceived in [8]. The similarity of the performance provided by the optimum setting and the proposed algorithm in this heterogeneous (in terms of videos) scenario confirms the limited impact of the simplifying hypothesis adopted in Section 4.2, according to which each source selects its retry limit assuming that the other sources select the same one. This latter result, combined with the other validations reported in this section, confirms that the adopted modeling approach represents a suitable solution for reaching the purposes considered at the beginning of this study: the development of a retry limit adaptation strategy that, moving from the initial objective of guaranteeing a low computation cost, be able to satisfy distortion and delay requirements for video transmission over 802.11e distributed networks.

6. Conclusions

A fast and simple retry limit adaptation method for video streaming applications over 802.11e distributed networks in the presence of distortion and delay requirements has been presented. The method has been derived by carefully modeling the evolution of the network and by introducing proper approximations, whose reliability has been validated by numerical simulations, which have allowed to considerably reduce the computational cost of the conceived solution.

The results have shown that the presented algorithm is able to accurately account for the impact of the higher priority VO AC on the video transmission, while simultaneously maintaining the frame delay below the video expiration time. The satisfactory performance has been reached maintaining a really low processing time for the retry limit estimation process. This latter advantage reveals that the developed algorithm may be also of interest for possible implementations on network devices characterized by very limited computational resources.

Acknowledgments

This work is supported in part by the National Inter-University Consortium for Telecommunications (CNIT) within the project "Multiple access improvements in high-capacity 802.11 networks for video streaming support", and by the Italian Ministry of University and Research (MIUR) within the project FRA 2013 (University of Trieste, Italy), entitled "Multi-packet communication in 802.11x heterogeneous mobile networks: models and antenna system algorithms."

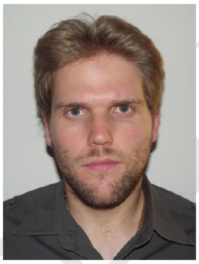
References

[1] R. Zhang, L. Cai, J. Pan, X. Shen, Resource management for video streaming in ad hoc networks, Elsevier Ad Hoc Netw. 9 (4) (2011) 623–634.
[2] Q. Ni, L. Romdhani, T. Turletti, A survey of QoS enhancements for IEEE 802.11 wireless LAN, Wiley Wireless Commun. Mobile Comput. 4 (5) (2004) 547–566.
[3] S. Kumar, V.S. Raghavan, J. Deng, Medium access control protocols for ad hoc wireless networks: a survey, Elsevier Ad Hoc Netw. 4 (3) (2006) 326–358.
[4] IEEE Std 802.11e, IEEE Standard for Wireless LAN Medium Access Control (MAC) and PHYsical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, November 2005.
[5] Q. Li, M. van der Schaar, Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation, IEEE Trans. Multimedia 6 (2) (2004) 278–290.
[6] M. van der Schaar, D.S. Turaga, R. Wong, Classification-based system for cross-layer optimized wireless video transmission, IEEE Trans. Multimedia 8 (5) (2006) 1082–1095.
[7] M.H. Lu, P. Steenkiste, T. Chen, A time-based adaptive retry strategy for video streaming in 802.11 WLANs, Wiley Wireless Commun. Mobile Comput. 7 (2) (2007) 187–203.
[8] Y. Zhang, Z. Ni, C.H. Foh, J. Cai, Retry limit based ULP for scalable video transmission over IEEE 802.11e WLANs, IEEE Commun. Lett. 11 (6) (2007) 498–500.
[9] J.-L. Hsu, M. van der Schaar, Cross layer design and analysis of multiuser wireless video streaming over 802.11e EDCA, IEEE Signal Process. Lett. 16 (4) (2009) 268–271.
[10] C.-M. Chen, C.-W. Lin, Y.-C. Chen, Cross-layer packet retry limit adaptation for video transport over wireless LANs, IEEE Trans. Circuits Syst. Video Technol. 20 (11) (2010) 1448–1461.
[11] H. Bobarshad, M. van der Schaar, M.R. Shikh-Bahaei, A low-complexity analytical modeling for cross-layer adaptive error protection in video over WLAN, IEEE Trans. Multimedia 12 (5) (2010) 427–438.
[12] H. Bobarshad, M. van der Schaar, A.H. Aghvami, R.S. Dilmaghani, M.R. Shikh-Bahaei, Analytical modeling for delay-sensitive video over WLAN, IEEE Trans. Multimedia 14 (2) (2012) 401–414.
[13] C. Greco, M. Cagnazzo, B. Pesquet-Popescu, Low-latency video streaming with congestion control in mobile ad-hoc networks, IEEE Trans. Multimedia 14 (4) (2012) 1337–1350.
[14] C.F. Kuo, N.W. Tseng, A.C. Pang, A fragment-based retransmission scheme with quality-of-service considerations for wireless networks, Wiley Wireless Commun. Mobile Comput. 13 (16) (2013) 1450–1463.
[15] J. Jimenez, R. Estepa, F.R. Rubio, F. Gomez-Estern, Energy efficiency and quality of service optimization for constant bit rate real-time applications in 802.11 networks, Wiley Wireless Commun. Mobile Comput. 14 (6) (2014) 583–595.
[16] R. Corrado, M. Comisso, F. Babich, On the impact of the video quality assessment in 802.11e ad-hoc networks using adaptive retransmissions, in: IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2014, pp. 47–54.
[17] P. Ameigeiras, J.J. Ramos-Munoz, J. Navarro-Ortiz, J.M. Lopez-Soler, Analysis and modelling of YouTube traffic, Wiley Trans. Emerging Telecommun. Technol. 23 (4) (2012) 360–377.
[18] ITU-T, Recommendation H.264: Advanced Video Coding for Generic Audiovisual Services, Annex G: Scalable Video Coding, January 2012.
[19] T. Stutz, A. Uhl, A Survey of H.264 AVC/SVC Encryption, IEEE Trans. Circuits Syst. Video Technol. 22 (3) (2012) 325–339.
[20] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, S. Kotsopoulos, Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks, Elsevier Ad Hoc Netw. 9 (4) (2011) 591–607.
[21] X. Zhu, B. Girod, A unified framework for distributed video rate allocation over wireless networks, Elsevier Ad Hoc Netw. 9 (4) (2011) 608–622.
[22] M. Schier, M. Welzl, Optimizing selective ARQ for H.264 live streaming: a novel method for predicting loss-impact in real time, IEEE Trans. Multimedia 14 (2) (2012) 415–430.
[23] S.-H. Chang, R.-I. Chang, J.-M. Ho, Y.-J. Oyang, A priority selected cache algorithm for video relay in streaming applications, IEEE Trans. Broadcasting 53 (1) (2007) 79–91.
[24] R. Zhang, S.L. Regunathan, K. Rose, Video coding with optimal inter/intra-mode switching for packet loss resilience, IEEE J. Select. Areas Commun. 18 (6) (2000) 966–976.
[25] Y. Wang, Z. Wu, J.M. Boyce, Modeling of transmission-loss-induced distortion in decoded video, IEEE Trans. Circuits Syst. Video Technol. 16 (6) (2006) 716–732.
[26] F. Babich, M. D'Orlando, F. Vatta, Video quality estimation in wireless IP networks: algorithms and applications, ACM Trans. Multimedia Comput. Commun. Appl. 4(1). Q2
[27] M. Baldi, J.C. De Martin, E. Masala, A. Vesco, Quality-oriented video transmission with pipeline forwarding, IEEE Trans. Broadcasting 54 (3) (2008) 542–556.
[28] Z. Li, J. Chakareski, X. Niu, Y. Zhang, W. Gu, Modeling and analysis of distortion caused by Markov-model burst packet losses in video transmission, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 917–931.
[29] F. Babich, M. Comisso, M. D'Orlando, F. Vatta, Distortion estimation algorithms (DEAs) for wireless video streaming, in: IEEE Global Telecommunications Conference (GLOBECOM), 2006, pp. 1–5.
[30] S. Adibi, R. Jain, S. Parekh, M. Tofighbakhsh, Quality of Service Architectures for Wireless Networks: Performance Metrics and Management, IGI Global, New York, 2010.
[31] G. Bianchi, Performance analysis of the IEEEE 802.11 distributed coordination function, IEEE J. Select. Areas Commun. 18 (3) (2000) 535–547.

[32] P. Chatzimisios, A.C. Boucouvalas, V. Vitsas, Influence of Channel BER on IEEE 802.11 DCF, IET Electron. Lett. 39 (23) (2003) 1687–1688.
[33] G.R. Cantieni, Q. Ni, C. Barakat, T. Turletti, Performance analysis under finite load and improvements for multirate 802.11, Elsevier Comput. Commun. 28 (10) (2005) 1095–1109.
[34] D. Malone, K. Duffy, D. Leith, Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions, IEEE/ACM Trans. Netw. 15 (1) (2007) 159–172.
[35] B. Alawieh, C. Assi, H. Mouftah, Power-aware ad hoc networks with directional antennas: models and analysis, Elsevier Ad Hoc Netw. 7 (3) (2009) 486–499.
[36] F. Babich, M. Comisso, M. D'Orlando, A. Dorni, Deployment of a reliable 802.11e experimental setup for throughput measurements, Wiley Wireless Commun. Mobile Comput. 12 (10) (2012) 910–923.
[37] K. Kosek-Szott, A comprehensive analysis of IEEE 802.11 DCF heterogeneous traffic sources, Elsevier Ad Hoc Netw. 16 (2014) 165–181.
[38] J.W. Tantra, C.H. Foh, A.B. Mnaouer, Throughput and delay analysis of the IEEE 802.11e EDCA saturation, in: IEEE International Conference on Communications (ICC), vol. 5, 2005, pp. 3450–3454.
[39] N. Cranley, M. Davis, Video frame differentiation for streamed multimedia over heavily loaded IEEE 802.11e WLAN Using TXOP, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2007, pp. 1–5.
[40] A. Politis, I. Mavridis, A. Manitsaris, C. Hilas, X-EDCA: a cross-layer MAC-centric mechanism for efficient multimedia transmission in congested IEEE 802.11e infrastructure networks, in: IEEE International Conference on Wireless Communications and Mobile Computing (IWCMC), 2011, pp. 1724–1730.

F. Babich received the doctoral degree in electrical engineering, from the University of Trieste, in 1984. From 1984 to 1987 he was with the Research and Development Department of Telettra, working on optical communications. Then he was with Zeltron (Electrolux group), where he held the position of Company Head in the Home System European projects. In 1992 he joined the Department of Electrical Engineering (DEEI), now converged in the Department of Engineering and Architecture, of the University of Trieste, where he is associate professor of digital communications and telecommunication networks. Fulvio Babich has served as co-chair for the Communication Theory Symposium, ICC 2005, Seoul, for the Wireless Communication Symposium, ICC 2011, Kyoto, for the Wireless Communication Symposium, WCSP 2012, Huangshan, China, and for the Communication Theory Symposium, ICC 2014 Sidney. He is senior member of IEEE. Fulvio Babich has been member of the Directive Board of CNIT (National Inter-University Consortium for Telecommunications, a non-profit Consortium among 37 Italian Universities, whose main purpose is to coordinate and foster basic and applied research).

M. Comisso received the "Laurea" degree in Electronic Engineering and the Ph.D. degree in Information Engineering from the University of Trieste, Italy. He worked for Alcatel in the field of optical communication systems and collaborated with Danieli Automation in the field of electromagnetic sensors' modeling. Currently Massimiliano Comisso is an Assistant Professor at the Department of Engineering and Architecture of the University of Trieste. He is author/co-author of more than 40 international scientific papers, and serves as reviewer/TPC member for several IEEE journals and conferences. He has been Best Student Paper Award Finalist at Globecom 2006 and has received the Best Paper Award at CAMAD 2009. His research interests involve smart antenna systems, distributed wireless networks, antenna array synthesis, and small antennas. Massimiliano Comisso is Member of IEEE.



R. Corrado received the Laurea degree in Telecommunications Engineering from the University of Trieste (Italy) in 2011, where, since 2012, he is Ph.D. student in Information Engineering within the Department of Engineering and Architecture. In 2013, he granted a scholarship from CNIT (National Inter-University Consortium for Telecommunications) within the field of low-complexity access techniques for streaming applications. His research interests involve video encoding and wireless ad hoc networking.