

RESEARCH

Open Access



Virtual queue dropping for robust real-time video over IEEE 802.11aa wireless LANs

Wen-Ping Lai* and En-Cheng Liou

Abstract

Stream classification service (SCS) is a novel concept proposed in the newly released IEEE 802.11aa standard for robust audio and video streaming, particularly for graceful degradation of streaming quality during network congestions. Based on intra-access category prioritisation (IACP), SCS is equipped with a pair of access category (AC) queues to differentiate *real-time* and *non-real-time* video streams, namely (AC_VI, AAC_VI), and thus introduces the need for an efficient scheduler design; similarly for audio. In this study, we propose a novel cross-layer design called virtual queue dropping (VQD) for packet scheduling between AC_VI and AAC_VI. Driven by a real-time video-packet-importance scheme, VQD adopts *conditional priority weighting* to enhance the priority of I-packets from AAC_VI so that they can fairly compete with P-packets from AC_VI. Moreover, to achieve graceful degradation during network congestions, VQD (i.e. based on *priority coupling* among different *component* queue lengths within the same *physical* AC queue) is adopted so that the dropping probabilities of lower-priority video packets are not only *larger* but also *affected-and-increased* by those of higher-priority ones. Different limits of *priority retries* of I- and P-packets can avoid unexpected long delays, and an optimal set of retry limits has been found to minimise the performance impact due to real-time constraints. Our results show that VQD outperforms PWD and SCS-CR in various performance metrics, and achieves a *win-win* game for the dilemma in maintaining *priority* for AC_VI and *fairness* for AAC_VI simultaneously.

Keywords: 802.11aa, Intra access category prioritisation, Stream classification service, Graceful degradation, Conditional priority weighting, Virtual queue dropping, Priority retries

1 Introduction

Video over wireless networks has been the driving force for the advancements of wireless technologies, and the family of IEEE 802.11 protocol standards [1] has successfully dominated the wireless LAN market. However, high-quality and robust wireless video transmission is still a challenging issue because the wireless media is error-prone in signal and time-varying in bandwidth.

In terms of wireless media contention, the legacy 802.11 [2] was originally based on distributed coordination function (DCF) [3] to support collision handling among wireless stations. In DCF, each wireless station adopts a single first-in-first-out (FIFO) queue for all types of traffic flows, as shown in Fig. 1a, and thus cannot support quality of service (QoS) for video flows. To address this issue, the 802.11e standard [4] proposed

a QoS scheme called enhanced distributed channel access (EDCA), where a QoS-capable station (QSTA) is equipped with multiple access category (AC) *queues* and *virtual collision handling* among these AC queues so as to achieve *in-station service differentiation* of video flows from other traffic types. Among the four equipped AC queues, as shown in Fig. 1b, AC_VO and AC_VI queues are reserved for *voice* and *video* flows, respectively, with higher transmission priorities than the other AC queues (i.e. AC_BE for *best effort* flows and AC_BK for *background* flows). However, the AC_VI queue is still FIFO-based intrinsically, and thus no further service differentiation among *video* flows or packets can be achieved. Similarly, no service differentiation among *voice* flows or packets either.

In the literature, there exist some studies providing service differentiation among video packets, where video packets of less importance are mapped to the lower-priority AC queues than AC_VI either *statically* [5] or

* Correspondence: wpl@saturn.yzu.edu.tw
Department of Communications Engineering, Yuan Ze University, 135 Yuan-Tung Rd, Chung-Li, Taoyuan City, Taiwan

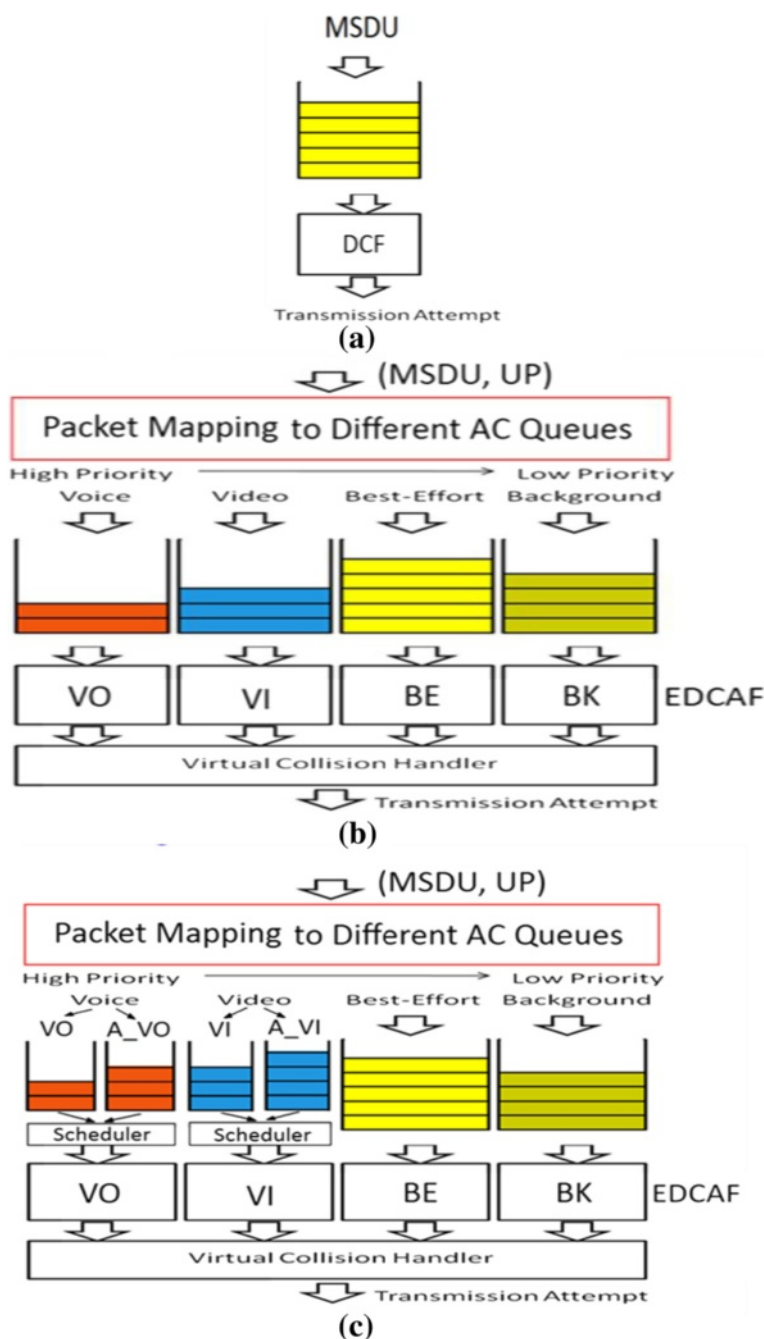


Fig. 1 Evolution of QoS strategies for wireless channel contention in IEEE Std. 802.11: **a** DCF in 802.11-1999, **b** EDCA in 802.11e-2005, and **c** IACP and SCS in 802.11aa-2012

dynamically [6–10]. Although these studies have achieved performance gains over EDCA, the traffic impact of those downward mapped video packets to the lower-priority AC queues has not been well-discussed. To address this, in the newly released 802.11aa-2012 standard [11], two new concepts called intra access category prioritisation (IACP) and stream classification service (SCS) have thus been proposed. IACP achieves

flow-level differentiation between *real-time* and *non-real-time* video by a double-queue framework, namely the *primary* (VI) and *alternative* (A_VI) AC queues as shown in Fig. 1c, also referred to as AC_VI and AAC_VI, respectively, throughout this study. Based on IACP, SCS further achieves *packet-level* differentiation to pursue *graceful degradation* of video streaming quality during congestions by introducing drop eligibility indicator (DEI) enabling

differential retransmission limits for lost video packets of different importance levels. Similarly, service differentiation can also be achieved for voice at both the *flow* and *packet* levels. In IACP and SCS, voice or video traffic flows are confined to their own (primary, alternative) AC queues, and thus generate zero traffic impact to the lower-priority AC queues.

Both IACP and SCS introduce the need of efficient video packet scheduling between the (primary, alternative) AC queues before passing a *head-of-line* video packet to the EDCA Function of video AC (VI-EDCAF), which adopts the same set of wireless channel contention parameters as EDCA. In 802.11aa-2012, despite incapability of *packet-level* differentiation, *credit-based scheduling* is recommended by default. To address this, this study proposes a novel cross-layer scheduler design called virtual queue dropping (VQD) aiming at substantial performance gains.

The remainder of this article is organised as follows. Section 2 briefly identifies the scope of 802.11aa-2012 and overviews the related studies of the SCS-related scheduler design. Section 3 describes the design principles and details of the proposed VQD scheduler, followed by various performance evaluations and extensive analyses in Section 4. Finally, Section 5 concludes this article.

2 Scope of 802.11aa and related studies of SCS packet scheduling

The authors of [12] have delivered an overview on the scope of 802.11aa-2012, as well as the *QoS management frame service issue* of 802.11ae-2012 [13], both of which serve as two amendments to the IEEE 802.11-2012 revised version. In addition to IACP and SCS, 802.11aa-2012 also covers several other advanced issues, including groupcast with retries (GCR) for reliable and scalable transmission of multicast streams [14–16], overlapping basic service sets (OBSS) management for handling the coordination problem among multiple access points (AP) caused by today's large deployment of 802.11 LANs, and *interworking with* the IEEE 802.1Q stream reservation protocol (SRP) [17] for supporting end-to-end SRP when one or more 802.11 links are involved in the transmission path. However, these advanced issues are out of the scope of this study. A good overview of these issues can be found in [12, 18, 19].

Open research areas related to these issues in both the two aforementioned amendments have also been pointed out in [12]; in particular, for IACP, performance analysis on the mapping of streams or individual MAC frames to either the *primary* and *alternative* AC queues are needed, and the selection of parameters for the *credit-based scheduler* is desired. As for SCS, it is important to decide which packets of *lower* coding

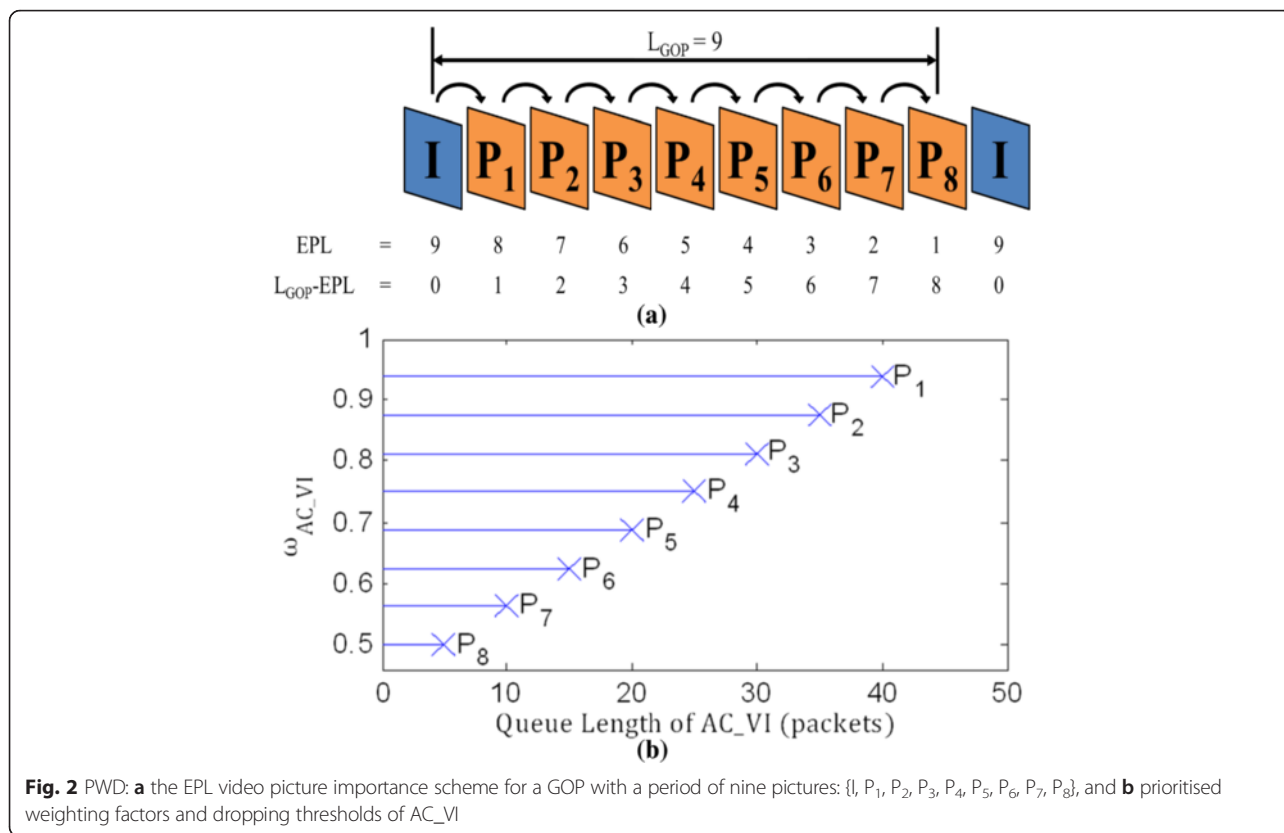
importance should be *less* retried if the wireless channel capacity becomes insufficient so as to achieve graceful degradation of multimedia transmission quality. Note that the term 'frame' is an official name for 'packet' at the MAC layer after encapsulation with the MAC header. However, to avoid any possible confusion with the term 'video frame' which means 'video picture', the terms 'packet' and 'video picture' will be used instead throughout the rest of this article.

Since the objective of this study is to design an efficient video packet scheduler for both real-time and non-real-time video streams in the scenario of SCS, the following subsections mainly focus on two related studies of SCS schedulers, and the main contributions of our proposed VQD design.

2.1 Priority weighting and dropping-based scheduling

To the best of our knowledge, our previous study [20] contributed the first article to address the performance analyses of IACP and SCS, which is one of the aforementioned open issues in [12]. In that study, a cross-layer design called priority weighting and dropping (PWD) was proposed and compared with weighted round robin (WRR) based IACP and SCS (denoted as IACP-WRR and SCS-WRR, respectively), where different sets of weighting parameters were chosen and analysed for IACP-WRR and SCS-WRR. In general, the *primary* AC queue, deserving *higher* transmission priority, needs a *larger* weighting parameter than the *alternative* AC queue, and it is difficult to achieve *priority* and *fairness* simultaneously between the (primary, alternative) queues of the same AC, assuming that *real-time* and *non-real-time* video streams are differentiated by mapping them to the *primary* and *alternative* AC queues respectively. However, our proposed PWD design can achieve a *win-win game* over IACP-WRR and SCS-WRR in its performance gains for both *real-time* and *non-real-time* streams. Here *real-time* streams are referred to as those for *interactive* (two-way) communication scenarios with *tight* end-to-end delay bounds ranging from 250 ms (human-eyes perceptible) to 400 ms (human-eyes unbearable) according to the ITU regulation, while *non-real-time* streams for *one-way* reception scenarios with only *loose* end-to-end delay bounds defined by the *jitter buffer* (also known as play-out buffer) on the video receiver side.

The performance gains are achieved in the sense that PWD performs a finer weighting algorithm using the importance levelling of video packets according to the error propagation length (EPL) index [21], where linearly descending importance levelling is assumed for those video pictures in the same group of pictures (GOP) and thus packets belonging to the same video picture share the same EPL value. For instance, for a GOP with a period of nine video pictures, Fig. 2a demonstrates their



corresponding EPL values: {9, 8, 7, 6, 5, 4, 3, 2, 1} for {I, P₁, P₂, P₃, P₄, P₅, P₆, P₇, P₈}, respectively. On the other hand, as shown in Fig. 2b, the weighting values of AC_VI (ω_{AC_VI}) for *real-time* video packets are *equally spaced* among the range of [0.5, 1], where the leading I-picture adopts the maximum value, and the trailing P₈-picture takes the minimum value. Meanwhile, $\omega_{AAC_VI} = 1 - \omega_{AC_VI}$ so that the sum of these two weighting values is normalised to 1. In addition, for graceful degradation, PWD also adopts a finer packet-dropping strategy based on the EPL value in the sense that P₈-packets are dropped with the *lowest* (earliest) dropping threshold, P₁-packets with the *highest* (latest) one, and I-packets are fully reserved until full queuing since I-packets are extremely important for subsequent decoding of P-pictures in the same GOP. Note that a missing part of [20] lies in no consideration or implementation for the credit-based scheduler design for SCS (SCS-CR), recommended by default in 802.11aa-2012.

2.2 Credit-based scheduling

Credit-based algorithms have been widely used in many areas, such as *credit-based flow control* in the PCI-Express bus system of personal computers, *credit-based fair queuing* for optimised transport of data traffic over optical fibre ring networks (also known as the IEEE 802.17 resilient packet ring networks [22]), and *credit-*

based traffic shaping in the IEEE 802.1Qav component of Ethernet audio/video bridging for future car bus and infotainment [23]. In this study, to demonstrate the superiority of VQD over PWD and SCS-CR, we have also implemented a simplified version of credit-based scheduling algorithm for the 802.11aa double-queue framework of the video AC, rooted from Annex L of both IEEE 802.1Q and IEEE 802.1Qav and based on the following design principles.

- Whenever one of the (primary, alternative) queues for the video AC is empty, the weighting factor of the non-empty one becomes 100 % in order for its *head-of-line* video packet to be passed to VI-EDCAF; otherwise, consider the following.
- Either queue is allowed to transmit only if its current credit value is positive. During the transmission period of VI-EDCAF, the credit value of the *sending* queue is decreased at a negative slope called *sendSlope*, while that of the *idle* queue is increased at a positive slope called *idleSlope*. Note that, by definition, AC_VI has a larger weighting factor, a steeper *idleSlope* and a smoother *sendSlope* than AAC_VI in order to get a larger chance of transmission. Note that both the credit values of the double queues remain intact when their VI-EDCAF

is *idle* so that both the double queues become *idle* simultaneously.

- As a result, the queue with a smaller *head-of-line* packet size (S), a larger accumulated credit value (C) and a larger weighting factor (ω) will form a smaller metric $(S - C)/\omega$ and thus have a better chance to win the transmission. Note that the credit of the *idle* queue is accumulated by $\omega_{idle} (S_{send} - C_{send})/\omega_{send}$ during its *idle* period.

For the implemented SCS-CR scheme (for performance comparison with this study), the adopted value set of $(\omega_{AC_VI}, \omega_{AAC_VI})$ is taken to be (0.9, 0.1) as a typical case which prefers much higher priority of AC_VI than AAC_VI, and thus one expects much better performance for *real-time* video streams while reserving some basic level of fairness to *non-real-time* video streams. (1, 0) is one extreme case that behaves pretty like a variant of *strict priority*, with only a minimum level of fairness. (0.5, 0.5) is the other extreme that behaves almost like round-robin, i.e. the perfect case for fairness. However, no matter which case is taken between the two extremes, SCS-CR still cannot tackle the dilemma problem between *priority* and *fairness* eventually.

2.3 Major contributions of VQD

Compared to PWD [20] (our previous study), VQD (this study) has the following major contributions:

- *Conditional priority weighting* for non-real-time I-packets from AAC_VI
The details of this contribution are given in Section 3.1. The novelty of this contribution lies in the *fairness* (and thus quality) enhancement of non-real-time video with a *minimum tradeoff* in real-time video quality degradation via *boosting the relative importance of non-real-time* I-packets from AAC_VI to make them equally compete with *real-time* P-packets of *any* importance level, not just with P-packets of *least* importance as in the case of PWD.
- *Virtual queue dropping* of P-packets based on priority-coupled component queue lengths
The details of this contribution can be found in Section 3.2. VQD adopts a packet dropping scheme which is *flexible, probabilistic* and based on *priority-coupled component* queue length in order to achieve *finer* adaptation to congestion level changes and thus *higher* video transmission quality, compared to the packet dropping scheme of PWD, which is *fixed-dropping-thresholds, non-probabilistic*, and based on *non-priority-coupled physical* queue length.
- *Priority retries* of I- and P-packets
The details of this contribution can be found in Section 3.3. The purpose is to find an optimal

combination of priority retry limits for these two packet types, with a larger limit for I-packets due to their higher importance, so as to minimise the performance impact due to real-time constraints. This contribution is totally unique to VQD and not considered in PWD at all.

3 Proposed cross-layer design (VQD)

In this study, the proposed cross-layer design also adopts the EPL index as the importance scheme of video packets to help the design of an efficient scheduler between the (primary, alternative) queues of the video access category before passing them to VI-EDCAF, where the *primary* queue (i.e. AC_VI) buffers *real-time* video packets and the *alternative* queue (i.e. AAC_VI) buffers *non-real-time* ones. The novelty of this scheduler design is three-folded: (1) *conditional priority weighting* for enhancing the scheduling probability of non-real-time I-packets from AAC_VI, (2) *virtual queue dropping* of less important P-packets for adaptation to different congestion levels based on priority-coupled component queue lengths, and (3) *priority retries* of I- and P-packets for minimising the performance impact due to real-time constraints with a larger retry limit for I-packets. The details of these are explained below with their design principles.

3.1 Conditional priority weighting for non-real-time I-packets from AAC_VI

As aforementioned, *real-time* streams are much more *tightly time-bounded*, and thus deserve better transmission resources than *non-real-time* streams. In terms of scheduling between AC_VI and AAC_VI, it is equivalent to requiring $\omega_{AC_VI} > \omega_{AAC_VI}$.

Recall from the PWD design that the concept of EPL-based *priority weighting* has been demonstrated to be successful, and the above inequality was slightly modified to be $\omega_{AC_VI} \geq \omega_{AAC_VI}$, where the equality holds only when *non-real-time* I-packets (from AAC_VI) encounter *real-time* P-packets of *least importance* (from AC_VI), e.g. P₈-packets in the case of Fig. 2a. Also recall that the importance levelling of EPL is *linearly descending* from the leading I-picture to its subsequent P-pictures within a given GOP, and the values of ω_{AC_VI} are proportional to the corresponding EPL values and *equally spaced* among the range [0.5, 1]. Meanwhile, the normalisation condition $\omega_{AAC_VI} = 1 - \omega_{AC_VI}$ constrains the values of ω_{AAC_VI} to be within [0, 0.5]. Hence, it is difficult in general for *non-real-time* I-packets to compete with any type of *real-time* packets, and the best chance is to get an equal weighting factor when a *non-real-time* I-packet encounters a real-time P-packet of least importance at the head-of-line. Obviously, this could be the weakness of PWD because I-packets with long delays can induce much more serious error propagation than any subsequent P-packets

within the same GOP in terms of the decoded video quality at the video receiver when exceeding the playback deadline.

Hence, in the proposed VQD design, the concept of EPL-based priority weighting is extended with a *conditional priority weighting* strategy: *boosting the relative importance of non-real-time* I-packets from AAC_VI to make them equally compete with *real-time* P-packets of *any* importance level, not just with P-packets of *least* importance as in PWD. Table 1 describes how this strategy is formed. Although the weighting scheme is the same as PWD when a *non-real-time* P-packet of arbitrary importance level encounters any *real-time* video packet, the major novelty lies in the fact that *boosting the relative importance of non-real-time* I-packets from AAC_VI can greatly enhance the transmission *fairness* and *quality of non-real-time* video due to the most prominent importance levelling of I-packets within a GOP, with a minimum cost of undermining the real-time video quality via *conditional priority weighting*. Note that such a priority boosting for *non-real-time* I-packets from AAC_VI still becomes invalid once they encounter *real-time* I-packets from AC_VI to ensure the transmission priority of the latter.

3.2 Virtual queue dropping of P-packets based on priority-coupled component queue lengths

In general, dropping P-packets of less importance can help to achieve some level of graceful degradation during network congestions. Recall that PWD adopted a *fixed* scheme for dropping P-packets based on the *physical* queue length of AC_VI or AAC_VI, where different dropping thresholds (DTs) were set for P-packets of different EPL indices, with a lower DT value for a less important packet and a higher DT value for a more important one. In this study, we propose a *flexible* scheme for dropping P-packets of different EPL indices based on the *component* queue length of each individual EPL index and its coupling relationship to those of other EPL indices. We call such a scheme as virtual queue dropping (VQD).

The objective of VQD is to achieve a better graceful degradation by *flexible adaptation* to the variation of congestion level, and the design principle behind VQD is to form *coupling* relationships among the multiple

Table 1 Conditional priority weighting

$(\omega_{AC_VI}, \omega_{AAC_VI})$	Non-real-time packets via AAC_VI	
	I	P ₁ ~ P ₈
Real-time packets via AC_VI	I	(1, 0)
	P ₁ ~ P ₈	(0.5, 0.5)
		(1 - α , α)

$\alpha = L_{GOP} - EPL$, as defined in Fig. 2a. α is the equally spaced distance between any two neighbouring weighting factors, as shown in Fig. 2b

component queue lengths associated with different EPL indices within the same *physical* AC queue, such as AC_VI or AAC_VI. More details are explained below.

To represent the *dropping* probability ($P_{drop,i}$) for an arriving P-packet of importance level i (P _{i} -packet) at either AC_VI or AAC_VI, a hyperbolic tangent function of *physical queue length* (q_{phy}) with the *inflection point* at F_i is chosen, as shown in Eq. (1). Note that there are two undetermined parameters: κ for the functional shape, and F_i for the inflection point. With the selection of κ , it creates a chance for performance optimisation in the *dropping probability* space of $P_{drop,i}$ as shown in Fig. 3a. Basically, $P_{drop,i}$ is characteristic of three phases:

- an *asymptotic* phase of *low* dropping probability when $q_{phy} \ll F_i$
- an *asymptotic* phase of *high* dropping probability when $q_{phy} \gg F_i$
- a *transition* phase where $P_{drop,i}$ rapidly changes from *low* to *high* or from *high* to *low* when q_{phy} is around F_i (note that $P_{drop,i} = 1/2$ when $q_{phy} = F_i$, i.e. at the inflection point).

On the other hand, as shown in Fig. 3b, the selection of F_i is a more subtle task with the consideration of different importance levels for P-packets. The subtlety is explained as follows.

- Let us consider a video stream of a GOP structure with a period of *nine* video pictures (one leading I-picture followed by eight P-pictures) as a typical example. Since every I-picture is referenced for decoding its subsequent P-pictures, none of I-packets should be dropped until the congestion level exceeds the *limit of physical queue* (q_{limit}).
- For P-packets, eight branches of *dropping probability functions* with different values of inflection points are adopted so that P₁-packets have the largest value (F_1) and P₈-packets the smallest value (F_8). As a result, for the same value set of κ and q_{phy} , the dropping probability of P₁-packets is the lowest one ($P_{drop,1}$) while that of P₈-packets is the highest one ($P_{drop,8}$). Namely, the smaller value for i , the higher importance level it stands for.
- Moreover, even for a given value of level i , F_i is still not a fixed value. Eq. (2) shows the definition of F_i , where the value of F_i runs within $(0, q_{limit})$ and is related to two parameters: $q_{v,i}$ and γ . $q_{v,i}$ is based on a concept called *virtual queue length* which is a ratio running between 0 and 1. Its definition can be found in Eq. (3), where the numerator is the sum of the *component queue length* of level i (denoted as q_i) and those of higher importance levels (denoted as $q_{j < i}$), and the denominator q_{phy} is the *total physical queue*

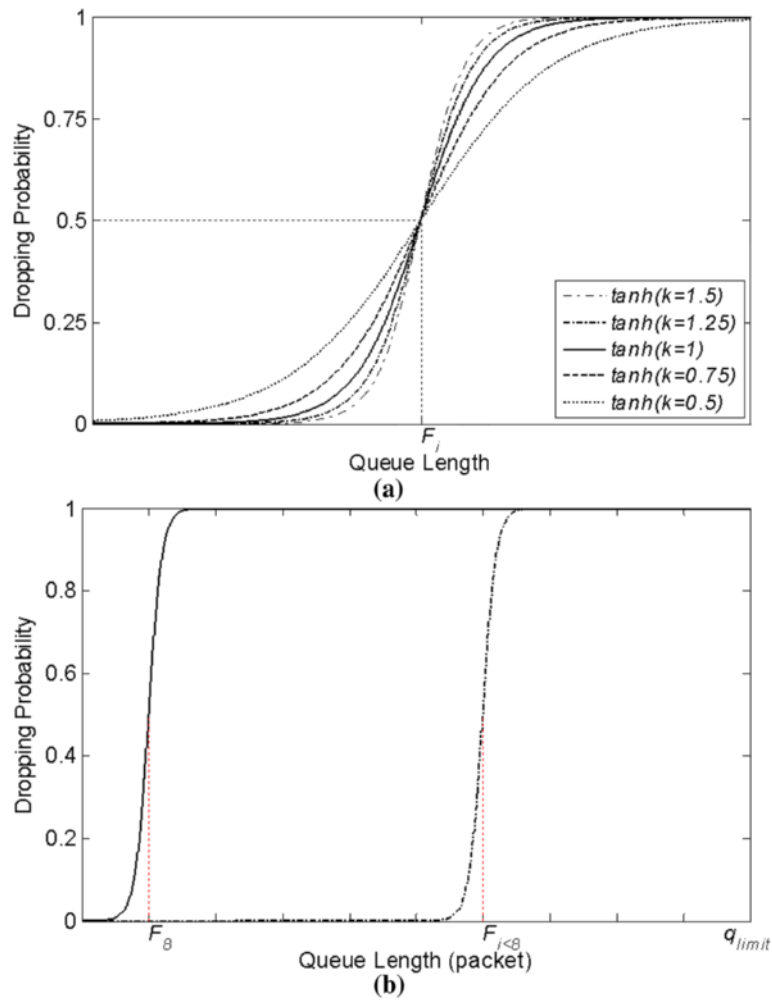


Fig. 3 Dropping probability functions based on \tanh : **a** with different functional shapes determined by κ , and **b** with different inflection points determined by F_i

length, namely the sum of all the *component queue lengths*. The design goal of $q_{v,i}$ is to allow for a *congestion-adaptive adjustment of the dropping probability of level i influenced by those higher importance levels* (i.e. those $j < i$). Note that the value of i is equivalent to $(L_{GOP} - EPL)$, as shown in Fig. 2a. For example, $q_{v,4}$ is the *virtual queue length* of P_4 -packets, and $q_{v,4} = (q_4 + q_3 + q_2 + q_1 + q_0)/q_{phy}$. Hence, it is easy to see the two extreme cases: $q_{v,1} = (q_1 + q_0)/q_{phy}$ and $q_{v,8} = 1$. Note that q_0 is the component queue length of I-packets, q_1 is that of P_1 -packets, and thus q_i is that of P_i -packets accordingly. On the other hand, γ simply serves to weaken the competition capability of P_1 -packets so that they will not jeopardise I-packets when the congestion level becomes extremely heavy. In principle, to confine F_i within $(0, q_{limit})$, the value of γ should also be limited within $(0, 1)$. An optimal value set of (κ, γ) is presented and discussed in Section 4.2 and Fig. 5.

Although we have explained the proposed cross-layer design based on the GOP coding pattern with a period of *nine* video pictures in Fig. 2a, the design can be easily extended to any GOP period.

$$P_{drop,i} = \frac{1}{2} \left(1 + \tanh(\kappa(q_{phy} - F_i)) \right) \quad (1)$$

$$F_i = q_{limit} * (1 - \gamma q_{v,i}) \quad (2)$$

$$q_{v,i} = \frac{q_i + \sum_{j=1}^{i-1} q_j}{q_{phy}} \quad (3)$$

3.3 Priority retries of I- and P-packets

Transmission retries for lost packets due to wireless errors can reduce the packet loss rate on one hand, but they could also generate unexpected packet delays on

the other hand, in particular during congestions. The unexpected packet delays may be harmless to *non-real-time* video applications such as *on-demand* video delivery as long as the play-out buffer on the video receiver side is set to be long enough, but should definitely be avoided for *real-time* applications such as *conversational* video communication. Hence, a *limit of retries* is required to avoid meaningless transmissions and can save the bandwidth resources. However, a further question to ask in this study is what the optimal limit of retries should be. To address this, our design principle is to allow for priority retry limits of I- and P-packets, as explained below:

- The priority retry limits of I- and P-packets should be controlled so that the performance impact due to *real-time constraints* can be as small as possible.
- Furthermore, because I-packets are more important than P-packets within the same GOP, I-packets should deserve a larger limit of retries than P-packets.

A series of studies to find the optimal combination of priority retry limits for these two packet types, under various congestion cases and time constraints, are presented and analysed in Fig. 6 and Section 4.3, and those performance comparisons among different designs are also based on this optimal combination to be fair.

To gain a deeper insight, it could be helpful to see how wireless errors induce packet drops if the limit of retries is exceeded. Although wireless error sources cover receiving power attenuation, signal interference, multipath fading etc., the VQD design adopts the *two-ray ground reflection* model for simplicity, to avoid the complexity of multipath fading so that the main focus can be placed on the multi-access collision issue. Multi-access collisions (including both *physical collisions* among wireless stations and in-station *virtual collisions* among ACs) can trigger transmission retries in the MAC layer, and eventually packet drops if the limit of retries is exceeded in *physical collisions*.

Obviously, the *physical collision probability* enlarges with the *number of wireless stations* (N_s) or the total traffic rate before it reaches a saturated rate. On the other hand, the *virtual collision probability* should also enlarge with the number of AC-traffic flows and their flow rates similarly. Nevertheless, based on our study, *virtual collisions* only increase packet delays, but do not drop packets even if over retried, and thus it is enough to understand or predict the *packet loss rate* (PLR) simply from the *physical collision probability*, without knowing the details of the *virtual collision probability*. In this study, we boldly assume that the *physical collision probability* of 802.11e and 802.11aa should be the same as that of 802.11 if they all adopt the same value of

initial contention window (W) for *random* backoff in the retry process, ignoring the fact that 802.11 might adopt a different value for *fixed* backoff. In other words, they all share the same functional form of W . The results for some tested conditions of PLR, presented in the second half of Section 4.3, verify and support this assumption. Thus, the remaining questions are (1) how to predict the *physical collision probability* and (2) how to predict PLR from the *physical collision probability*.

For question (1), it was shown in [24] that the *mean-value* model of contention window (W_{mean}) in the random backoff process can be used to predict the *physical collision probability* $P_{\text{phy_col}}$ (802.11) in DCF-based multi-access channel contention under ‘saturated’ traffic rates, as shown in Eqs. (4) and (5), and a good agreement in $P_{\text{phy_col}}$ (802.11) was obtained between the prediction and simulation data ($ns-2$), with differences around 1~3 %. Note that Eqs. (4) and (5) are mutually iterative and can be solved with a unique solution P if given the parameter set $\{m, K, N_s\}$. For question (2), based on the aforementioned arguments and big assumption (i.e. they all share the same functional form of W_{mean}), we can thus predict the packet loss rates for 802.11, 802.11e and 802.11aa, all based on Eq. (6).

$$P = 1 - \left(1 - \frac{1}{W_{\text{mean}}}\right)^{N_s - 1} \quad (4)$$

where $P \equiv P_{\text{phy_col}}$ to be concise notation-wise, and

$$\begin{aligned} W_{\text{mean}} = & \eta \left(\frac{W-1}{2}\right) \\ & + \eta P \left(\frac{2W-1}{2}\right) + \dots \\ & + \eta P^m \left(\frac{2^m W-1}{2}\right) \\ & + \eta P^{m+1} \left(\frac{2^{m+1} W-1}{2}\right) + \dots \\ & + \eta P^{K-1} \left(\frac{2^{m+1} W-1}{2}\right) \end{aligned} \quad (5)$$

with $\eta = \frac{1-p}{1-p^K}$, $2^m W = W_{\text{max}}$, $(K-1) =$ limit of retries, $K = \#$ of max transmissions.

$$\text{PLR} = P^K \quad (6)$$

where the derivation of Eq. (6) is given below:

$$\begin{aligned} \text{PLR} &= 1 - \text{PRR} \\ &= 1 - \sum_{i=0}^{K-1} (1-P)^i P^i \\ &= 1 - (1-P) [1 + P + P^2 + \dots + P^{K-1}] \\ &= 1 - (1-P^K) \\ &= P^K \end{aligned}$$

with $\text{PRR} =$ packet receivable rate.

As also discussed in [24], the above *mean-value* model is actually a special case of the two-state Markov chain model when the conditional channel access probability of the current slot immediately after an idle time slot is equal to that immediately after a busy time slot. It is because of this special case that the physical collision probability can be solved uniquely. Markov chain modelling has been useful in analysing communication or networking problems, more applications of Markov chain modelling can also be easily found elsewhere, such as semi-Markov decision process modelling in helping the hand-off design of a train moving between access points so as to improve the train control system performance [25], or in achieving an optimal joint session admission control scheme in integrated WLAN(802.11e)/CDMA networks to utilise overall radio resources [26], both in a fashion of cross-layer design.

4 Results

This section presents our simulation results to demonstrate the superiority of the proposed VQD design over the existing ones in achieving a *win-win game* for both the *real-time* and *non-real-time* video performances through AC_VI and AAC_VI, respectively. Firstly, we

describe the simulation setup, including the networking topology and experimental settings for various traffic flow types, followed by analyses spanning over different performance metrics and influential factors, including the effects of the functional shape of *tanh* due to optimisation of κ and the inflection point of *tanh* due to optimisation of γ , the retry effects of I- and P-packets, the average queue lengths, the packet delay probability distributions and their associated *real* and *effective* losses without and with a *real-time* constraint respectively, and the effect of the adopted reference video sequences on the *peak-signal-to-noise-ratio* (PSNR) variation under various congestion levels.

4.1 Experimental setup

A simulation environment based on ns-2 [27] was adopted to conduct experiments for performance analyses. As shown in Fig. 4, an 802.11aa wireless LAN with access to the Internet was established, where the AP and all the wireless stations were 802.11aa QoS-enabled. The network capacity of the wireless LAN was adopted to be 2 Mbps and assumed to be the bottleneck of any end-to-end traffic, including the traffic from the Internet. Note that the AP plays a role as the traffic hub in general, and

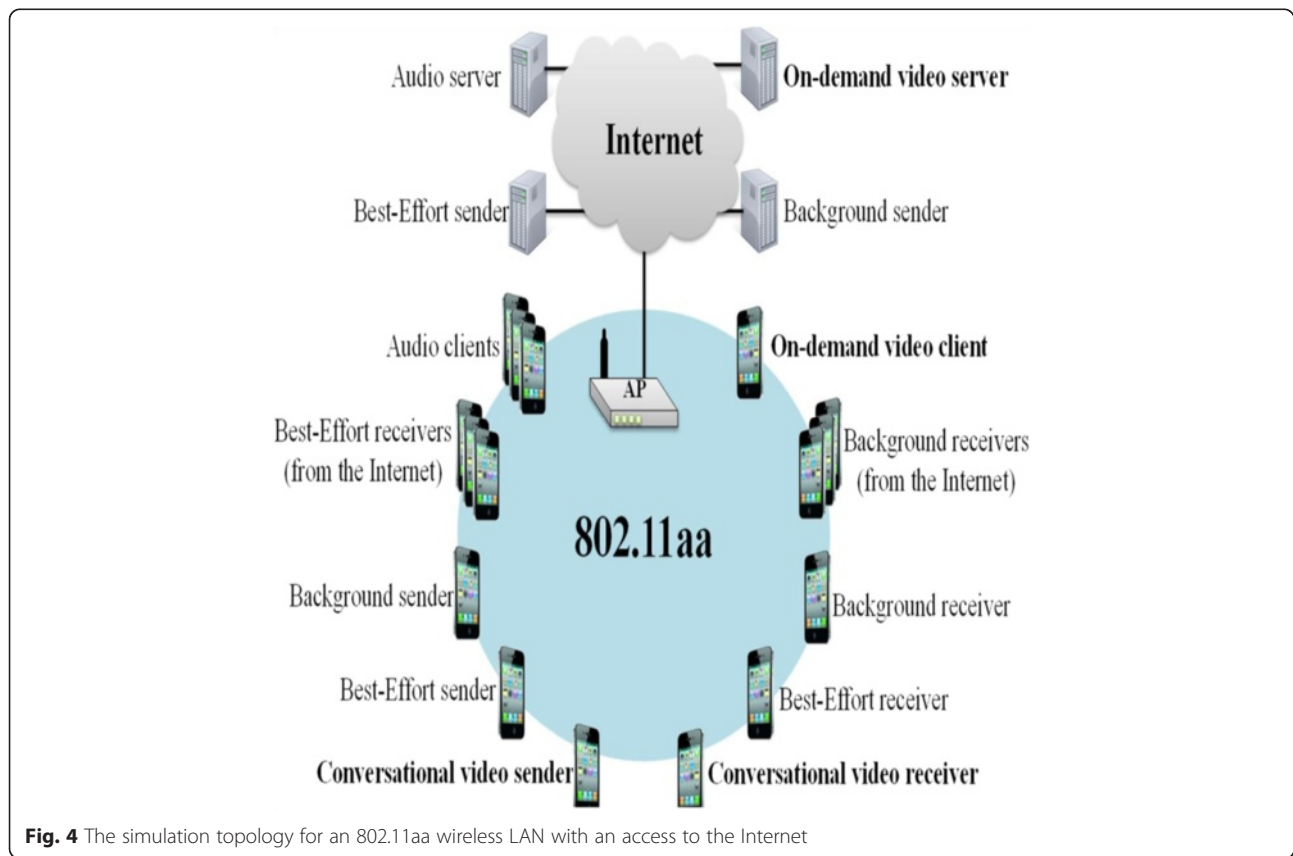


Fig. 4 The simulation topology for an 802.11aa wireless LAN with an access to the Internet

Table 2 Experimental settings for various traffic types, mapped AC queues, traffic sources and input rates

Traffic type	AC queues of AP	Rate from the Internet (bps)	Rate from the wireless LAN (bps)
Real-time audio	AC_VO	Not applied	Not applied
Non-real-time audio	AAC_VO	$n \times 40$ k	Not applied
Real-time video (conversational)	AC_VI	Not applied	512 k
Non-real-time video (on-demand)	AAC_VI	512 k	Not applied
Best effort	AC_BE	$n \times 300$ k	50 k
Background	AC_BK	$n \times 300$ k	50 k

for simplicity it was chosen to fully run in the contention mode and no contention free period was assigned. The experimental settings of different traffic types/sources/input rates and their corresponding AC queues on the AP are summarised in Table 2, where some traffic sources were set to come from the Internet (including the *non-real-time* on-demand video via the AAC_VI queue on the AP) and some from the wireless LAN (including the *real-time* conversational video via the AC_VI queue on the AP). Note that both *best effort* (BE) and *background* (BK) traffic flows were set to be UDP-based. Due to running in the full con-

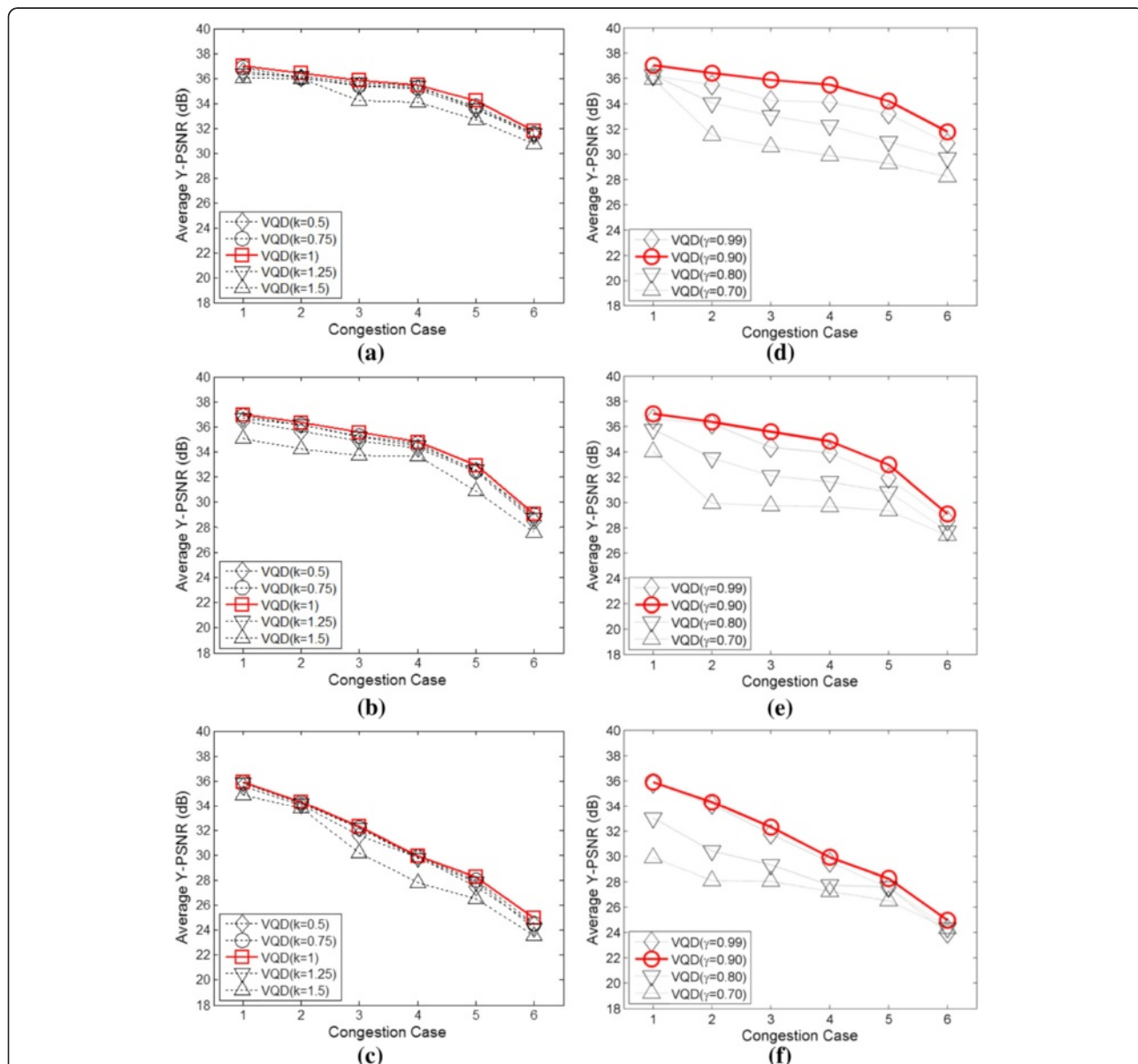


Fig. 5 Effects of κ (i.e. the functional shape of \tanh , based on Eq. (1)) and γ (the inflection points F_i of \tanh , based on Eq. (2)) on average Y-PSNR variations of the proposed VQD design under various congestion cases, with different real-time constraints, for κ : **a** RT-Cut-400 ms, **b** RT-Cut-300 ms, and **c** RT-Cut-200 ms, and for γ : **d** RT-Cut-400 ms, **e** RT-Cut-300 ms, and **f** RT-Cut-200 ms

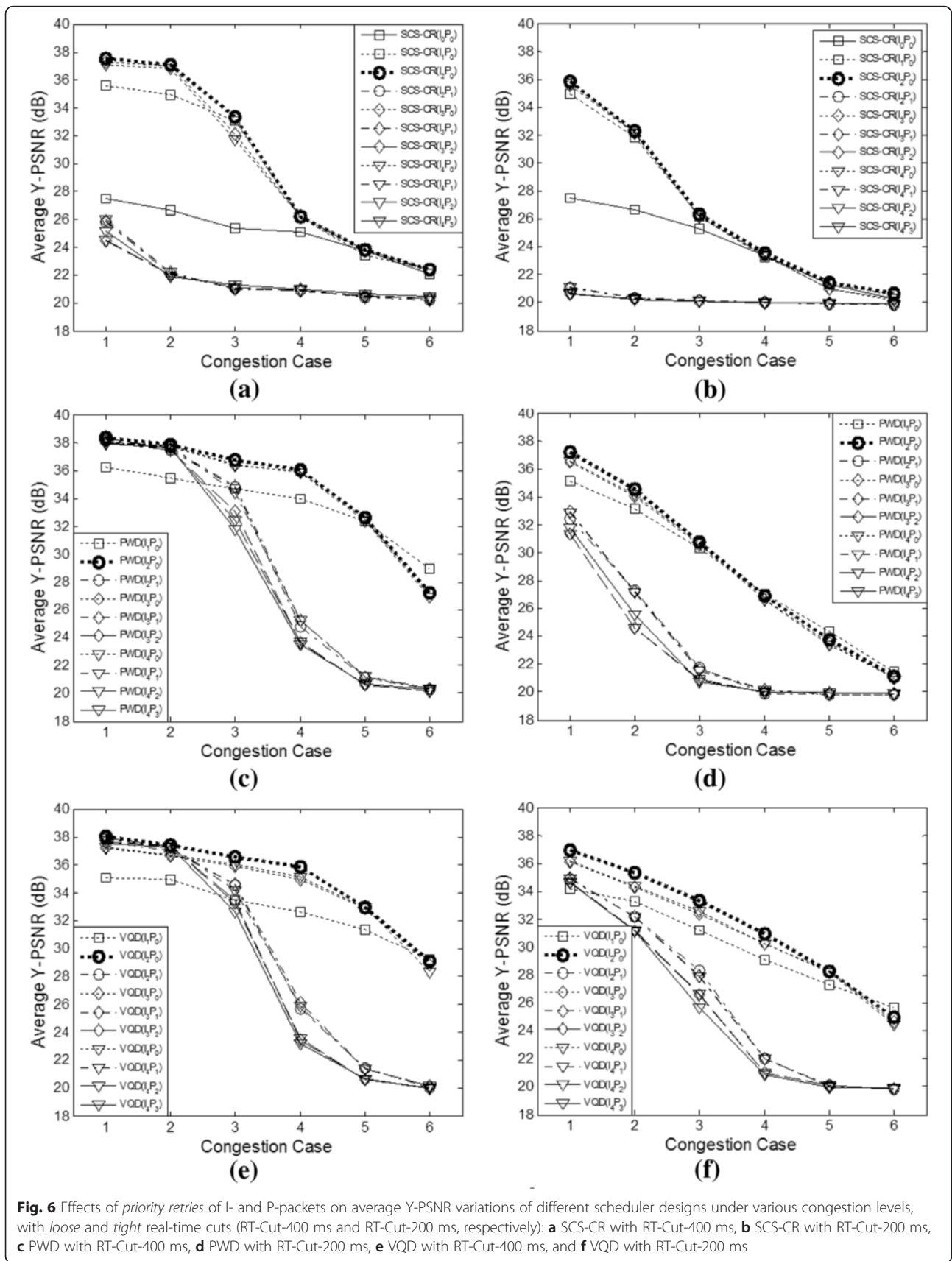


Fig. 6 Effects of *priority retries* of I- and P-packets on average Y-PSNR variations of different scheduler designs under various congestion levels, with *loose* and *tight* real-time cuts (RT-Cut-400 ms and RT-Cut-200 ms, respectively): **a** SCS-CR with RT-Cut-400 ms, **b** SCS-CR with RT-Cut-200 ms, **c** PWD with RT-Cut-400 ms, **d** PWD with RT-Cut-200 ms, **e** VQD with RT-Cut-400 ms, and **f** VQD with RT-Cut-200 ms

tention mode, the traffic sources from the Internet were needed to control/adjust the congestion levels of other non-video AC queues, in terms of the number of cross-traffic flows, denoted by n , running from 1 to 6 to represent *light* to *heavy* congestion levels respectively. In order to present fair performance analyses, the same value set of EDCAFs was adopted for all the target schedulers under comparison, as shown in Table 1 of [8].

Two video streams encoded from the same reference video sequence were sent, respectively, and asynchronously from the *conversational* video sender and *on-demand* video server to their corresponding receiver and client respectively. Note that the analyses of Figs. 5, 6, 8 and 9 were all based on *Container QCIF*. To confirm the performance superiority of VQD is a general effect among the adopted sequences, four reference video sequences [28] with *slow/fast* motions and *QCIF/CIF* YUV formats were adopted, including *Container QCIF (medium-slow)*, *Foreman QCIF (fast)*, *Mother-and-Daughter CIF (slow)* and *Carphone CIF (fast)*.

Each reference video sequence was encoded into a 512-kbps bit-stream in the GOP pattern $\{I, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$ with a period of *nine* video pictures by the H.264 [29] reference software (JM 10.2). The video transmission quality evaluation toolset *EvalVid* [30] was used to packetise the target bit-stream, and the video packets were then transmitted over the adopted network topology to obtain the packet loss information so that the transmission quality can be evaluated according to the information of packet losses. The transmission quality of each received video picture was evaluated based on the *yield* part of PSNR (Y-PSNR), as defined by Eq. (7), where $(Y_{ij} - Y_{ij}^*)$ is the *yield* difference per pixel *before* and *after* quality impairment. The performance metric for the entire video sequence was then based on the *average* of all the picture-based Y-PSNR values. To be statistically meaningful, the performance data point for each congestion case was obtained from averaging over 30 simulation runs, denoted as Average Y-PSNR throughout the rest of this article.

$$Y\text{-PSNR} = 10 \log_{10} \frac{255^2 MN}{\sum_{i=1}^M \sum_{j=1}^N (Y_{ij} - Y_{ij}^*)^2} \quad (7)$$

4.2 Optimal values for κ and γ

To understand how the value of κ affects the shape of the adopted *tanh* function in our proposed VQD design and thus its average Y-PSNR performance, five values of κ (0.5, 0.75, 1, 1.25, 1.5) have been selected to demonstrate the performance variations in different congestion cases. As shown in Fig. 5a–c, three *real-time constraints* or *cuts* (denoted as RT-Cut-400 ms, RT-Cut-300 ms, RT-Cut-200 ms) have been applied,

respectively. As expected, the congestion robustness of VQD should go down with a *tighter* real-time cut. Two major messages can further be delivered below.

- For all the applied real-time constraints, $\kappa = 1$ can generate the optimal performance.
- The performance differences between $\kappa = 1$ and the other values are not so large, around 1~2 dB, and thus it indicates the appropriateness and stability of the adopted *tanh* function for performance robustness of the proposed VQD design.

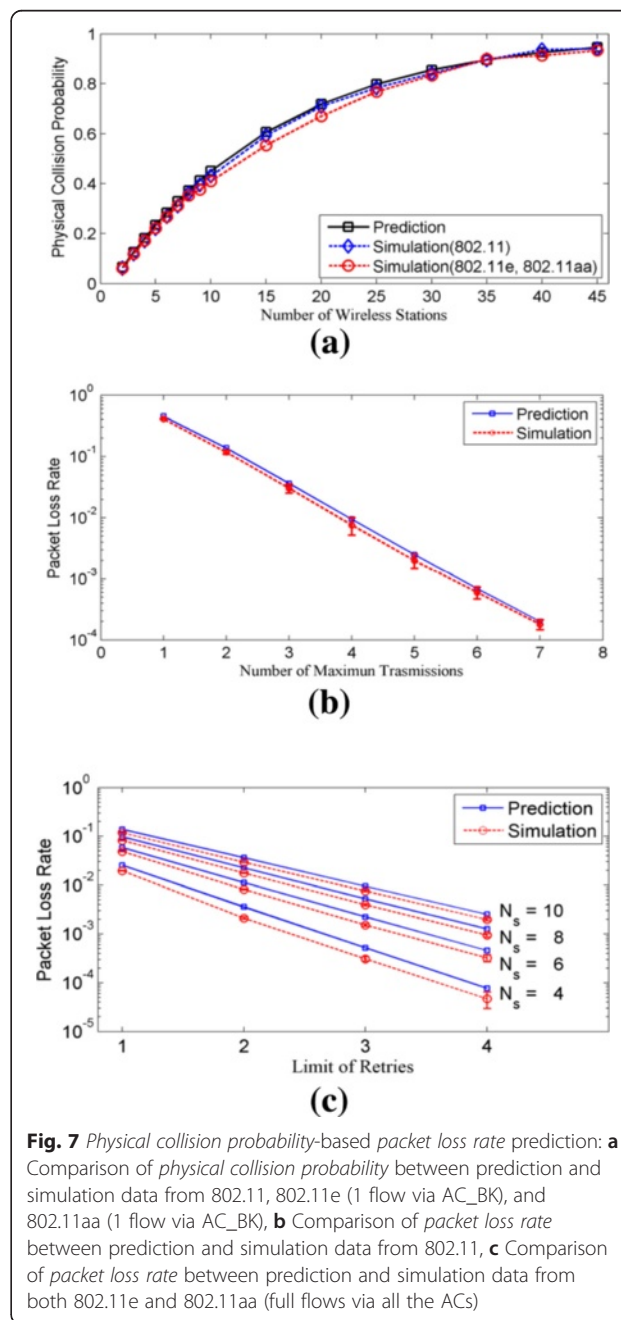
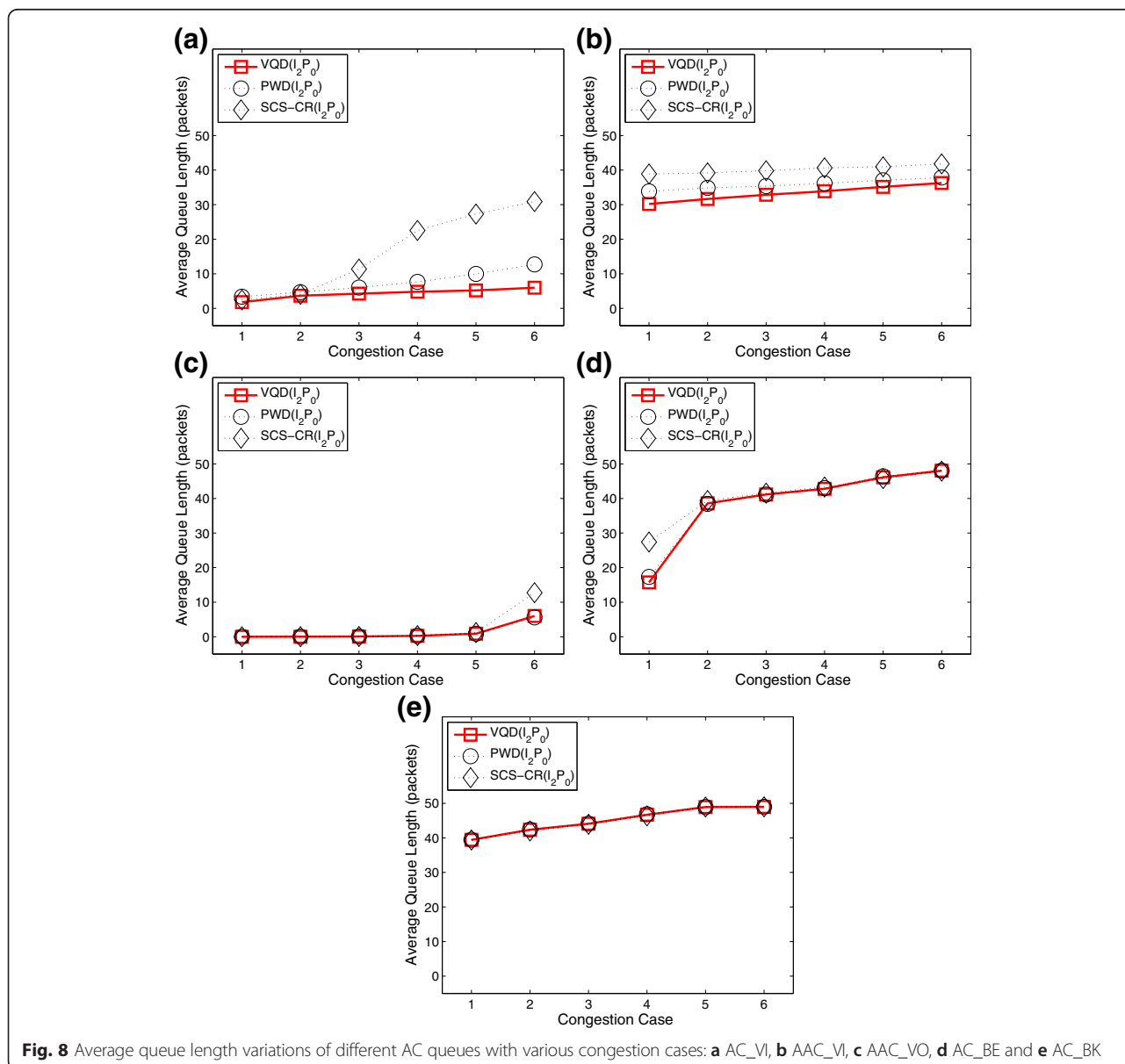


Fig. 7 Physical collision probability-based packet loss rate prediction: **a** Comparison of physical collision probability between prediction and simulation data from 802.11, 802.11e (1 flow via AC_BK), and 802.11aa (1 flow via AC_BK), **b** Comparison of packet loss rate between prediction and simulation data from 802.11, **c** Comparison of packet loss rate between prediction and simulation data from both 802.11e and 802.11aa (full flows via all the ACS)

The above analysis was conducted assuming a fixed value of γ at 0.9. To further understand how the value of γ affects the inflection point of each *tanh* function for priority dropping of P-packets in VQD and thus its average Y-PSNR performance, four values of γ (0.7, 0.8, 0.9, 0.99) have been selected to demonstrate the performance variations in different congestion cases, given the optimal value $\kappa = 1$. Again, the three aforementioned *real-time constraints* have also been applied, respectively, for Fig. 5d–f. As expected, the congestion robustness of VQD also goes down with a *tighter* real-time constraint. Two major messages can further be delivered below.

- For all the applied real-time constraints, $\gamma = 0.9$ can generate the optimal performance.
- The performance gains of $\gamma = 0.9$ over the other values are much larger (particularly in comparison with $\gamma = 0.7$ and 0.8), namely more visible than the performance difference among the κ values. Hence, it implies that it is more sensitive and significant to select an optimal value of γ . Meanwhile, the performance becomes more stable and robust when $\gamma \geq 0.9$.

Based on the above analyses, the optimal value set ($k = 1$, $\gamma = 0.9$) is adopted for further performance analyses throughout the rest of this article.

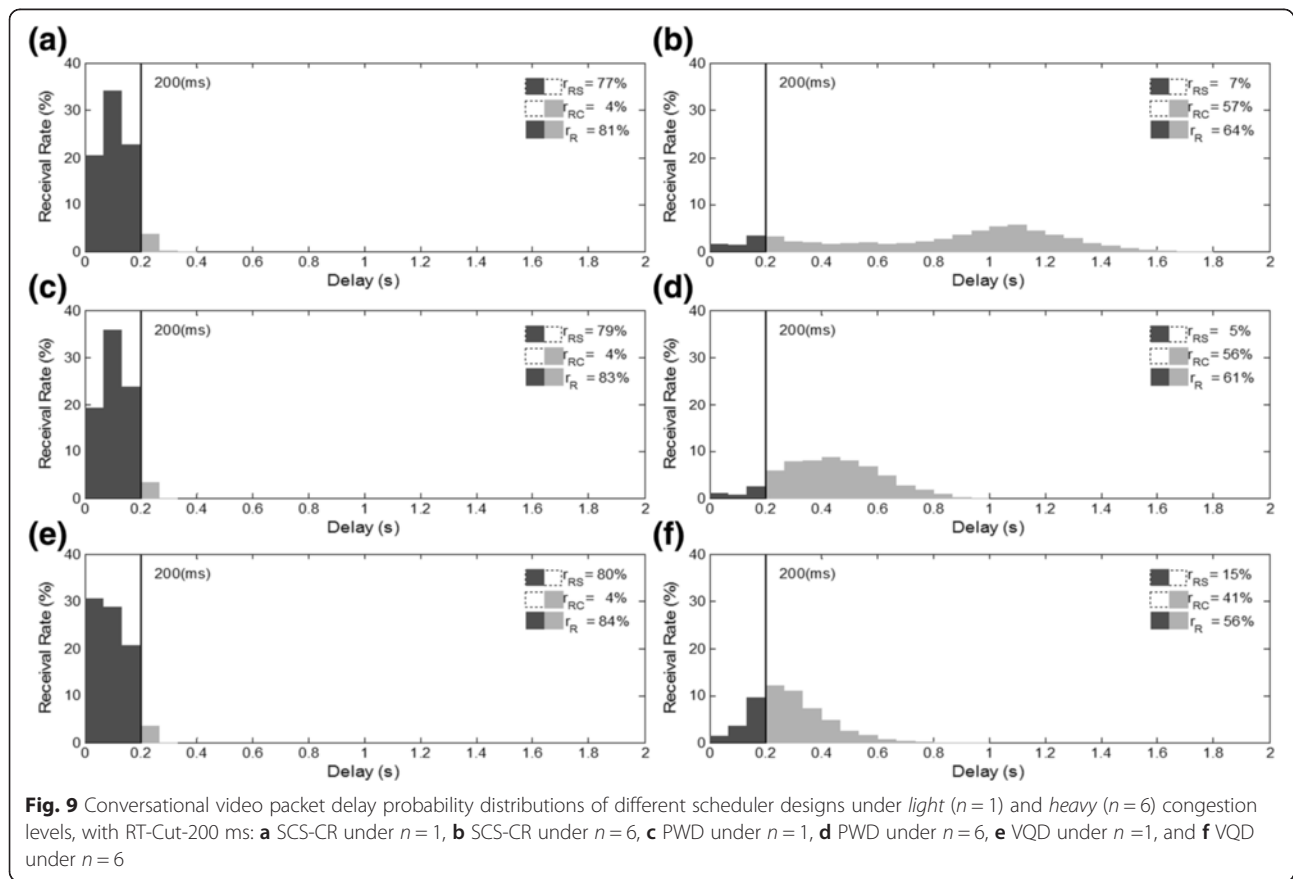


4.3 Effects of priority retries of I- and P-packets

The design principle for *priority retries* of I- and P-packets has been described and discussed in Section 3.3. Figure 6 presents a series of studies on the effects of *priority retries* of I- and P-packets on Average Y-PSNR variations of different schedulers (SCS-CR, PWD, VQD) under various congestion cases, with *loose* and *tight* real-time cuts (i.e. RT-Cut-400 ms and RT-Cut-200 ms), respectively, where the limit of retries ranges from 0 to 4 for I-packets and from 0 to 3 for P-packets, requiring the limit of retries for I-packets be always larger than that for P-packets. A common agreement has been observed: every design reaches its best performance when one sets the limits to be two retries for I-packets, and no retry for P-packets, denoted as I_2P_0 . For further performance studies throughout the rest of this article, I_2P_0 is the default case for all the designs in order to provide fair performance comparisons.

As a deeper insight to understand the PLR due to over-retries for multi-access physical collisions, Fig. 7 verifies and supports the prediction of PLR from the *physical collision probability*, based on Eqs. (4) to (6). To study the multi-access effect with the number of wireless stations, a scenario with N_s wireless stations sending

traffic to the Internet via an AP was adopted. Figure 7a demonstrates the physical collision probability without retries as a function of N_s for comparisons between prediction and simulation data from 802.11, 802.11e and 802.11aa, where a common value of initial contention window ($W = 32$) was adopted and only a single saturated CBR flow was sent, via AC_BK in the case of the latter two and thus they share the same results. It is seen that the above physical collision probabilities all increase with N_s , as well expected. There are good agreements among them in general, except that the simulation data for both 802.11e and 802.11aa are slightly lower when $N_s = 15$ and 20, which may well be due to a longer default value of *fixed* backoff (AIFS) adopted for both 802.11e and 802.11aa than the one (DIFS) adopted for 802.11. Figure 7b shows a good agreement of PLR between prediction and simulation data ($N_s = 10$) for 802.11, where the agreement becomes better as the number of maximum transmissions increases, particularly in terms of log scaling for PLR. Figure 7c further extends such log-scale comparisons of PLR between prediction and simulation data ($N_s = 4, 6, 8, 10$) for both 802.11e and 802.11aa, and the agreements are still good in general, despite not fully within the error bars (at the 95 % confidence level), but still on the same order



of magnitude. These show a strong support for our prediction and understanding of Eq. (6), which predicts PLR from the *physical collision probability* given by Eqs. (4) and (5).

4.4 Average queue lengths

According to the experimental settings in Table 2, the average length variations with congestion case n are depicted for the five AC queues of the AP (AAC_VO, AC_VI, AAC_VI, AC_BE, and AC_BK), as shown in Fig. 9, where $q_{limit} = 50$ is adopted for each of these five AC queues. Note that AC_VO was kept to be empty for simplicity. For cross-traffic of non-video types, there is no apparent difference among different designs except that SCS-CR is somewhat more congested than PWD

and VQD at $n = 6$ for AAC_VO and at $n = 1$ for AC_BE. Actually, the focus should be placed on AC_VI and AAC_VI, respectively, for *real-time (conversational)* and *non-real-time (on-demand)* video performances. The main messages are delivered as follow.

- For AC_VI, the proposed VQD design outperforms the others in the sense that it is least congested at all the congestion cases, and its superiority is amplified when n increases.
- For AAC_VI, the superiority of VQD over the others is still clearly seen, but no obvious increasing trend with n is observed since the ratio of scheduling probability for AAC_VI to

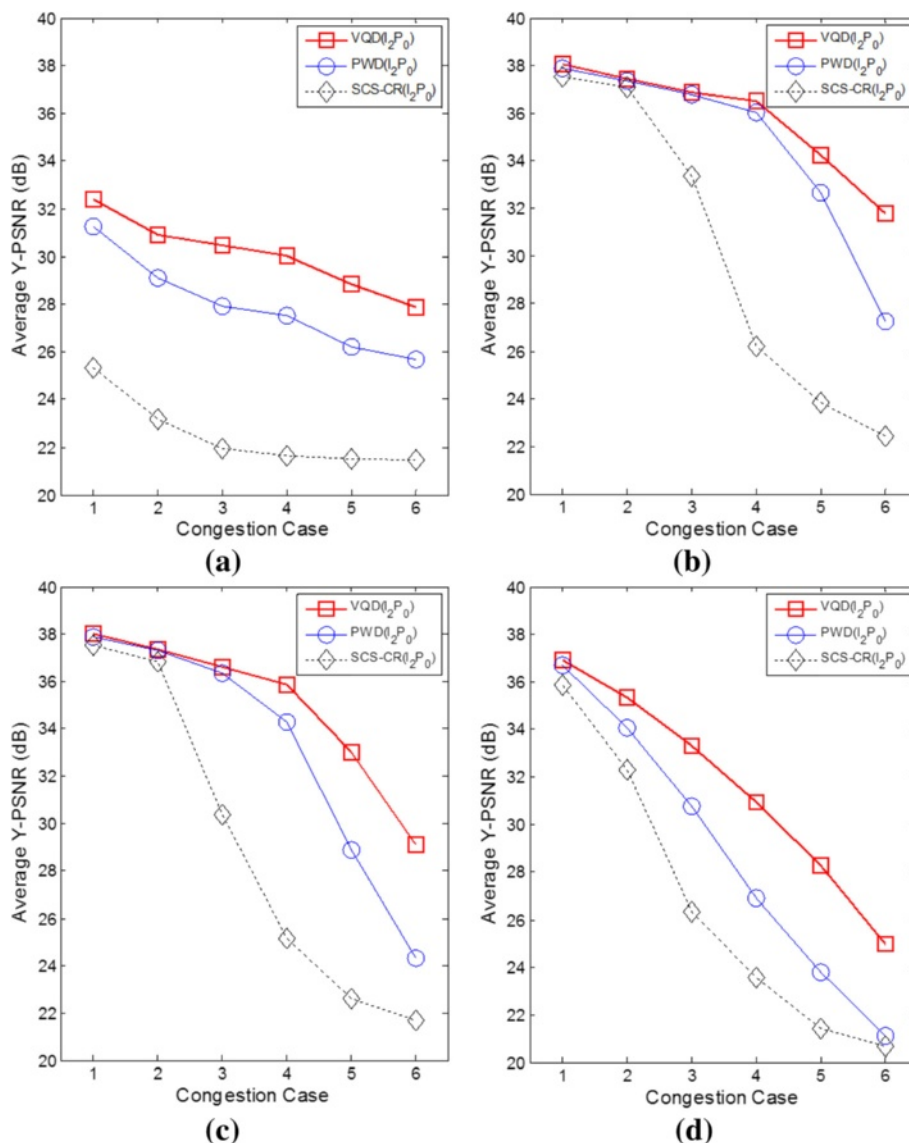


Fig. 10 Average Y-PSNR variations with n under different time constraints: **a** AAC_VI, **b** AC_VI with RT-Cut-400 ms, **c** AC_VI with RT-Cut-300 ms, and **d** AC_VI with RT-Cut-200 ms (Container QCIF, medium-slow)

that for AC_VI is low in general, and thus the average queue lengths of AAC_VI are longer than those of AC_VI for all the congestion cases, even at $n = 1$.

To sum up, a *win-win game* for both *real-time* and *non-real-time* video applications has been achieved by the proposed VQD design in terms of the *shortest* average queue length for each congestion case.

4.5 Packet delay probability distributions with/without a real-time constraint

In order to gain a deeper understanding for the effect of the aforementioned *real-time cuts* on the performances of different designs for *conversational* video communication, it is helpful to observe their corresponding packet delay probability distributions. Figure 9 shows the packet delay probability distributions of different scheduler designs under *light* ($n = 1$) and *heavy* ($n = 6$) congestion levels. Note that the total area under each probability distribution represents the percentage of the successfully *received* video packets (r_R), as defined in Eq. (8). When a *tight* real-time constraint such as RT-Cut-200 ms is applied, each distribution is divided into two parts: a *dark* shaded area representing the percentage of *received and survived* video packets (r_{RS}) and a *light* shaded area standing for the percentage of *received but cut-away* video packets (r_{RC}), as defined in Eqs. (9) and (10), respectively. Note again that these three ratios are normalised to the *total* number of video packets (N_{total}) in a given reference video sequence for the sake of comparison in the same scale.

Table 3 (r_{RS} , r_{RC} , r_R) of AC_VI versus congestion case n , due to RT-Cut-200 ms, with the leading values from the target designs written in bold-face

	Congestion case					
	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
r_{RS}						
r_{RC}						
r_R						
VQD	80 %	70 %	52 %	42 %	29 %	15 %
	4 %	9 %	16 %	23 %	31 %	41 %
	84 %	79 %	68 %	65 %	60 %	56 %
PWD	79 %	68 %	50 %	27 %	12 %	5 %
	4 %	12 %	25 %	43 %	54 %	56 %
	83 %	80 %	75 %	70 %	66 %	61 %
SCS-CR	77 %	65 %	30 %	20 %	9 %	7 %
	4 %	15 %	50 %	55 %	59 %	57 %
	81 %	80 %	80 %	75 %	68 %	64 %

$$r_R = \frac{N_{received}}{N_{total}} \times 100 \% \quad (8)$$

$$r_{RS} = \frac{N_{received \text{ and survived}}}{N_{total}} \times 100 \% \quad (9)$$

$$r_{RC} = \frac{N_{received \text{ but cut-away}}}{N_{total}} \times 100 \% \quad (10)$$

In addition, in terms of packet loss rate, $(1 - r_R)$ stands for the *real* loss rate due to *transmission* while r_{RC} represents the *effective* loss rate due to RT-Cut-200 ms. The main messages delivered from Fig. 9 are listed below.

- In *light* congestion, all the packet delay distributions are mostly focused below 200 ms. However, it still clearly indicates that $\bar{d}_{VQD} < \bar{d}_{PWD} < \bar{d}_{SCS-CR}$, where \bar{d} stands for the average packet delay.
- In *heavy* congestion, all the packet delay distributions are spread out in different extents: SCS-CR is spread out most seriously, PWD secondly seriously, and the proposed VQD least seriously in the sense that the inequality $\bar{d}_{VQD} < \bar{d}_{PWD} < \bar{d}_{SCS-CR}$ still holds.

Table 3 summarises the values of (r_{RS} , r_{RC} , r_R) under various congestion levels, where the leading values (i.e. the maximum r_{RS} , the minimum r_{RC} , and the maximum r_R) among the target designs under comparison are written in bold-face. These values can be reasoned as follow.

- Although the proposed VQD design does not always have the maximum value in r_R (in fact, it always takes the final place in r_R except for $n = 1$ due to its effective packet dropping behaviour, particularly when the congestion level increases), it does always take the lead in both r_{RS} and r_{RC} for all the congestion levels, meaning that it keeps the minimum value in average packet delay no matter how the congestion level changes, as aforementioned.
- SCS-CR is at the other extreme; although it in general takes the lead in r_R , it is almost the all-

Table 4 Ratio of received packets (r_R) of AAC_VI versus congestion case n , where the *real* packet loss rate due to transmission can be derived from $(1 - r_R)$, with the leading values from the target designs written in bold-face

	Congestion case					
	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
r_R						
VQD	34 %	29 %	26 %	21 %	18 %	16 %
PWD	31 %	25 %	21 %	17 %	15 %	13 %
SCS-CR	35 %	28 %	23 %	21 %	20 %	19 %

time loser in both r_{RS} and r_{RC} . This originates from the fact that SCS-CR lacks an effective strategy in packet dropping to mitigate the congestion.

- The performance of PWD is in general in between VQD and SCS-CR; its performance is closer to VQD from *light* to *medium-low* congestions ($n = 1, 2, 3$), but closer to SCS-CR from *medium-high* to *heavy* congestions ($n = 4, 5, 6$).

So far, the above analyses in this subsection are all based on *real-time* video via AC_VI with a real-time cut, and the proposed VQD design in general takes the lead in both r_{RS} and r_{RC} , but almost the loser in r_R compared

to PWD and SCS-CR. For *non-real-time* video via AAC_VI, however, Table 4 gives a much better ranking in r_R for VQD; PWD becomes the all-time loser, SCS-CR is no longer the all-time winner, and VQD can even take the lead for half of the congestion cases. Considering the fact that both PWD and VQD root from a similar concept called *early prioritised packet dropping* depending on the congestion level, this result is somewhat surprisingly good, and it delivers the following messages:

- VQD outperforms PWD by a superior realisation of *prioritised early packet dropping* which is based on the coupling of video packet dropping probability

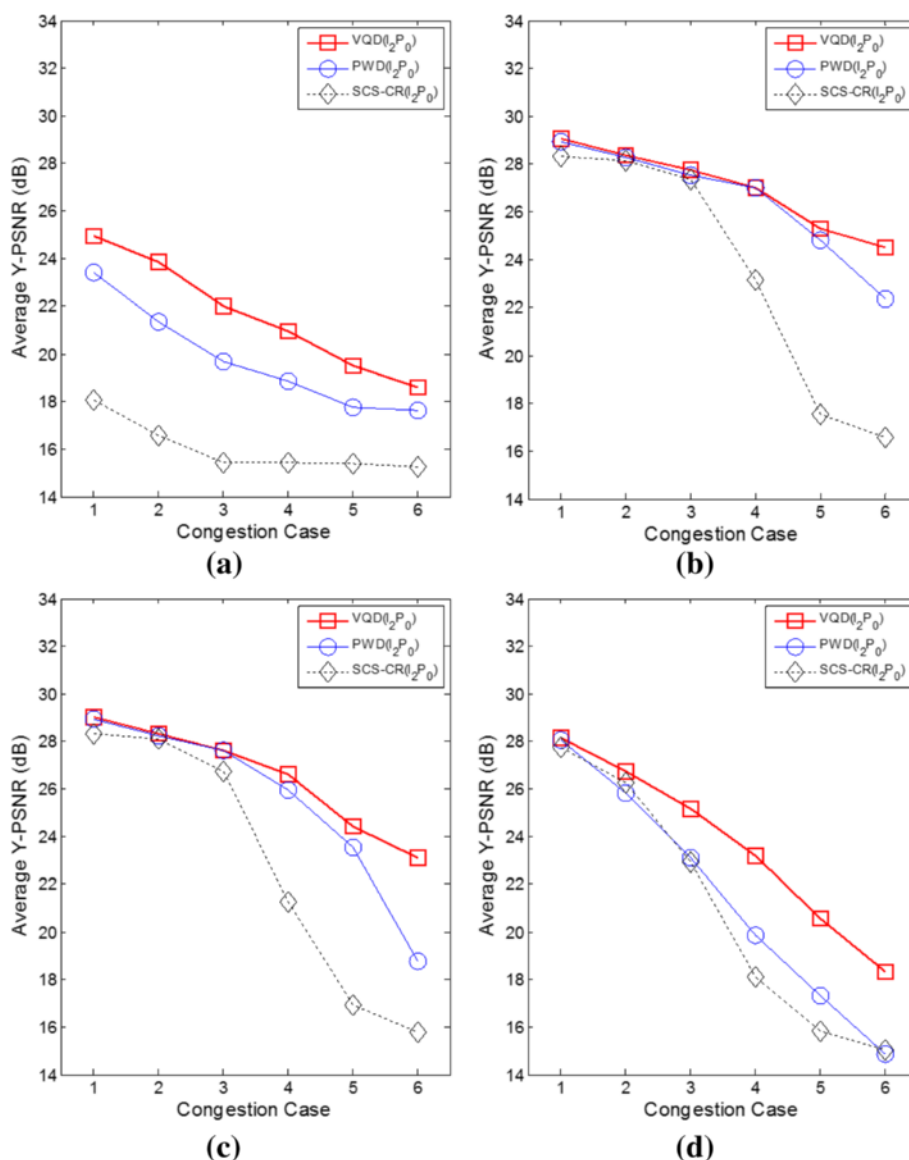


Fig. 11 Average Y-PSNR variations with n under different time constraints: **a** AAC_VI, **b** AC_VI with RT-Cut-400 ms, **c** AC_VI with RT-Cut-300 ms, and **d** AC_VI with RT-Cut-200 ms (*Foreman QCIF, fast*)

for a given priority with those higher priorities, as shown in Eq. (3).

- VQD can effectively alleviate the congestion level by *early dropping* video packets of less importance (i.e. those lower-priority ones) at congestion cases $n = 2, 3$ and 4.
- Although VQD is not as good as SCS-CR in the other cases in terms of r_R , Fig. 10 demonstrates that VQD is still the all-time winner in terms of Average Y-PSNR. Again, this supports that VQD can achieve a *win-win* game in both *real-time* and *non-real-time* video applications, namely it successfully tackles the dilemma problem in achieving *priority* and *fairness* simultaneously.

Section 4.6 will address more on the generality of such a win-win game.

4.6 Generality of performance superiority in average Y-PSNR

To show that the performance superiority of VQD over PWD and SCS-CR is general and not a special effect of a specifically adopted video sequence, more performance comparisons based on average Y-PSNR have thus been conducted on the aforementioned four reference video sequences of different YUV formats (*QCIF* and *CIF*) and motion levels (*slow* and *fast*), and the corresponding results are summarised in Figs. 10, 11, 12 and 13, each of which consists of several

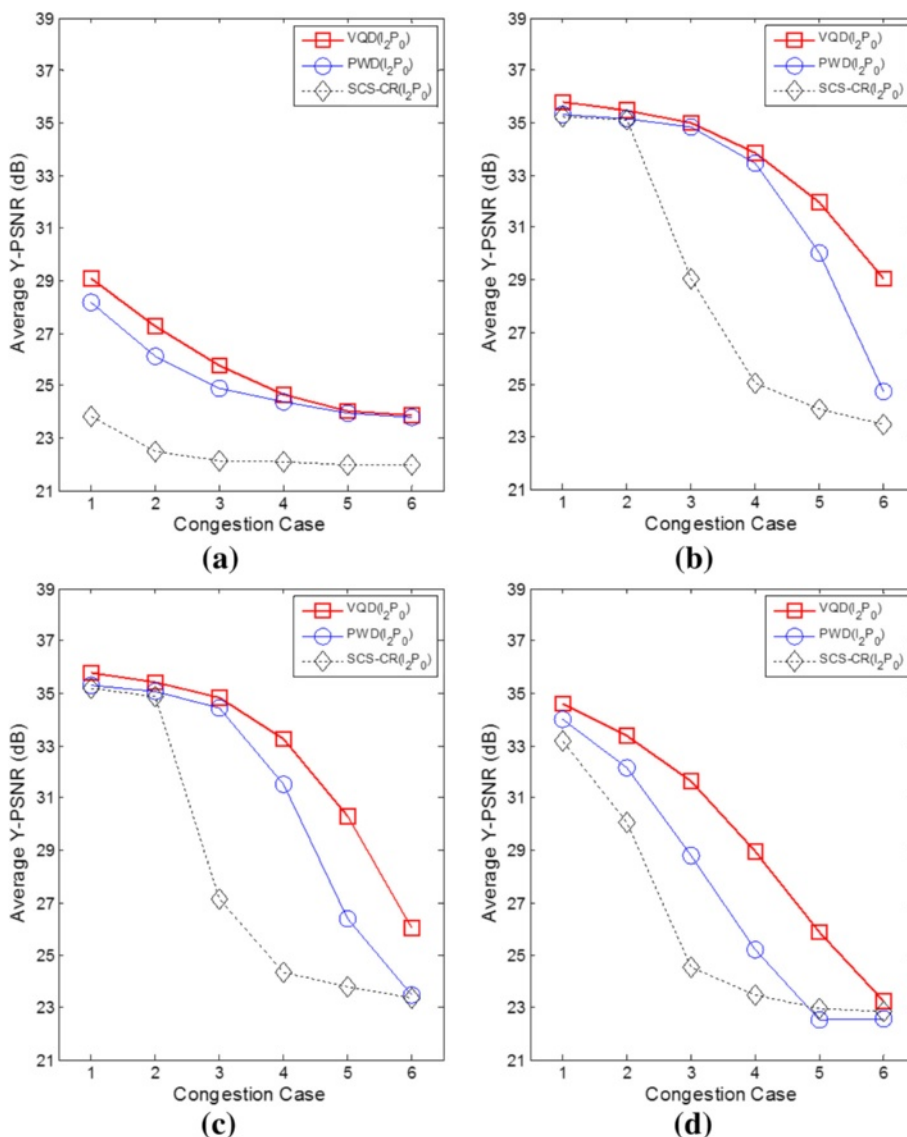


Fig. 12 Average Y-PSNR variations with n under different time constraints: **a** AAC_VI, **b** AC_VI with RT-Cut-400 ms, **c** AC_VI with RT-Cut-300 ms, and **d** AC_VI with RT-Cut-200 ms (*Mother-and-Daughter CIF, slow*)

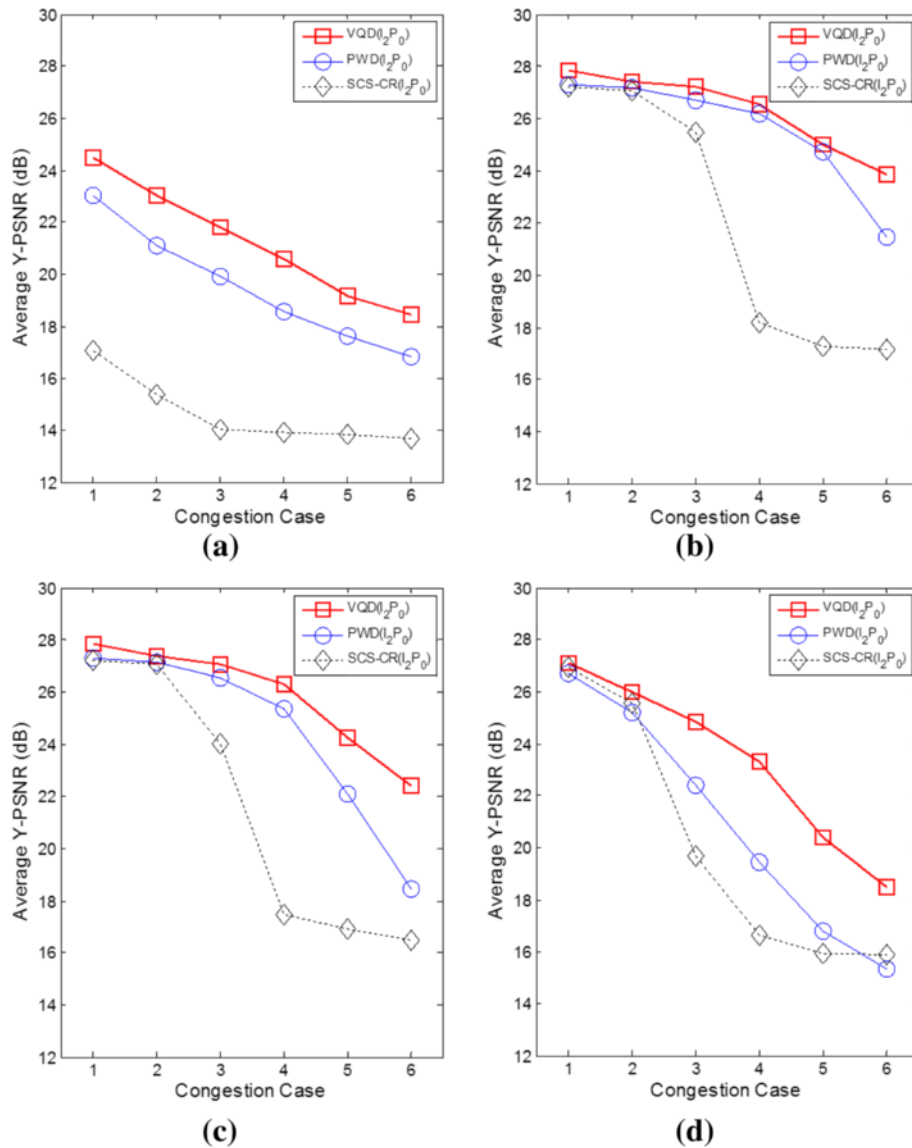


Fig. 13 Average Y-PSNR variations with n under different time constraints: **a** AAC_VI, **b** AC_VI with RT-Cut-400 ms, **c** AC_VI with RT-Cut-300 ms, and **d** AC_VI with RT-Cut-200 ms (*Carphone CIF, fast*)

subfigures to present the Average Y-PSNR variations with congestion case n under different time constraints: (a) AAC_VI without any real-time cut, (b) AC_VI with RT-Cut-400 ms, (c) AC_VI with RT-Cut-300 ms, and (d) AC_VI with RT-Cut-200 ms. These figures deliver the following common messages.

- The performance superiority of VQD over PWD and SCS-CR is indeed a general phenomenon, and there is no strong effect due to the format or motion level of the adopted video sequence.
- VQD can always achieve a *win-win game* over the others in both the performances of AAC_VI

(*without* any real-time constraint) and AC_VI (*with* different real-time constraints), and its real-time performance gains over the others are amplified in general as the congestion level increases.

- Although the congestion-robustness capabilities of different designs all go down with the tightness of the adopted *real-time constraint* as expected, the performance gains of VQD over PWD (the all-time second place) also become more visible. This shows the true power of VQD in its efficiency of early packet dropping based the coupling of component queue lengths of different priorities.

5 Conclusions

In this article, we have presented a novel scheduler design called VQD for robust *real-time* video communication over IEEE 802.11aa wireless LANs. Based on the three novel concepts, namely *conditional priority weighting* for non-real-time I-packets from AAC_VI, *virtual queue dropping* of P-packets based on priority-coupled component queue lengths and *priority retries* of I- and P-packets, the proposed VQD design has been demonstrated to be in general superior to other target designs such as SCS-CR and PWD. The optimal value of κ that determines the best *tanh* functional shape, namely the virtual queue-based packet dropping probability functions for AC_VI and AAC_VI, has been found to be 1, despite that the performance differences using its neighbouring values are not so large, between 1~2 dB in general. The retry effect for lost I- and P-packets indicates that the best retry limits are twice for I-packets, zero for P-packets. This result is common to all the target designs, and thus all the performance comparisons are based on such a combination of retry limits.

VQD has revealed its superiority over PWD and SCS-CR in the sense that it has the *shortest* average queue lengths for both AC_VI and AAC_VI, the *minimum* average video packet delay, the *maximum* number of *received and survived* video packets, and the *highest* value of average Y-PSNR, considering their variations with different congestion levels under various real-time constraints. Moreover, the superiority of VQD is a common phenomenon among different video sequences, and not obviously affected by the format or motion level of the adopted video source. It is also worth mentioning that VQD has won a *win-win game* over the other designs in both the performance gains of AC_VI and AAC_VI, namely a dilemma in achieving *priority* and *fairness* simultaneously for scheduling between the *primary* and *alternative* queues in the access category of video for delivering video streams *with* and *without* real-time constraints, respectively.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This study was supported by the Taiwan National Ministry of Science and Technology under grants MOST 102-2221-E-155-007, MOST 103-2221-E-155-052, MOST 104-2218-E-155-002 and MOST 105-2218-E-155-001.

Received: 28 July 2015 Accepted: 7 May 2016

Published online: 21 June 2016

References

1. IEEE Std 802.11-2012, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std (2012)
2. IEEE Std 802.11-1999, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std (1999)
3. G Bianchi, Performance analysis of the IEEE 802.11 distributed coordinated function. *IEEE J Sel Areas Commun* **18**(3), 535–547 (2000)
4. IEEE Std 802.11e-2005 (Amendment 8 to IEEE Std 802.11-1999), Medium access control (MAC) quality of service enhancements. IEEE Std (2005)
5. A Ksentini, M Naimi, A Gueroui, Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture. *IEEE Commun Mag* **44**(1), 107–114 (2006)
6. CH Lin, CK Shieh, CH Ke, NK Chilamkurti, S Zeadally, An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11e WLAN. *Telecommun Syst* **42**(3), 223–234 (2009)
7. N Chilamkurti, S Zeadally, R Soni, G Giambene, Wireless multimedia delivery over 802.11e with cross-layer optimization techniques. *Multimed Tools Appl* **47**(1), 189–205 (2010)
8. WP Lai, EC Liou, A novel cross layer design using comb-shaped quadratic packet mapping for video delivery over 802.11e wireless ad-hoc networks. *EURASIP J Wireless Commun Net* **2012**(1), 59 (2012)
9. CH Lin, YC Wang, CK Shieh, WS Hwang, An unequal error protection mechanism for video streaming over IEEE 802.11e WLANs. *Computer Networks* **56**(11), 2590–2599 (2012)
10. Z Wan, B Xiong, N Ghani, AV Vasilakos, L Zhou, Adaptive unequal protection for wireless video transmission over IEEE 802.11e networks. *Multimed Tools Appl* **72**(1), 541–571 (2014)
11. IEEE Std 802.11aa-2012, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Amendment 2: MAC enhancements for robust audio video streaming. IEEE Std (2012)
12. K Kosek-Szott, M Natkaniec, S Szott, A Krasilov, A Lyakhov, A Safonov, What's new for QoS in IEEE 802.11? *IEEE Network* **27**(6), 95–104 (2013)
13. IEEE Std 802.11ae-2012, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Amendment 1: prioritization of management frames. IEEE Std (2012)
14. A Lyakhov, M Yakimov, Analytical study of QoS-oriented multicast in wireless networks. *EURASIP J Wireless Commun and Net* **2011**(1), 307507 (2011)
15. A Banchs, A de la Oliva, L Eznarriaga, DR Kowalski, P Serrano, Performance analysis and algorithm selection for reliable multicast in IEEE 802.11aa wireless LAN. *IEEE Trans Veh Technol* **63**(8), 3875–3891 (2014)
16. P Salvador, L Cominardi, F Gringoli, P Serrano, A first implementation and evaluation of the IEEE 802.11aa group addressed transmission service. *ACM SIGCOMM. Comput Commun Rev* **44**(1), 36–41 (2014)
17. IEEE Std 802.1Q-2011, IEEE standard for local and metropolitan area networks—media access control (MAC) bridges and virtual bridge local area networks. IEEE Std (2011)
18. E Charfi, L Chaari, L Kamoun, PHY/MAC enhancements and QoS mechanisms for very high throughput WLANs: a survey. *IEEE Commun Surv Tut* **15**(4), 1714–1735 (2013)
19. P. Serrano, P. Salvador, V. Mancuso, Y. Grunenberger, Experimenting with commodity 802.11 hardware: overview and future directions. *IEEE Commun Surv Tut* **17**(2), 671–699 (2015)
20. WP Lai, EC Liou, A novel scheduler design for wireless video over 802.11aa networks using priority weighting and dropping. *Int J Future Generation Commun Net* **6**(4), 137–146 (2013)
21. HY Kang, CL Chang, The decision scheme of frame importance factor for video streaming system, in *Proc. 5th Int Conf Broadband, Wireless Computing, Communication and Applications (BWCCA'10), Fukuoka, Japan*, 2010, pp. 517–521
22. IEEE Std 802.17-2004, Part 17: Resilient packet ring (RPR) access method and physical layer specifications. IEEE Std (2004)
23. IEEE Std 802.1Qav-2009, IEEE standard for local and metropolitan area networks—virtual bridged local area networks, Amendment 12: Forwarding and queuing enhancements for time-sensitive streams. IEEE Std (2009)
24. H Yu, T Sakurai, Collision probability in saturated IEEE 802.11 networks, in *Proc. Australian Telecommunication Networks and Application Conference, (ATNAC'06), Australia*, 2006, pp. 21–25
25. L Zhu, FR Yu, B Ning, T Tang, Cross-layer handoff design in MIMO-enabled WLANs for communication-based train control (CBTC) systems. *IEEE J Sel Areas in Comm (JSAC)* **30**(4), 719–728 (2012)
26. F Yu, V Krishnamurthy, Optimal joint session admission control in integrated WLAN and CDMA cellular networks with vertical handoff. *IEEE Trans Mobile Comput* **6**(1), 126–139 (2007)
27. K. Fall, K. Varadhan, The ns manual. <http://www.isi.edu/nsnam/ns>. Accessed 4 Apr 2015

28. Video Trace Library (US Arizona State University or Germany TKN Telecommunications Networks Group), <http://trace.eas.asu.edu/yuv> or <http://www.tkn.tu-berlin.de/research/evalvid>. Accessed 5 Mar 2015
29. ITU-T Rec H.264, Advanced video coding for generic audiovisual services. ITU Std (2010)
30. EvalVid (Germany TKN Telecommunications Networks Group), <http://www.tkn.tu-berlin.de/research/evalvid>. Accessed 10 July 2014