

RESEARCH

Open Access

A novel cross-layer design using comb-shaped quadratic packet mapping for video delivery over 802.11e wireless ad hoc networks

Wen-Ping Lai* and En-Cheng Liou

Abstract

Cross-layer design is a promising direction and challenging issue for quality delivery of multimedia over wireless networks. This article proposes a cross-layer design which can substantially enhance the transmission quality of video streaming over 802.11e ad hoc networks. The proposed design consists of two parts: a *dispersive video frame importance* (DVFI) scheme in the application layer that can correctly label the priorities of video packets, and a *comb-shaped quadratic mapping* (CQM) algorithm in the *medium access control* (MAC) layer that can provide a better congestion control mechanism among the multiple *access category* (AC) queues than 802.11e Enhanced Distributed Channel Access (EDCA) for wireless video delivery. The DVFI value of a video frame is measured from its own transmission loss and the accompanied impact to other coding-dependent video frames, and is accordingly adopted to prioritize the video frame associated packets. When these prioritized video packets arrive at the reserved AC queue in the MAC layer, the CQM algorithm can accordingly provide multi-branched service differentiation among them by dynamically downward mapping less significant video packets to lower-priority ACs based on the instant congestion level of the reserved AC queue, which can thus offer a better channel access resource to those video packets of higher priorities. In addition, the minimum-delay-time rule embedded in the CQM will select a better destiny for those downward mapped video packets among the lower-priority AC queues. Under various video-input-rate and cross-traffic tests, the simulation results show that this study outperforms the existing works, including EDCA, Adaptive Mapping, and Enhanced Adaptive Mapping.

Keywords: congestion control, cross-layer design, video transmission quality, comb-shaped quadratic packet mapping.

1. Introduction

Over the past decade, IEEE 802.11 has become the dominant technology for wireless local area network (WLAN) because of low cost and easy deployment, and the promise of high-quality multimedia service is beyond doubt one of the major driving forces of the next generation WLAN. Since multimedia over WLAN is bandwidth-restricted and delay-sensitive, and the conventional 802.11 *medium access control* (MAC) mechanism [1,2] is not equipped with quality of service (QoS), quality delivery of multimedia over the conventional 802.11 WLAN is very challenging.

To address this issue, the IEEE 802.11e standard [3] was proposed in late 2005 to offer multimedia service differentiation. This new standard enables service differentiation in the MAC layer by a special congestion control/avoidance mechanism based on the concept of multiple *access category* (AC) queues within a mobile station and *virtual collision* resolution among them. In other words, the conventional distributed coordination function of 802.11 MAC has only a single first-in-first-out (FIFO) queue so that collisions occur among the wireless stations when they try to access the wireless channel simultaneously, whereas the *enhanced distributed channel access* (EDCA) of 802.11e MAC offers multiple prioritized AC queues, given different resources for channel access and denoted by $\{AC[n], n = \text{priority}\}$, to allow for resolving an earlier-stage collision among these ACs within a single wireless

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

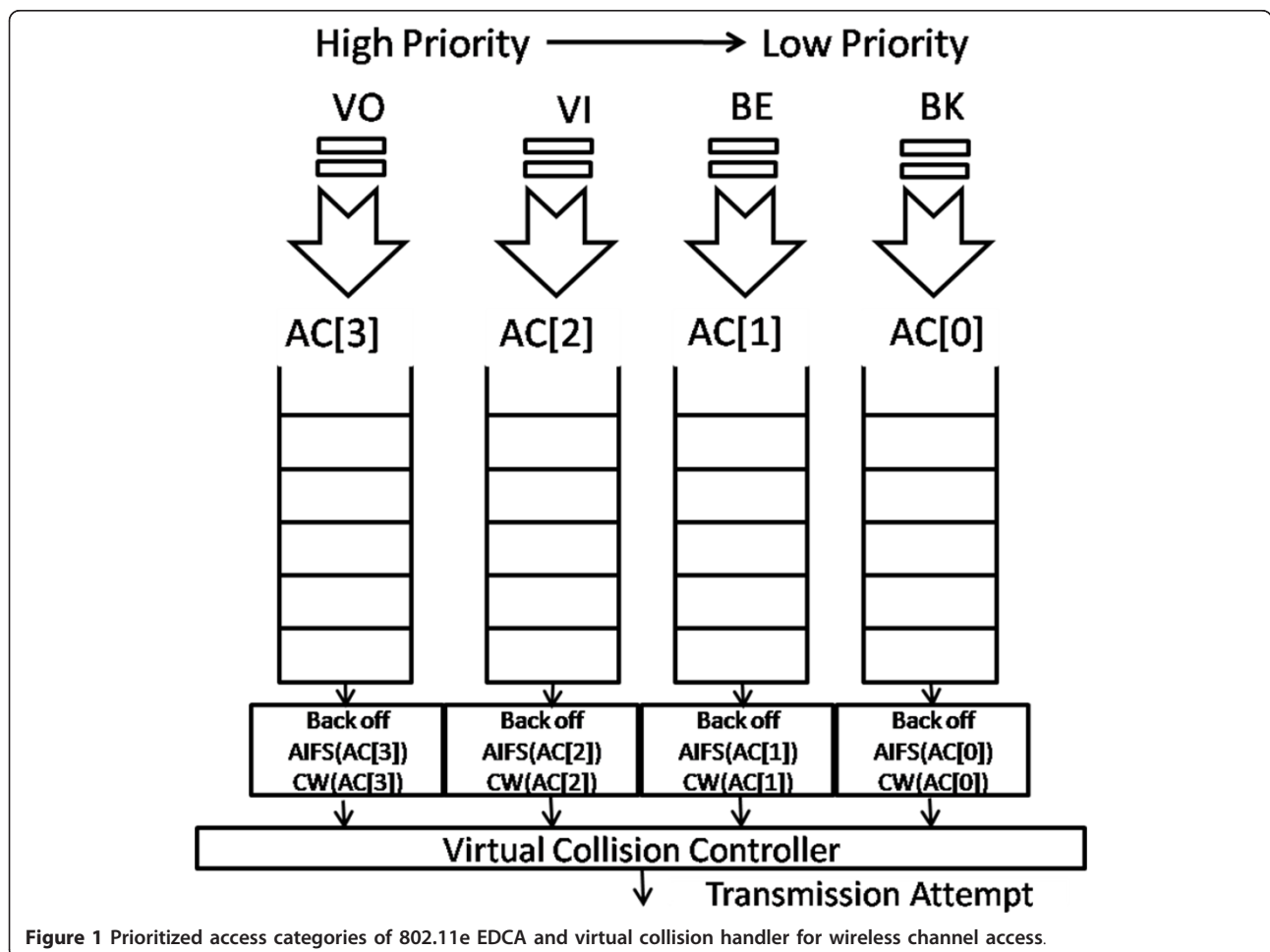
station before a packet is sent out to the wireless channel. As shown in Figures 1 and 2, such an earlier-stage resource contention is treated by a virtual collision handler, and service differentiation is achievable by assigning various prioritized traffic flows to their associated AC[n] where prioritized network resources are allocated.

In this manner, multimedia can be given a higher priority than other delay-insensitive data traffic flows. However, the service differentiation offered by 802.11e is still very primitive. Namely, *voice* is assigned to the highest-priority AC (AC_VO, or AC[3] for short), *video* to the second-priority AC (AC_VI, or AC[2] for short), and the others to *best-effort* and *background* ACs (AC_BE and AC_BK, or AC[1] and AC[0] for short, respectively). The priorities of these ACs are determined by their corresponding operational parameters such as *arbitration inter frame space* (AIFS), *contention window* (CW) size, and *transmission opportunity limit* (TXOPlimit). In the literature, some mechanisms have been proposed to improve the 802.11e performance through adjusting the operational parameters, e.g., CW [4-7] and TXOPlimit [8-10], as well as

data transmission rate [11,12]. However, none of them has taken into account the importance levels of video packets.

An obvious problem for 802.11e lies in the fact that the reserved AC queue for video traffic, i.e., AC[2], is FIFO based, and thus no room is available for further differentiation among video packets themselves based on their importance level which is inferable by a modern video coding technology from the application layer. This certainly limits the quality performance of video over 802.11e. In recent years, cross-layer design has become a promising research direction and is still a challenging issue for improving the *transmission quality* of multimedia over wireless networks.

To further support service differentiation among video packets, the authors of [13] proposed a cross-layer framework for H.264-based video streaming over 802.11e, which introduced the concept of *Static mapping* (SM), where prioritized video packets based on their H.264 coding importance from the application layer were statically assigned to different ACs in the MAC layer, and achieved a better video transmission quality than EDCA. Despite



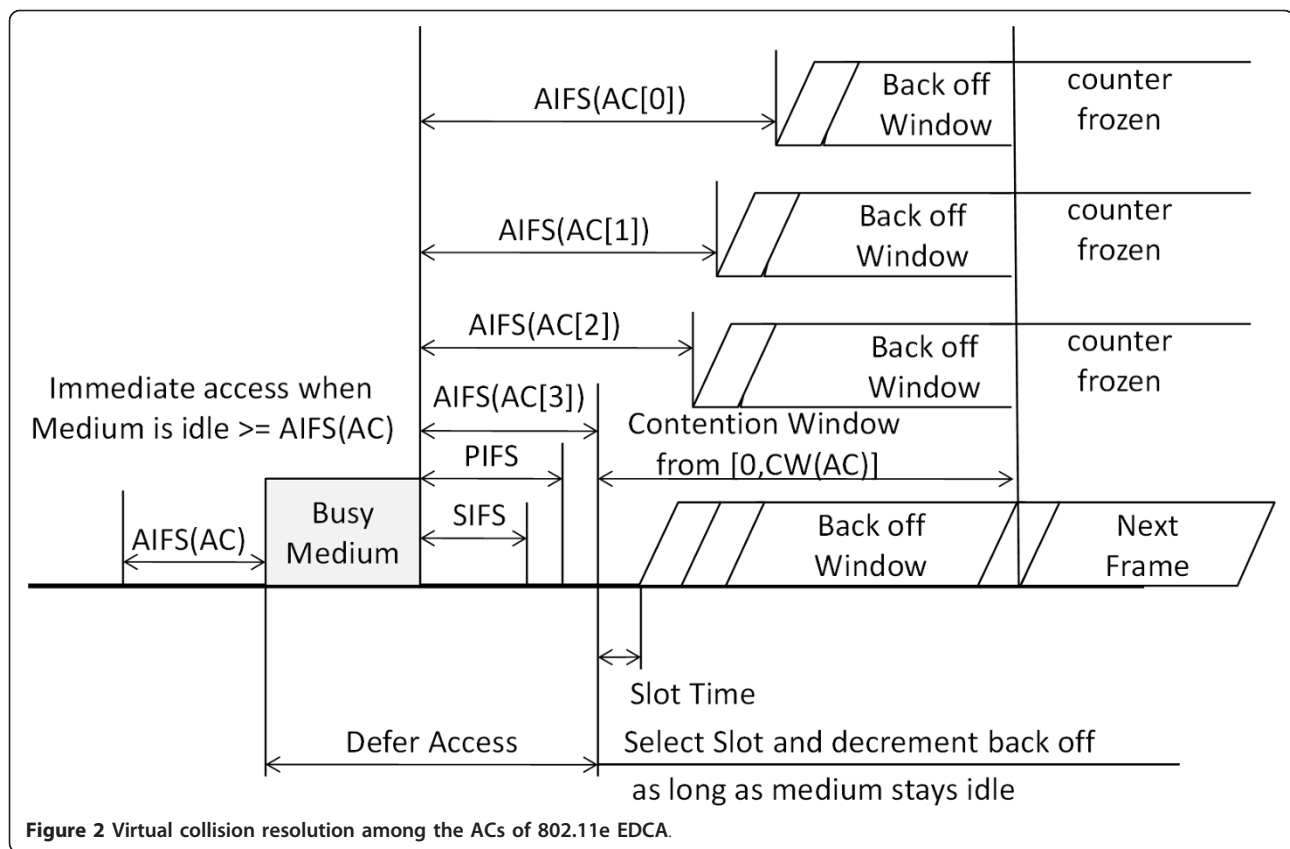


Figure 2 Virtual collision resolution among the ACs of 802.11e EDCA.

this, SM does not change the fundamental queuing behaviors of ACs, namely the individually prioritized ACs are still all FIFO based. In contrast to [13], the authors of [14] proposed an *Adaptive mapping* (AM) algorithm for MPEG-4-based streaming over 802.11e, where the packet importance classification scheme was simply based on the I/P/B video frame types and the main addressed issue was focused on finding an *active queue management* (AQM) for AC[2]. In other words, the AC[2] (i.e. AC_VI) in the AM algorithm is no longer a FIFO queue, but a *multi-level random early detection* (M-RED) queue, where each I/P/B packet follows a specific RED-like packet mapping function, i.e., this algorithm modified the random early 'drop' concept into random early 'downward mapping' to the lower-priority AC queues. They showed that AM outperforms SM and EDCA in various traffic loading conditions. They also found that SM could be even worse than EDCA in light congestions if the lower-priority queues have concurrent cross traffic. The authors of [15] proposed an *enhanced version* of AM (EAM) in the H.264 coding scheme, where the same AQM was applied to AC [2] with a different set of *low* and *high* thresholds ($threshold_{low}$, $threshold_{high}$) for the queue length of AC[2], and a more subtle consideration was given to those downward transitioned video packets into AC[1] or AC[0].

This article proposed a novel cross-layer design called DVFI+CQM, where a hyperfine video packet priority scheme based on the concept of *dispersive video frame importance* (DVFI) was adopted in the application layer while a corresponding AQM-based packet mapping function called *comb-shaped quadratic mapping* (CQM) was proposed in the MAC layer. In the following sections, the corresponding details of both DVFI and CQM are described and their combined superiority over the existing works are presented and explained.

The remainder of this article is structured as follows. Section 2 describes the background of the video coding-based packet importance and priority, the 802.11e standard, and the related studies addressing the cross-layer design issue for video delivery over 802.11e. The proposed DVFI+CQM framework is detailed in Section 3, followed by the experimental results and analyses in Section 4. Section 5 concludes this study and outlooks the future works.

2. Background

This section glances at the evolution of video coding standards, introduces the concept of video coding-based packet priority, explains the mechanism of 802.11e EDCA, and overviews the related studies.

2.1. Video coding-based packet priority

The existing video coding standards are dominated by two tracks: the ITU H.26x family and the ISO/IEC MPEG-x family [16]. H.261 was mainly proposed for video conferencing, MPEG-1 for VCD, H.262/MPEG-2 for DVD and DVB, MPEG-4 [17] for low-bit-rate video streaming, and H.264/MPEG-4 AVC [18] for high definition TV. In recent years, scalable video coding [19] for scalable video streaming over heterogeneous networking bandwidths or terminal displays and multi-view video coding [20] for stereoscopic video/3D or free-viewpoint TV have become the main streams of research interest and may generate promising applications.

To achieve a better rate-distortion, i.e., better quality for a given bit rate or lower bit rate for the same quality, the general trend for advanced video coding is to increase the complexity of the coding algorithms, including spatial prediction *intra* a video frame and temporal prediction *inter* video frames. If a video frame referenced by other video frames for decoding is lost, it will cause error propagation to those frames, and may make them undecodable unless the lost video frame is error concealed. In general, there are three types of coded video frames: I (Intra-coded) frames, P (Predictive-coded) frames, and B (Bi-directionally predictive-coded) frames. In order to shorten or stop the error propagation, a periodic Group of Pictures (GOP) structure should be adopted.

Figure 3a demonstrates a typical GOP structure with a period of nine video frames (i.e., IBBPBBPBB), where the predictive dependences among the I/P/B video frames are illustrated. The simplest importance level of the I/P/B frames is: $I > P > B$. The reasons are given as follow. (1) The I frame is intra-coded and its decoding is independent of other frames, but it is referenced by its succeeding P and B frames within the same GOP structure; the loss or damage of a I frame can cause a serious error propagation to its subsequently received P and B frames and make them undecodable unless error concealed, and thus the I frame is the most important one. (2) The P frame is uni-directionally predictive-coded and its decoding needs to reference its preceding I or P frame, and it is also referenced by its succeeding P and surrounding B frames; thus, it should take the second place in the importance level. (3) The B frame is bi-directionally predictive-coded and its decoding needs to reference both its preceding and succeeding I or P frames, but it will not be referenced by other video frames, and thus it causes no error propagation and should be marked as the least important one. This simple I/P/B frame importance scheme was adopted by both AM [14] and EAM [15] in their cross-layer design frameworks. However, it is obvious that such a simple video frame importance scheme ignores the potential importance differences among P frames, since the loss or damage of a P frame may potentially cause a

specific error propagation, the effect of which depends on its frame position in the GOP.

A further importance classification scheme among P frames called EPL has been proposed in [21]. Figure 3b demonstrates the basic idea of EPL for the {IPPP...} GOP patterns with periods of 9 and 12 video frames, where the video importance indexing for each video frame is quite straight forward: I is still the most important one, and the earlier position a P frame is located within a GOP, the more important it is. However, they did not apply the EPL concept to the scenario of 802.11e.

2.2. 802.11e EDCA

The objective of 802.11e EDCA and the service differentiated classes of ACs (voice, video, generic data) have been well described in Section 1. Figure 1 shows the priorities of these ACs, from high to low, where AC[3] has the highest one and AC[0] has the lowest one. The higher priority an AC has, the shorter delay and random backoff time for channel access it would be assigned with. The priority of each AC is assigned based on its QoS requirements using a set of four parameters (AIFS, CW_{min} , CW_{max} , TXOPlimit). Collisions among the ACs of an 802.11e QoS Station (QSTA) are resolved by its virtual collision handler. A better assignment of these parameters can lead to a more efficient transmission performance through these ACs [22-24]. Table 1 lists the values of the parameter sets for all the ACs of 802.11e EDCA.

Figure 2 shows the virtual collision resolution among the ACs of 802.11e EDCA. Immediate access can be granted for the transmission attempt of any AC[n] queue ($n = 0, 1, 2, 3$) only after the medium is idle longer than AIFS(AC[n]). In case of busy medium, a fixed time interval plus a random backoff time is needed before AC[n] can transmit a packet. AIFS(AC[n]) is the minimum time interval after which the QSTA detecting the idle channel can start a random backoff timer chosen from $[0, CW(AC[n])]$ for collision avoidance among ACs. For each channel access attempt, $CW(AC[n])$ is initialized to $CW_{min}(AC[n])$ and the random backoff timer starts to count down in units of slot when the channel is free for one AIFS. The station starts to transmit when the backoff timer reaches zero and the channel is still idle. If unsuccessful, $CW(AC[n])$ is doubled for another channel access attempt until up to $CW_{max}(AC[n])$. TXOPlimit(AC[n]) is the maximum duration which allows the QSTA to transmit a burst of MAC frames without entering another channel contention period. Notably, as shown in the last column of Table 1, AC[2] (video) has a longer TXOPlimit than AC[3] (voice), whereas the zero values of TXOPlimit for both AC[1] and AC[0] represent that the QSTA can only transmit one MAC frame in a new contention period.

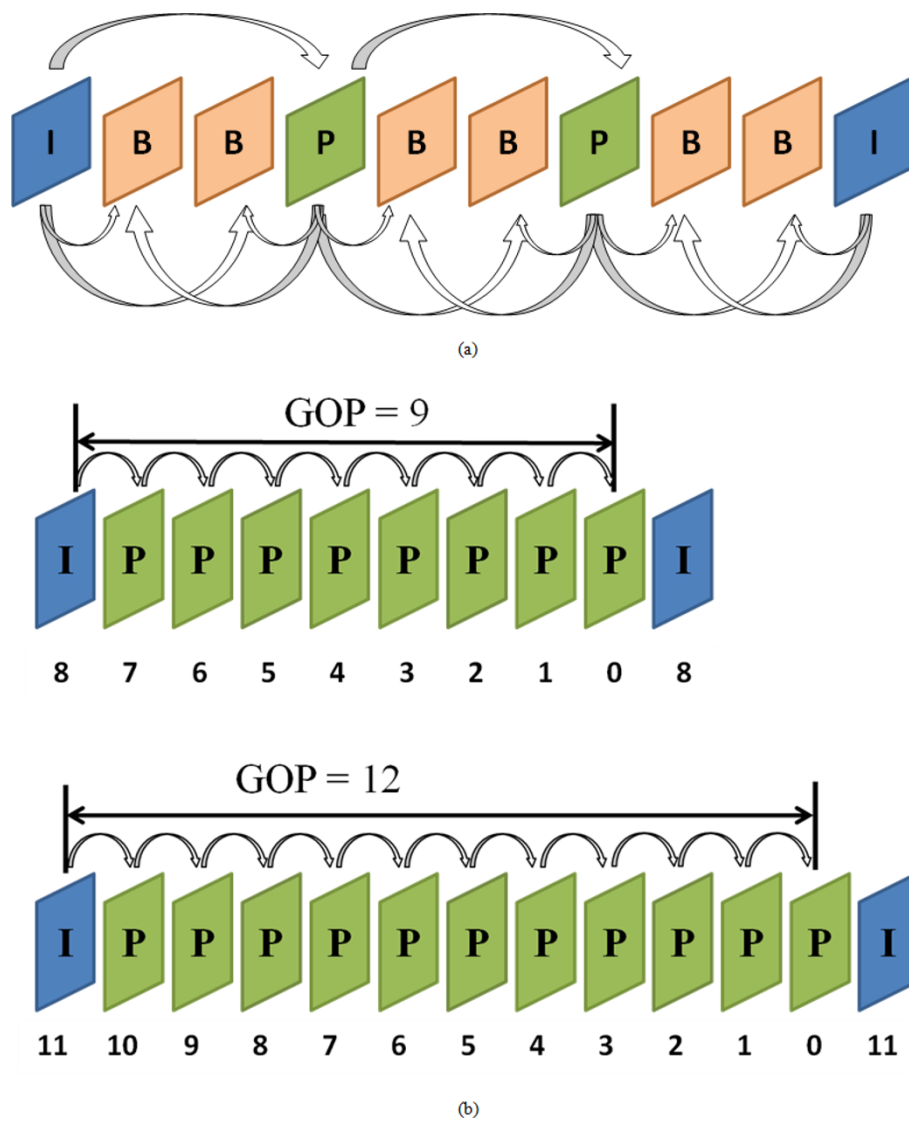


Figure 3 Coding dependences in various GOP structures: (a) A typical GOP structure and predictive dependences of I/P/B video frames, and (b) The video frame importance index (represented by the number below each video frame) for GOP = 9 and GOP = 12 in the EPL scheme.

2.3. Related studies

To address the performance limitation issue of 802.11e, Ksentini et al. [13] proposed a QoS cross-layer framework to exploit both the application layer and the 802.11e MAC layer features. In their work, *data-partitioning* for H.264 video source and packet marking in

NAL (Network Abstraction Layer) were applied in the application layer, and the SM algorithm assigned these packets to different ACs of 802.11e EDCA in the MAC layer based on their marked priorities. Their results showed that such a cross-layer design with further differentiation among video packets could provide a better quality than EDCA, where video packets are all mapped together into a single AC, namely AC[2]. As aforementioned, both the EDCA mechanism and the SM algorithm assumes that all the ACs are FIFO based, thus not AQM based, and this may lead to a dramatic video quality degradation in heavy network congestion.

To address this issue, instead of mapping prioritized video packets into different ACs, Lin et al. [14] proposed the AM algorithm based on the priorities of video

Table 1 Resource contention parameter of 802.11e EDCA within a QSTA

Priority	AC	Designation	AIFS	CW _{min}	CW _{max}	TXOPlimit
3	AC_VO	Voice	2	7	15	0.003008
2	AC_VI	Video	2	15	31	0.006016
1	AC_BE	Best effort	3	31	1023	0
0	AC_BK	Background	7	31	1023	0

packets. According to the queue length of AC[2], denoted as QL_2 , AM can probabilistically determine whether the prioritized video packets should enter AC [2] or other lower-priority ACs, namely AC[1] or AC[0]. In terms of queuing discipline, AM can be viewed as a variant of WRED (Weighted Random Early Detection) [25], which is AQM based and widely deployed in broadband backbone core routers to realize differentiated services by using different low queue thresholds, denoted as $threshold_{low}$, within a single physical queue, where its total physical queue length is considered to determine the differentiated packet *drop-probabilities*. Likewise, to support service differentiation for prioritized video packets in the AC[2] of 802.11e, AM adopts the total physical queue length of AC[2] to determine the differentiated *downward-mapping-probabilities* of video packets; however, unlike WRED, AM uses the same pair of $(threshold_{low}, threshold_{high})$ for all prioritized video packets but different mark probabilities, i.e. probabilities when QL_2 is approaching $threshold_{high}$ from the low side, or denoted as $MaxProb_{I/P/B}$, to determine the differentiated *downward-mapping-probabilities* to AC[1] or AC[0]. In other words, each prioritized video packet type follows a specific RED function of QL_2 and the corresponding $MaxProb_{I/P/B}$ in AM.

As shown in Figure 4, for each RED function in AM, there are three *downward-mapping* or *downward-transition* periods: (1) Zero Transition ($QL_2 \leq threshold_{low}$), (2) Linearly Proportional Transition ($threshold_{low} < QL_2 < threshold_{high}$), and (3) Full Transition ($QL_2 \geq threshold_{high}$). The highest-priority video packet is the most preferred one to enter AC[2], and the lowest-priority one has the highest chance to be downward mapped into other lower-priority ACs. In [14], MPEG-4 video source files were analyzed, and video packets were classified according to their I/P/B types, with priorities from high to low for simplicity. The authors of [14] also applied the same experimental setup for SM and EDCA, and their results showed that AM outperforms both SM and EDCA.

The authors of [15] proposed an *enhanced version* of AM (EAM) in the H.264 coding scheme, where the same AQM was applied to AC[2] with a different set of low and high thresholds for the queue length of AC[2], and a more subtle consideration was given to those downward-transitioned video packets into AC[1] or AC[0]. Instead of a fixed probability for going to AC[1] or AC[0], EAM further considered the traffic condition of AC[1] to determine the destiny of those downward-transitioned video packets, and the underline principle is to allow these video packets make the best use of AC[1] unless it is too

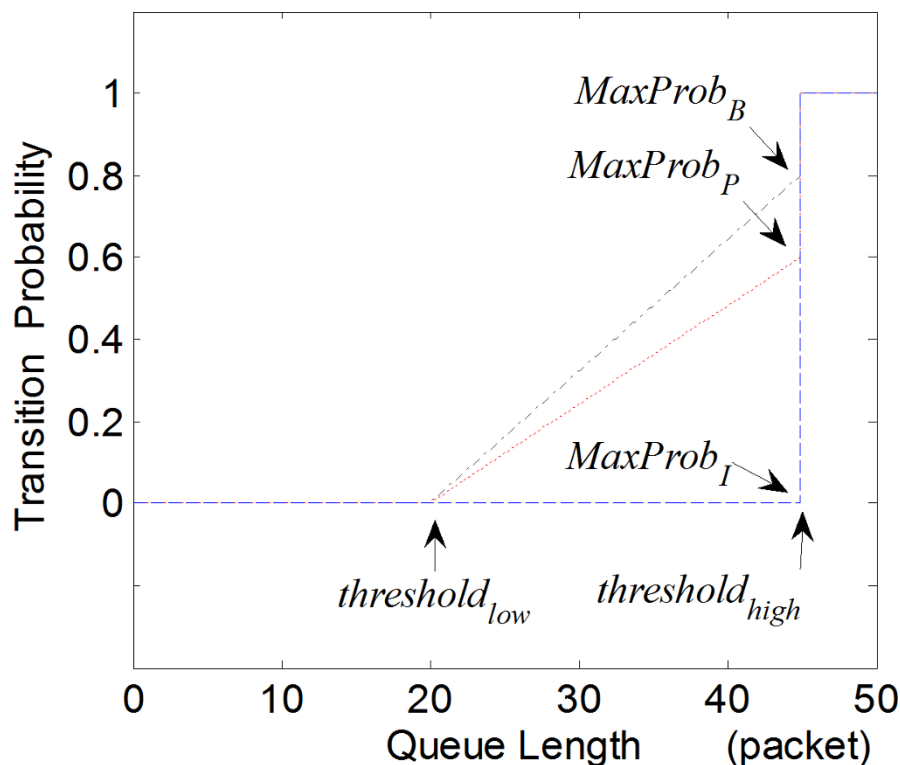


Figure 4 Linearly proportional downward transition probability function of AC[2] queue length (AM [14] and EAM [15]).

congested, since AC[1] has a better resource for channel access than AC[0]. Unfortunately, they only showed that the performance of EAM was better than EDCA. Neither various traffic effects were studied, nor a direct performance comparison between EAM and AM was given.

Mai et al. [26] proposed another extreme idea that goes back to the original case of EDCA where all the ACs are FIFO based, namely they did not rely on finding a good AQM to enhance the video transmission quality. Instead, when a video packet arrives at MAC, any AC queue which has the shortest estimated delay time among all the ACs will be selected to serve the video packet. However, such a selection among all the ACs involves complicated queue delay time calculations for all the ACs, since they always mutually affect each other.

3. Proposed cross-layer design

In terms of a better cross-layer design for further video transmission quality enhancement than the existing works, there are two major relevant issues for this study to address: (1) how to decide the video frame importance level more precisely based on the temporal coding dependence from the application layer so that the priority of the video frame associated packets can thus be determined, and (2) how to find a better AQM in the MAC layer that can dynamically and efficiently keep the AC[2] congestion level low by downward mapping those relatively unimportant video packets to other lower-priority ACs while still taking good care of these packets by choosing a better destiny from the lower-priority ACs.

The major contribution of this study lies in that we proposed a novel cross-layer design called DVFI+CQM to further enhance the transmission quality of video streaming over 802.11e. Figure 5 illustrates the proposed DVFI+CQM framework, where DVFI is the proposed scheme for precisely indexing the video frame importance from the Application layer, while the proposed CQM is a corresponding video packet mapping algorithm among the ACs in the MAC layer based on the DVFI grouping so as to achieve a higher video transmission quality than the existing works. Both DVFI and CQM have their own design concepts summarized in the corresponding box areas in Figure 5, more details of which can be found in the following sub-sections.

3.1. Dispersive video frame importance

The concept of DVFI is based on the *PSNR* degradation due to the transmission loss of a single video frame. In general, it contains two effects: (1) imperfect error concealment (*IEC*) on the lost video frame itself; and (2) error propagation (*EP*) to those successfully received surrounding video frames which will be decoded by referencing the lost video frame. The principle for determining the importance level of a video frame is that the more *PSNR* is

degraded due to a lost video frame, the more significant the lost video frame is.

The video-frame-based *PSNR* is defined in (1), where a frame size of $M \times N$ pixels is assumed, and $(f_{i,j} - f'_{i,j})$ stands for the luminosity difference per pixel due to the video quality degradation. Note that the most degraded part of *PSNR* usually occurs in the luminosity, and this is also adopted throughout the rest of the article.

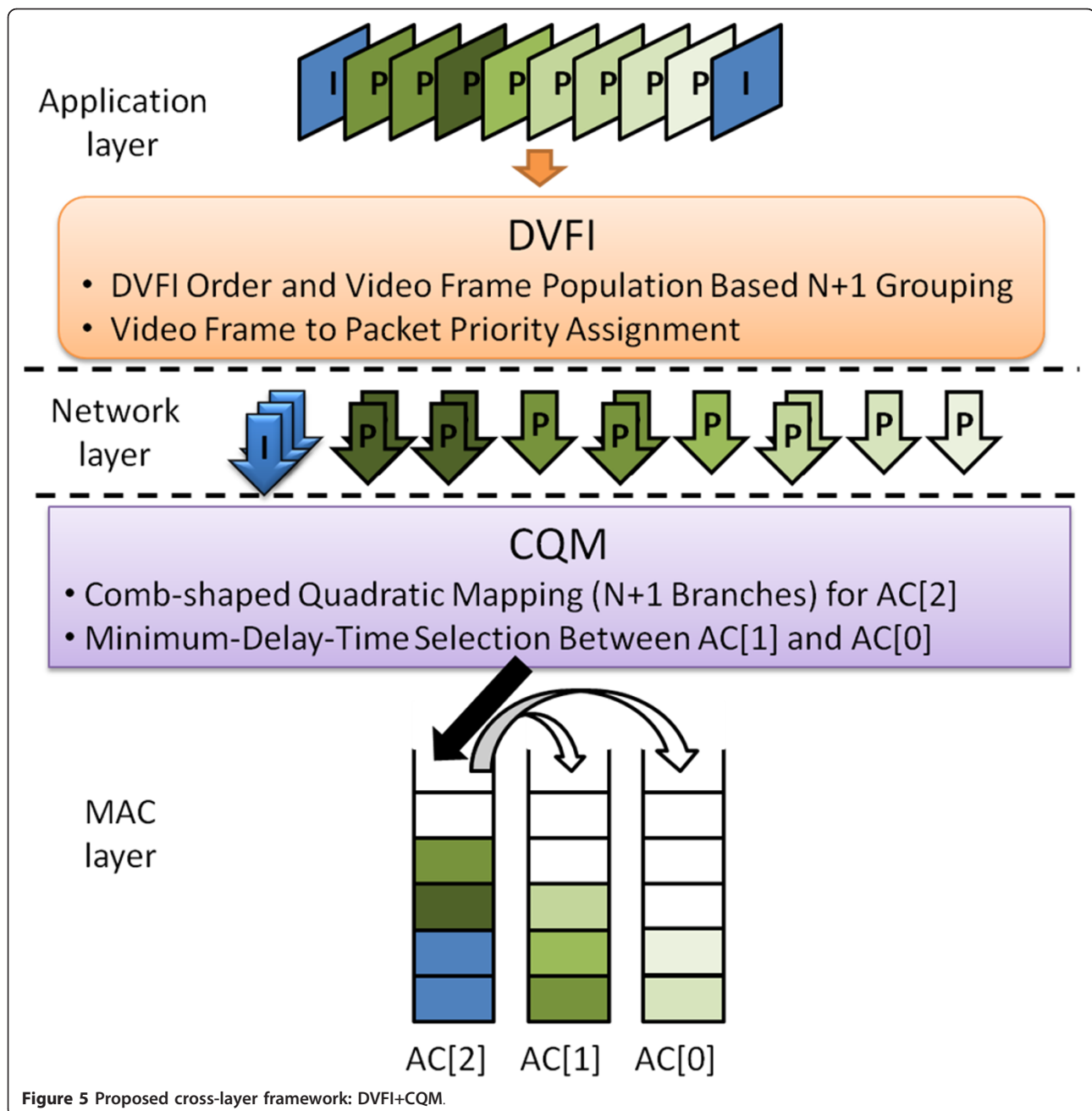
$$PSNR(\text{dB}) = 10 \log_{10} \frac{255^2 MN}{\sum_{i=1}^M \sum_{j=1}^N (f_{i,j} - f'_{i,j})^2} \quad (1)$$

3.1.1. Definition of DVFI

The DVFI value of a lost video frame is defined by Equation (2). $PSNR_{i,ed}$ is the *PSNR* of video frame i purely due to the coding loss, i.e., quality impairment after being *encoded* and *decoded*, while $PSNR_{i,etd}(n)$ is the combined *PSNR* of video frame i due to both the *coding* and *transmission* losses of video frame n . Note that $PSNR_{i,etd}(n)$ stands for the *IEC* effect when $i = n$, and the *EP* effect when $i \neq n$ and frame n is the reference frame of frame i for decoding. The numerator of (2), i.e., $\sum_{i=1}^N (PSNR_{i,ed} - PSNR_{i,etd}(n))$ can be reasoned in a simple example given by Figure 6a-d, where a GOP pattern {IPPPPPPPP} (GOP = 9) was adopted for video coding, and the transmission losses of frames 10 (I), 11 (P), and 12 (P) were considered individually. The black solid lines in Figure 6-d stands for the frame-by-frame *PSNR* curve due to the coding loss of each individual video frame i , whereas the colored dashed/dashed-dotted/dotted lines represent the frame-by-frame *PSNR* curves due to both the coding loss of each individual frame i and transmission losses of frames 10 (I), 11 (P), and 12 (P), respectively. As shown in Figure 6d, the pure transmission loss of a lost video frame can thus be obtained by taking the difference between the black solid line and the corresponding colored lines, and summing up all the frame-by-frame differences gives the numerator $\sum_{i=1}^N (PSNR_{i,ed} - PSNR_{i,etd}(n))$. Assuming that the I frame can effectively stop the error propagation over one GOP size, i.e. S_{GOP} , the numerator $\sum_{i=1}^N (PSNR_{i,ed} - PSNR_{i,etd}(n))$ can then be simplified as $\sum_{i=1}^{S_{GOP}} (PSNR_{i,ed} - PSNR_{i,etd}(n))$. Now, the remaining part in (2) is the denominator $\Delta PSNR_{max} \cdot S_{GOP}$, which serves as a normalization factor to keep the DVFI value less than 1.

$$DVFI_n = \frac{\sum_{i=1}^N PSNR_{i,ed} - \sum_{i=1}^N PSNR_{i,etd}(n)}{\Delta PSNR_{max} \cdot S_{GOP}} \quad (2)$$

$$= \frac{\sum_{i=1}^{S_{GOP}} (PSNR_{i,ed} - PSNR_{i,etd}(n))}{\Delta PSNR_{max} \cdot S_{GOP}}$$

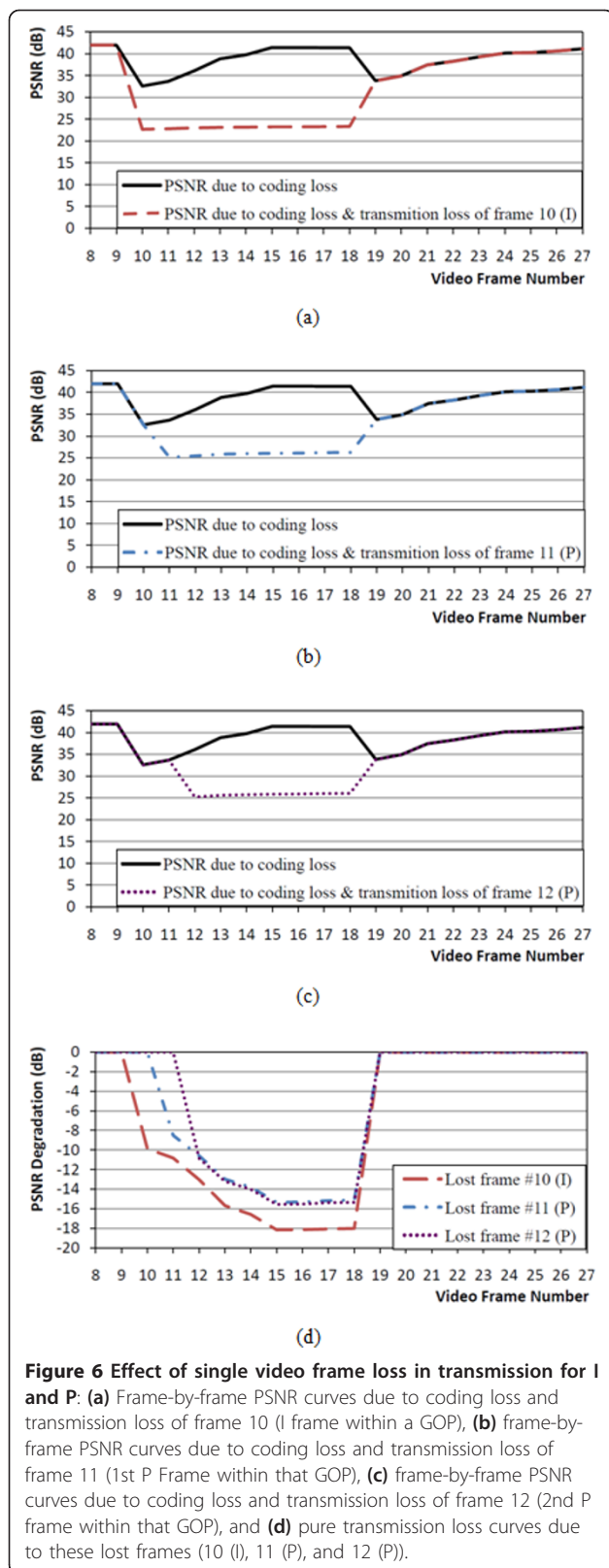


$$\text{where } \Delta PSNR_{max} = \max_{i=1}^N \{PSNR_{i,ed} - PSNR_{i,etd}(n)\}$$

3.1.2. Video frame population based grouping of DVFI

In this study, to match with the multi-branched structure of the CQM algorithm in the MAC layer, say N branches, the video frames in a test video sequence can be segmented into $N + 1$ groups based on both the absolute order of their DVFI values and the population of video frames, i.e., N prioritized groups of P frames and 1 top-priority group of I frames. Figure 7 demonstrates the case of $N = 5$ (5 groups of P frames and 1 group of I frames) where the P frames of each DVFI

group is equally populated, and the I frames form the top-priority group. Among P frames, DVFI group-5 is the most important one following the downward transition probability function of branch-5 (the rightmost branch) in the CQM packet mapping algorithm as seen in Figure 8, while DVFI group-1 is the least important one following the branch-1 (the leftmost branch) probability function. For simplicity, we assumed in this study that the packets associated with a specific video frame share the same importance level, i.e., the DVFI value of that video frame. This assumption may not be perfect



for a precise video packet importance, but is at least a much better one than the existing works, and good enough for our proposed cross-layer design to win.

3.2. Comb-shaped quadratic mapping

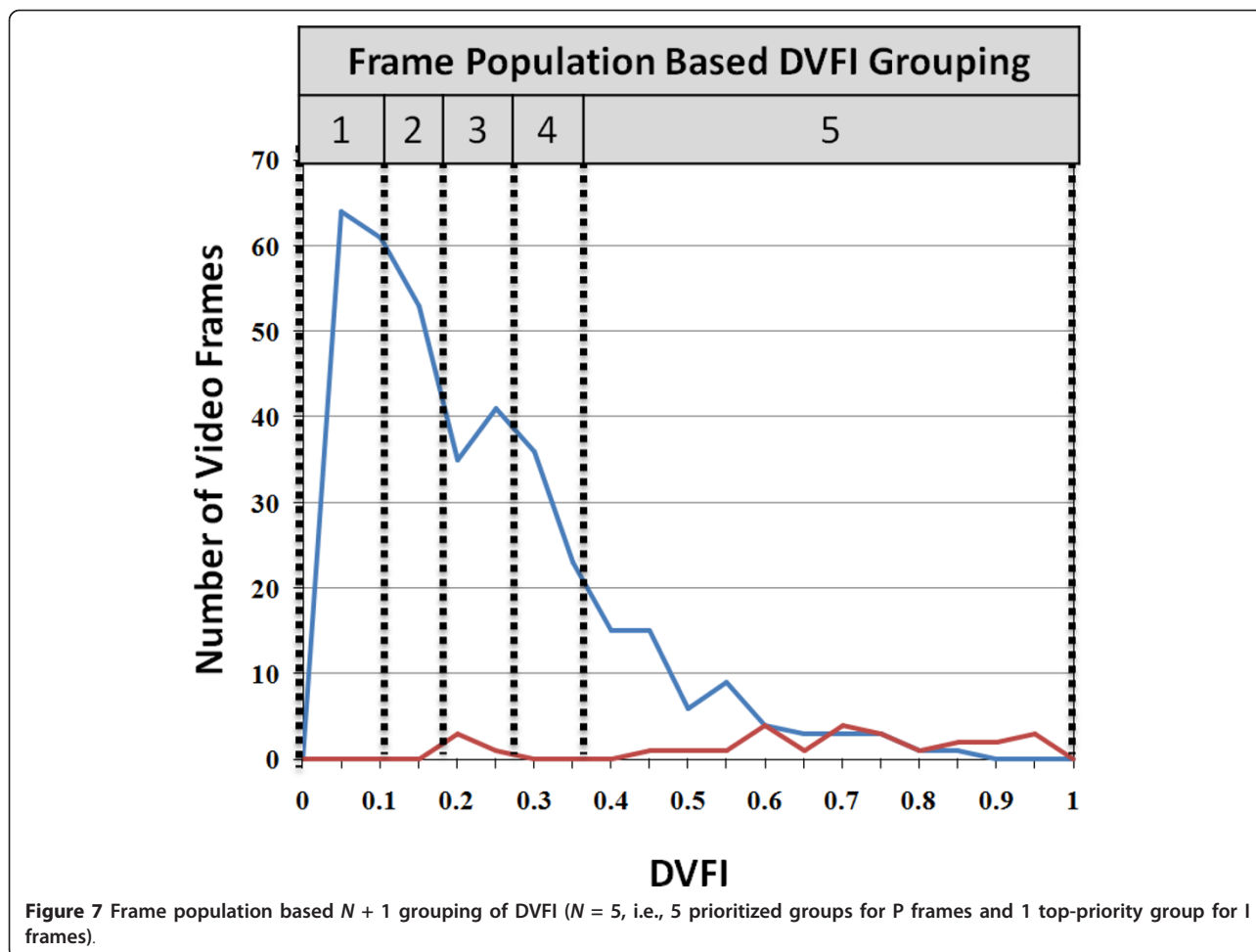
The objective of the proposed CQM algorithm is to find a good AQM-based video packet mapping algorithm among ACs in the 802.11e MAC layer that can make the best use of the DVFI scheme for video packet importance so as to achieve further video transmission quality enhancement over the existing works. The CQM achieves this objective via the following two principles.

- *Principle 1.* Keep the most important video packets in AC[2] as many as possible to make the best use of the AC[2] resources;
- *Principle 2.* Find a better destiny between AC[1] and AC[0] for those downward mapped video packets depending on which one has a shorter delay time by considering their instant queue length and the corresponding resource.

Note that AC[3] was assumed to be reserved for voice traffic in both the existing works, i.e., AM and EAM, and we follow this convention to make a fair comparison. Besides, *Principle 1* makes more sense for video streaming over 802.11e, and need to be modified for a two-way interactive video application where a maximum delay time for AC[2] should be imposed. However, this consideration is beyond the scope of this study.

In our proposed CQM algorithm, *Principle 1* can be achieved by forming multi-branched and comb-shaped downward transition probability functions for video packets based on the queue length of AC[2], denoted as QL_2 , as shown in Figure 8. Like AM and EAM, each branch (say branch i) of probability function in CQM is also characteristic of one pair of *low* (L) and *high* (H) thresholds, $(th_L[i], th_H[i])$, which separates the downward transition for DVFI group i into three phases: (1) Zero Transition ($QL_2 \leq th_L[i]$), (2) Quadratically Proportional Transition ($th_L[i] < QL_2 < th_H[i]$), and (3) Full Transition ($QL_2 \geq th_H[i]$). However, the CQM is unique and novel based on the following:

- $MaxProb_{packet_type}$ is no longer needed in CQM, which also smoothes the swapping between phases 2 and 3, instead of a discontinuous jump.
- Instead of *linearly proportional*, a *quadratically proportional* function is adopted for phase 2 to emphasize the objective of keeping the most important video packets in AC[2] as many as possible.
- To downward map the least important ones into AC [1] or AC[0] as early as possible, a flexible multi-branched structure is allowed for CQM to co-work with



the intended number of DVFI grouping, where I frames are reserved as the top priority group and follow a step-function-based transition, and P frames follow a corresponding three-phases-transition-branch based on their DVFI grouping.

More details of the CQM algorithm, including *Principles 1* and *2*, can be found from the pseudo code of CQM, as given in Figure 9. This pseudo code consists of a step function for I frames which merges $threshold_{high}$ and $threshold_{low}$ together at $maximum_AC[2]_{buffer_size} - 1$ (in units of packet) and a comb-shaped structure of five-branch functions for P frames. This can be clearly seen if the input argument $packet_type$ in the function call to $DVFI_CQM(\dots)$ is carefully understood. The function call to $Min_Delay()$ implements the idea of *Principle 2*, where the minimum delays of $AC[1]$ and $AC[0]$ can be estimated based on the following concept: the delay time of $AC[i]$ is proportional to a ratio R , as defined in (3).

$$R = \frac{QL(AC[i])}{(AIFS(AC[i]) + \frac{CWmin(AC[i])}{2})} \quad (3)$$

where $QL(AC[i])$ is the queue length of $AC[i]$ and it is assumed that no retry is allowed for $AC[i]$ if unsuccessful channel access occurs due to collision. Thus, the item $\frac{CWmin(AC[i])}{2}$ reflects the average delay time

contributed from the random backoff timer chosen from $[0, CW_{min}(AC[i])]$. Other studies such as the shape of the comb structure and the branch multiplicity, in terms of symmetric and asymmetric combs as well as five and ten branches, can be found in Section 4.

4. Results

To demonstrate the validity and superiority of our proposed cross-layer design, an experimental environment of video delivery over an 802.11e ad hoc network was setup with a description of details in Section 4.1, including a special consideration and explanation why various video-input-rate and cross-traffic test conditions are needed. In Section 4.2, the DVFI distributions of Foreman CIF & QCIF under various test conditions have been obtained and their mean values in terms of both I

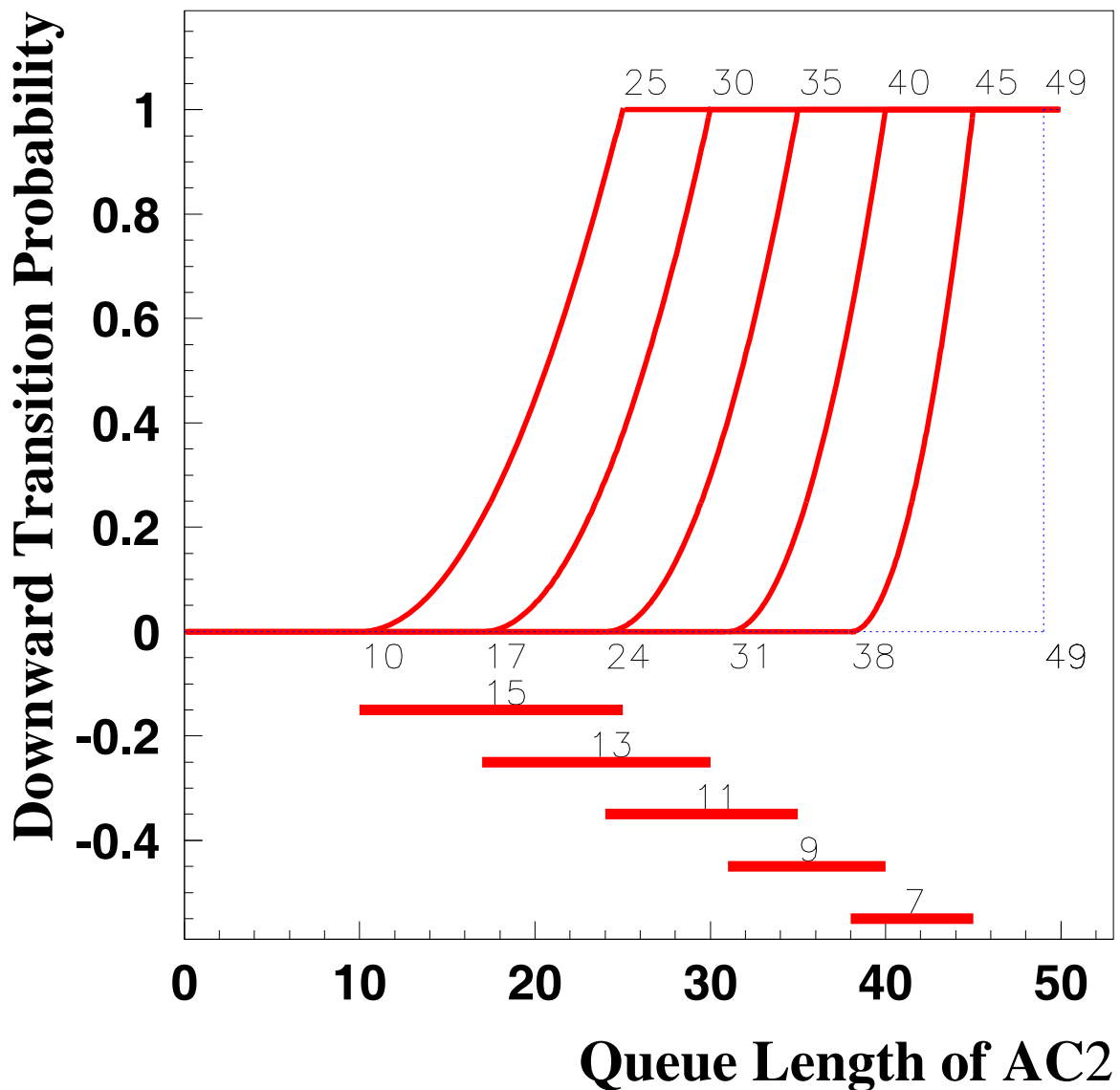


Figure 8 Five-branched quadratically proportional downward transition probability functions for P packets (red solid lines) and step function for I packets (blue dotted line) versus AC[2] queue length. (The proposed CQM packet mapping algorithm.)

and P frames are illustrated in Figures 10 and 11. Section 4.3 presents the threshold effects of symmetric and asymmetric comb shapes of the CQM algorithm. Besides, the instant queuing effects of AC[2] and the average congestion levels of all the ACs are shown and discussed in Section 4.4. Furthermore, Section 4.5 presents detailed PSNR-based performance comparisons of the proposed cross-layer design with the existing works using several representative video sequences under various test conditions, together with a subtle discussion

on how the transmission quality is affected by a novel performance metric called Weighted Performance Ratio of DVFI (WPR) that we propose in this study to show the correlation between PSNR and the relative distributions and mean values of lost I and P frames.

4.1. Experimental setup

The following simulation setup was established in this study. Eight YUV reference video sequences [27] were adopted for testing, with half of them in the QCIF (176

```

/* Assign the High (H) and Low (L) threshold values of the five-branches comb for the five DVFI groups of P packets.*/
thH[1,2,3,4,5] = {25, 30, 35, 40, 45}
thL[1,2,3,4,5] = {10, 17, 24, 31, 38}
while (a video packet arrives)
{
    if (packet_type == 'I')
    /* Merged thresholds of the Step Function for I packets. */
    thresholdhigh = maximum_AC[2]_buffer_size - 1
    thresholdlow = maximum_AC[2]_buffer_size - 1
    elseif (packet_type == 'P')
    /* high and low thresholds according to DVFI grouping. */
    switch DVFI {
        case (DVFI belongs to group_1):
            thresholdhigh = thH[1]
            thresholdlow = thL[1]
        case (DVFI belongs to group_2):
            thresholdhigh = thH[2]
            thresholdlow = thL[2]
        case (DVFI belongs to group_3):
            thresholdhigh = thH[3]
            thresholdlow = thL[3]
        case (DVFI belongs to group_4):
            thresholdhigh = thH[4]
            thresholdlow = thL[4]
        case (DVFI belongs to group_5):
            thresholdhigh = thH[5]
            thresholdlow = thL[5]
    }
    end
    DVFI_CQM(packet_type, thresholdlow, thresholdhigh)
}
void DVFI_CQM(packet_type, thresholdlow, thresholdhigh) {
    if (length(AC[2]) < thresholdlow)
        video packet → AC[2]
    elseif (length(AC[2]) < thresholdhigh)
        if (packet_type = 'I')
            video packet → AC[2]
        else
            Probtransition = { (length(AC[2]) - thresholdlow) / (thresholdhigh - thresholdlow) }2
            rn = uniform(0,1)
            if (rn > probtransition)
                video packet → AC[2]
            else
                Min_Delay()
            end
        end
    else
        Min_Delay()
    end
}
void Min_Delay() {
    if (AC[0] has a shorter queue delay than AC[1])
        video packet → AC[0]
    else
        video packet → AC[1]
    end
}
}

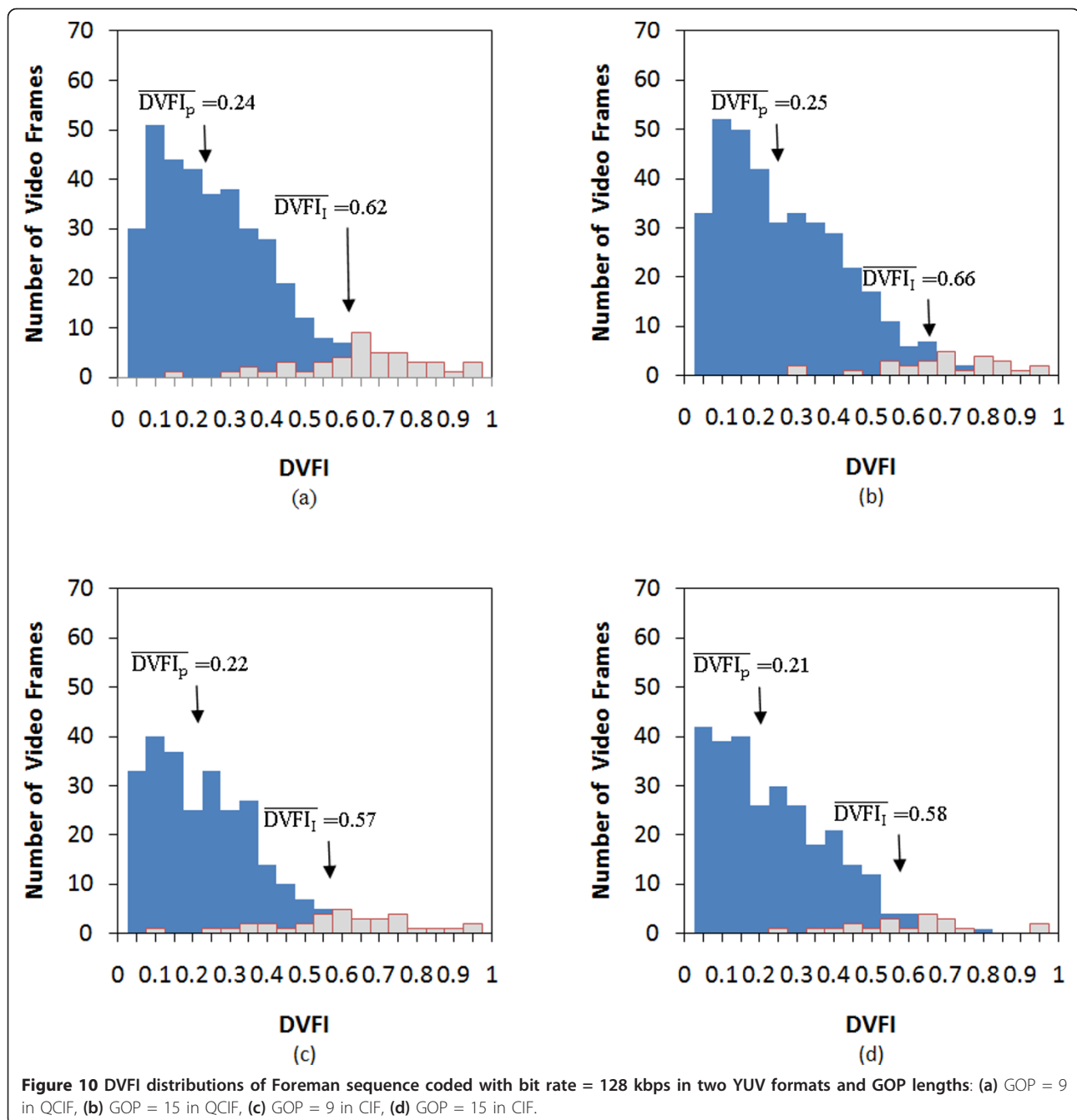
```

Figure 9 Pseudo code of the proposed DVFI+CQM packet mapping algorithm in the MAC layer.

× 144) format and the other half in the CIF (352 × 288) format. Table 2 lists these eight video sequences and marks their levels of motion to explain why they can be representative, where the numbers of video frames of

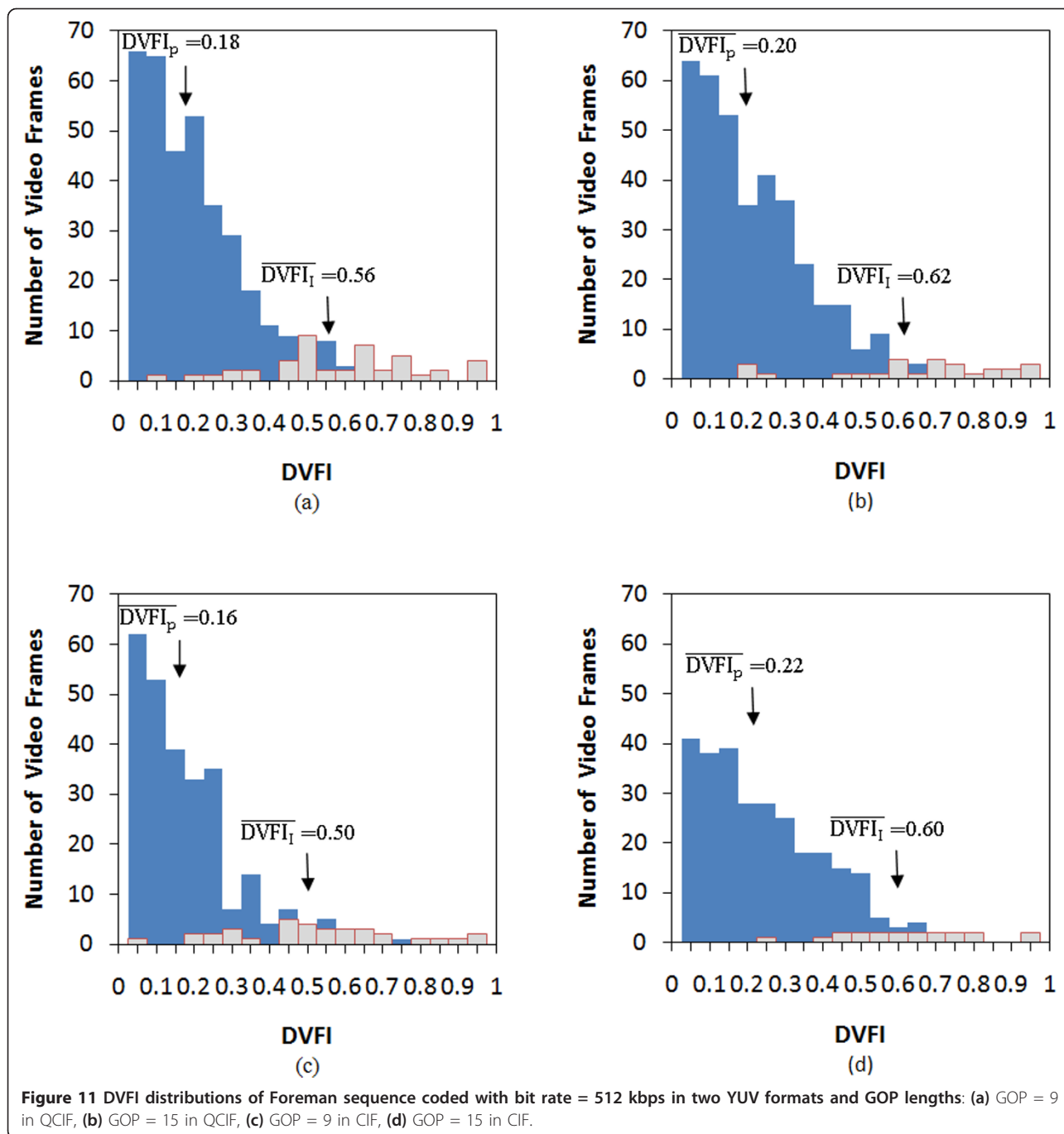
these video sequences are also summarized, including *total*, *I*, and *P* frames.

These video sequences are encoded into H.264 (JM10.2) bit streams based on a GOP structure of



{IPPP...} with periodic lengths of 9 or 15 video frames (denoted as GOP = 9 or GOP = 15) and a frame rate at 30 Hz, and packetized with the maximum transmission packet size of 1000 bytes over an 802.11e ad hoc network, as shown by the topology in Figure 12. To simplify the error propagation effect and confine it within a GOP, the number of reference frames N_{ref} is adopted to be 1. Note that N_{ref} is a new feature of H.264, but it is in general quite time-consuming and does not help much in coding efficiency unless for periodic motions.

Furthermore, the error propagation could become more complicated if $N_{ref} > 1$, where error propagation might go beyond one GOP size. The EvalVid multimedia framework [28] integrated with the ns-2 network simulator [29] was used to provide the H.264 video streaming [30] over an ad hoc wireless network simulation environment, where the DSDV routing protocol was used and the adopted data rate of the wireless link was 1 Mbps. Four ad hoc nodes were setup, where one served as a video server and another as a video client with cross



traffic flows established among both the connections (Node 1 → Node 2) and (Node 3 → Node 4). Table 3 defines six congestion cases of cross traffic, where Case n ($n = 1, 2, 3, 4, 5,$ or 6) stands for, on each connection, n flows of cross traffic in voice (64 kbps per flow), UDP (10 kbps per flow), and TCP (not characteristic of constant-bit-rate, but featured by self-congestion-control in variable-bit-rate) are, respectively, established through AC[3], AC[1], and AC[0] of the traffic sender nodes.

4.1.1. Congestion levels of ACs

It is not trivial at all to understand the meaning of congestion level of each AC queue. Note that the congestion levels of all the ACs within a single wireless station are mutually affected since they are contending the same wireless channel, as previously explained in Section 2.2. In other words, the congestion levels of these ACs can only be better understood if both the input rate (denoted as IR , i.e., the coded bit rate) of video

Table 2 Numbers of frames and packets in the test video sequences

YUV seq. format	Number of video frames			Motion
	Total	I	P	
Foreman QCIF	400	45	355	Fast
Foreman CIF	400	45	355	Fast
Carphone QCIF	382	43	339	Medium fast
Carphone CIF	382	43	339	Medium fast
News QCIF	300	34	266	Medium slow
News CIF	300	34	266	Medium slow
Mother-&Daughter CIF	300	34	266	Slow
Mother-&Daughter QCIF	300	34	266	Slow

source into AC[2] and the number of cross traffic flows in the other ACs are considered concurrently.

For instance, when considering a fixed IR in EDCA, the congestion level of AC[2] should increase with n . Recall that the AC[2] in EDCA are FIFO based, not AQM based. There should be a similar increasing trend for the AQM-based mapping algorithms for AC[2], including the existing works and the proposed CQM algorithm, and the differences are that only the AQM-based algorithms will perform their own mechanism to downward map those less significant video packets to lower-priority ACs, which could again increase the congestion level of the destined lower-priority AC. On the other hand, for a fixed n , the congestion level of AC[2] could be dominated by IR , i.e., the larger the IR , the more congested the AC [2].

Table 3 Definition of six congestion cases of cross traffic, $n = 1, 2, 3, 4, 5, 6$

Connections of nodes	Number of traffic flows			
	VoIP traffic (AC[3])	Video traffic (*)	UDP traffic (AC[1])	TCP traffic (AC[0])
1 → 2	n	1	n	n
3 → 4	n	0	n	n

*AC[2] with different mapping algorithms, including EDCA, I/P+AM[10/45], I/P+AM[20/45], I/P+EAM[10/45], I/P+EAM[25/40], and the proposed DVFI+CQM.

To conduct a fair comparison under such a complicated combination of the video source's input rate and the number of cross traffic flows, four input rates ($IR = 128, 256, 384, \text{ and } 512 \text{ kbps}$) were considered together with different number of congestion cases due to cross traffic. To be more specific, we consider two more cases of n (i.e., $n = 7$ and 8) when $IR = 128$ or 256 kbps . The reasons for that have been explained previously and become obvious now: (1) the fixed n for different IR cases could stand for different congestion levels of AC [2], also different congestion levels of other ACs; (2) in smaller IR cases such as 128 or 256 kbps , $n = 6$ may just stand for a medium congestion level of AC[2] while the same n could mean a high congestion level of AC[2] in larger IR cases such as 384 or 512 kbps .

Note that such combined IR and n test cases will be adopted throughout the rest of the article. Now, it should also become clear why Table 4 summarizes the numbers of *total/IIP* packets of all the test video

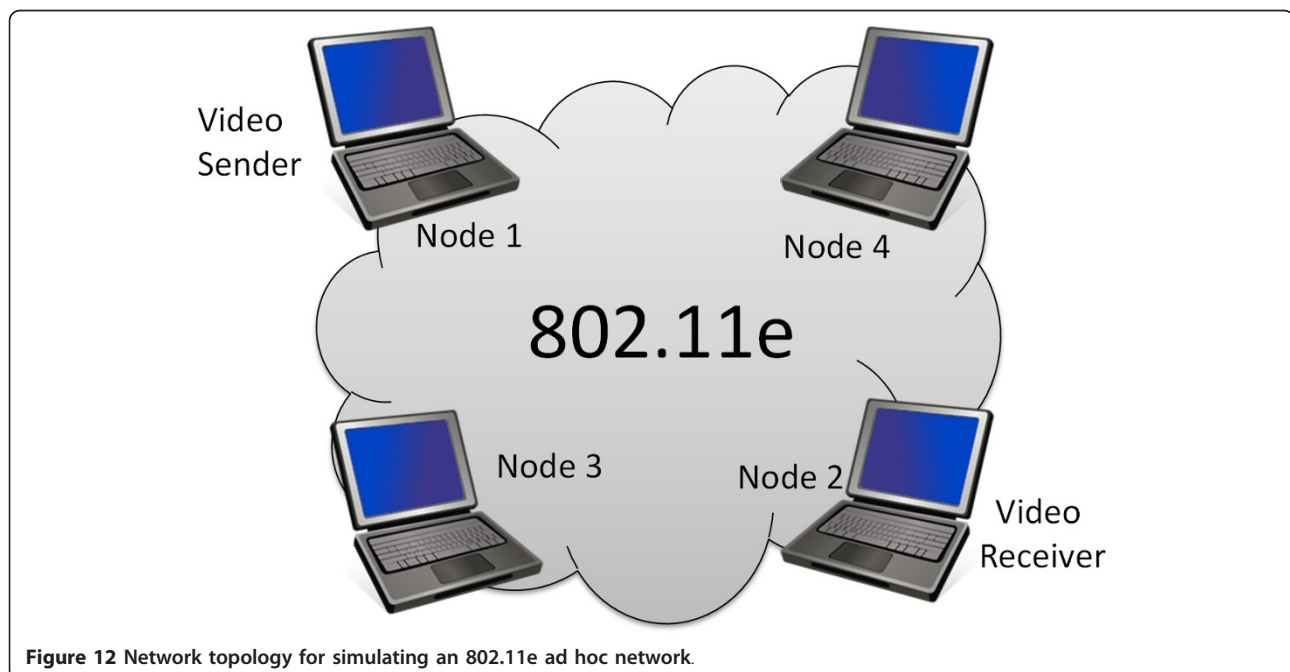


Figure 12 Network topology for simulating an 802.11e ad hoc network.

Table 4 Number of packets of video sequences for various YUV formats, GOP sizes, and coded bit rates (i.e., input rates to AC[2])

YUV seq./ format	GOP size (**)	Number of packets (kbps)											
		128			256			384			512		
		T	I	P	T	I	P	T	I	P	T	I	P
Foreman/CIF	9	361	95	266	447	177	270	637	245	392	794	301	493
Foreman/CIF	15	349	69	280	415	120	295	645	158	487	779	188	591
Foreman/QCIF	9	486	131	355	574	200	374	867	251	616	1033	289	744
Foreman/QCIF	15	457	84	373	538	124	414	883	147	736	1003	172	831
Carphone/CIF	9	456	117	339	545	195	350	830	253	577	986	300	686
Carphone/QCIF	9	450	111	339	520	165	355	856	211	645	965	244	721
News/CIF	9	384	118	266	481	214	267	589	284	305	808	349	459
News/QCIF	9	379	113	266	450	171	279	611	221	390	796	264	532
M&D*/CIF	9	379	113	266	468	190	278	603	244	359	810	301	509
M&D*/QCIF	9	370	104	266	452	159	293	620	195	425	804	226	578

* M&D = Mother-&-Daughter.

** The GOP pattern is {IPPP...} with a period of 9 video frames.

sequences in terms of these four input rates of video source, as aforementioned.

4.1.2. Comb's shape and threshold settings

It is imaginable that the threshold settings, including *high* and *low*, could affect the performances of all the AQM-based packet mapping algorithms. Besides, it also affects the comb's shape in the proposed CQM algorithm. To address and simplify these issues, we set up four case studies:

- SC-5B (Symmetric Comb of five *parallel* Branches)
- AC-5B (Asymmetric Comb of five *unparallel* Branches)
- SC-10B (Symmetric Comb of ten *parallel* Branches)
- AC-10B (Asymmetric Comb of ten *unparallel* Branches)

where their multi-branched ($threshold_{low}$, $threshold_{high}$) pairs, i.e., $\{(th_L[i], th_H[i]), i = 1-5 \text{ for } 5B\}$ and $\{(th_L[i], th_H[i]), i = 1-10 \text{ for } 10B\}$, are defined in Table 5.

Table 5 Low and high threshold settings of the proposed CQM algorithm with various comb shapes and branch multiplicities

	AC-5B					SC-5B				
$i =$	1	2	3	4	5	1	2	3	4	5
$th_H[i]$	25	30	35	40	45	17	24	31	38	45
$th_L[i]$	10	17	24	31	38	10	17	24	31	38
	AC-10B									
$i =$	1	2	3	4	5	6	7	8	9	10
$th_H[i]$	25	27	30	32	34	36	39	41	43	45
$th_L[i]$	10	13	17	20	24	27	31	34	38	41
	SC-10B									
$i =$	1	2	3	4	5	6	7	8	9	10
$th_H[i]$	13	17	20	24	27	31	34	38	41	45
$th_L[i]$	10	13	17	20	24	27	31	34	38	41

The idea for these is to test both the effects of *low-and-high-threshold* scenarios and the branch multiplicity of the comb shape in the proposed CQM algorithm. For the existing works AM [14] and EAM [15], [20, 45] and [25, 40] were the adopted *low* and *high threshold* pairs. To make a more fair comparison, we have added [10, 45] for both of them in this study. Considering the various cross-layer frameworks with different packet importance schemes, we rename them in the following:

- DVFI+CQM[10/45]

The video packet's priority scheme is based on DVFI, and the AC-5B case is assumed here for CQM, as shown later it makes no big difference with the other three cases. [10/45] stands for that five pairs of low-and-high thresholds are located in the range from 10 to 45.

- I/P+AM[20/45]

For AM, the video packet's priority scheme is only based on two importance levels, i.e., I or P, and $[threshold_{low}, threshold_{high}] = [20, 45]$.

- I/P+AM[10/45]

Ditto except that $[threshold_{low}, threshold_{high}] = [10, 45]$.

- I/P+EAM[25/40]

For EAM, the video packet's priority scheme is only based on two importance levels, i.e., I or P, and $[threshold_{low}, threshold_{high}] = [25, 40]$.

- I/P+EAM[10/45]

Ditto except that $[threshold_{low}, threshold_{high}] = [10, 45]$.

4.2. DVFI distributions

As aforementioned in Section 3.1, the priority of a video packet is determined according to (2), i.e., the DVFI value of its associated video frame. Since this study primarily adopted the H.264 coding scheme with a periodic

{IPPPPPPP} (GOP = 9) pattern, it is interesting to see the DVFI distribution in terms of both I and P packets. The DVFI distribution certainly manifests itself as a much more precise priority scheme for video packets, compared to the two-level-priority scheme (denoted as I/P). The {IPPPPPPPPPPPPP} (GOP = 15) case is also presented here to see how the GOP length affects the DVFI distribution. Figures 10 and 11 show the DVFI distributions of Foreman sequence coded with bit rate (i.e., IR) = 128 and 512 kbps, respectively, in two YUV formats (QCIF and CIF) and GOP lengths ({IPPP...} of GOP = 9 and GOP = 15). Two observations can be clearly seen as follow.

- In both the IR cases, the GOP = 15 case generated a larger video frame population ratio of P to I than the GOP = 9 case, which is well expected.

- In both the IR cases, the DVFI mean values of I and P frames in GOP = 15 are larger than those in GOP = 9, which is also reasonable since both the error propagation effects of I and P frames in GOP = 15 should be more serious than those in GOP = 9. The only exception is the DVFI mean of P frames in the CIF case of IR = 128 kbps, where there seems to be no GOP effect.

In addition to the Foreman sequence, the other selected video sequences have also been checked, and they all generated similar distributions and observations.

Recall that the grouping issue of DVFI in Section 3.1.2, i.e., the video frames in a test video sequence can be segmented into $N + 1$ groups based on both the absolute order of their DVFI values and the equal population of video frames, i.e., N prioritized groups of P frames and 1 top-priority group of I frames. To match with the CQM packet mapping algorithm in the MAC layer, there should be accordingly $N + 1$ branches of mapping probability functions in CQM, i.e., comb-shaped functions of N branches (i.e., the comb's branching multiplicity) for P packets, and 1 step function for I packets. More details about the comb's shape and the threshold effects are given below.

4.3. Threshold effects of symmetric and asymmetric combs

The objective of understanding the performance effects due to the comb's shape and threshold settings of the proposed CQM algorithm have been well motivated in Section 4.1.2, where four case studies (SC-5B, AC-5B, SC-10B, AC-10B) have also been well defined. Figure 13 presents a combined view of the results based on these comb shapes and threshold settings under various tests of video input rates and congestion cases of cross traffic, the definitions of which can be found in Section 4.1.1. These results are based on the average PSNR of the Foreman QCIF video sequence, and they deliver the following messages.

- The performance differences among the case studies are essentially small (about 0.1-0.3 dB in general).

- No single case wins over the others in all the video input rates (IR) and congestion cases of cross traffic (n).

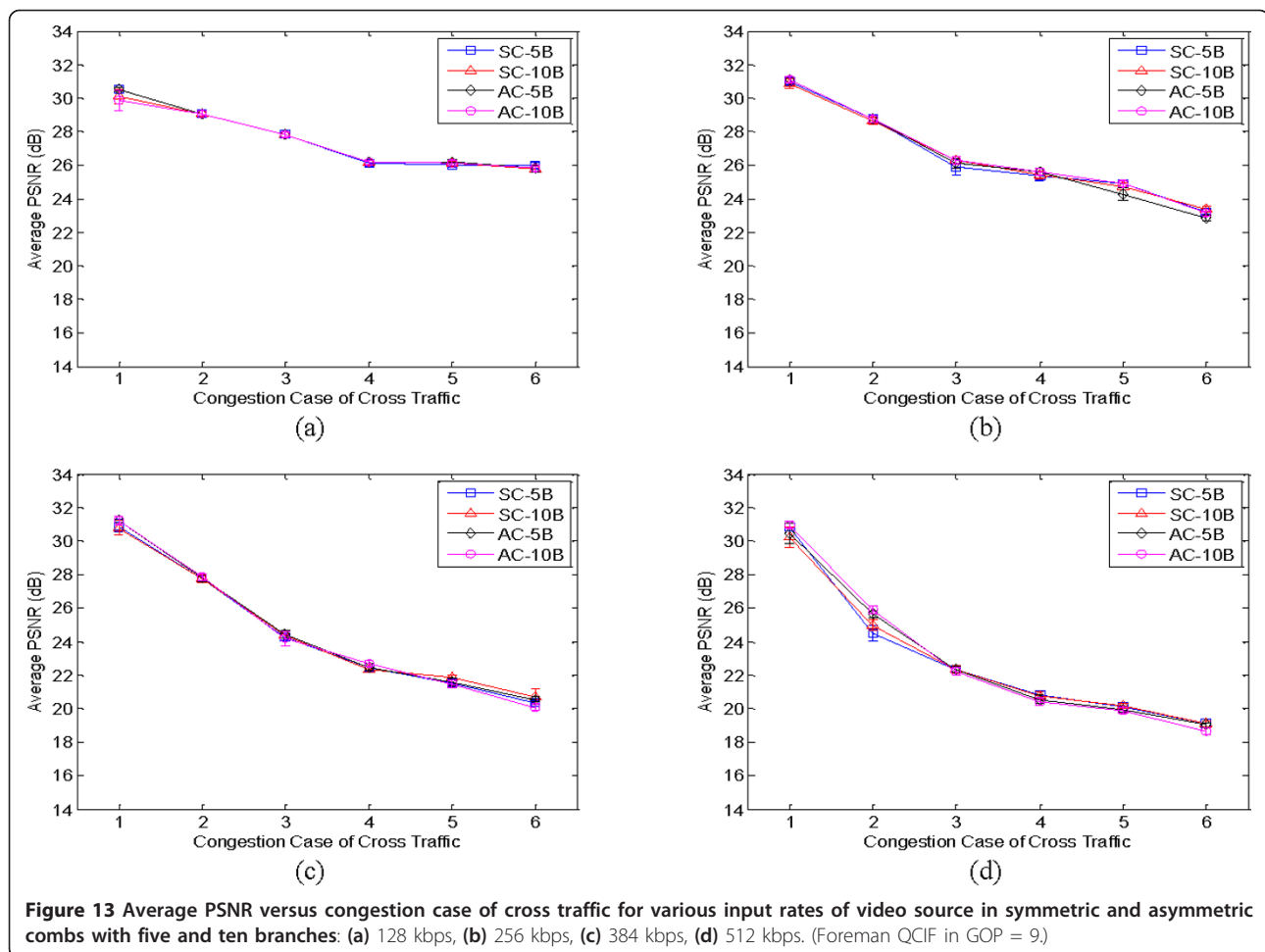
- The AC-5B case seems to be slightly preferred than the others in all the congestion levels of n when IR = 512 kbps (in particular, AC-5B wins over SC-5B and SC-10B by around 1.1 and 0.7 dB when n = 2, and AC-10B is not preferred since it is the worst one when n = 6 in both IR = 512 kbps and IR = 384 kbps), though it also seems to be slightly worse than the others in heavy congestion levels of n when IR = 256 kbps. Thus, it should be safe to take AC-5B as a typical case throughout the rest of this article.

4.4. Queuing effects of ACs

Recall from Section 3.2 that the major principles of the proposed CQM packet mapping algorithm are to keep more important video packets in AC[2] to make the best use of its resources for better channel access while taking good care of those less significant video packets by choosing a better one between AC[1] and AC[0] which has a shorter queue delay time. It is thus interesting to see how the queuing effects of all the ACs support these design principles.

To follow the discussions in the previous section, it is convenient to take the Foreman QCIF sequence as an example, considering various video input rates and congestion cases of cross traffic. Let us start with observing the queue length of AC[2], or $QL_2(t)$ for short. Figure 14 shows $QL_2(t)$ of two typical video input rates, i.e., IR = 512 kbps and IR = 128 kbps, with the same congestion case of cross traffic (n = 4). Such a combination of IR and n can reinforce the main concept of Section 4.1.1, namely the same congestion case of cross traffic (n) could mean very different congestion levels of AC [2]. As shown in Figure 14, it is clearly seen that, in the case of IR = 512 kbps, all the $QL_2(t)$ curves of different mapping algorithms are congested in different levels (from medium to high) and the congestion level of the proposed CQM algorithm is the minimum, and thus meets the goal of the design principle for AC[2], i.e., *Principle 1*, as explained in Section 3.2. Besides, it is also seen that the other AQM-based algorithms (EAM and AM in different thresholds settings) do perform better than EDCA (non-AQM-based). On the other hand, in the case of IR = 128 kbps, n = 4 still means very low congestions of AC[2] in all the mapping algorithms, and actually these $QL_2(t)$ curves merge into the same one because none of them ever exceeds the lowest value of $threshold_{low}$, namely every AQM-based mapping algorithm behaves just like EDCA.

Figure 15 gives the time-averaged queue length of all the ACs as a function of n in the same video input



rates, i.e., n ranges from 1 to 6 for $IR = 512$ kbps and from 1 to 8 for $IR = 128$ kbps, based on the same reasoning mentioned previously. For AC[2], both the IR cases again support the idea of *Principle 1*. For *Principle 2*, the Minimum-Delay-Time selection rule between AC [1] and AC[0] does show its effects in both the IR cases, as summarized below.

- Both AC[1] and AC[0] are least congested in EDCA since no downward mapping of video packets from AC [2] is allowed.
- The Minimum-Delay-Time selection rule embedded in CQM can generate more balanced congestions among AC[1] and AC[0] so that the congestion level of AC[1] in CQM seems to be the minimum among the AQM-based algorithms.

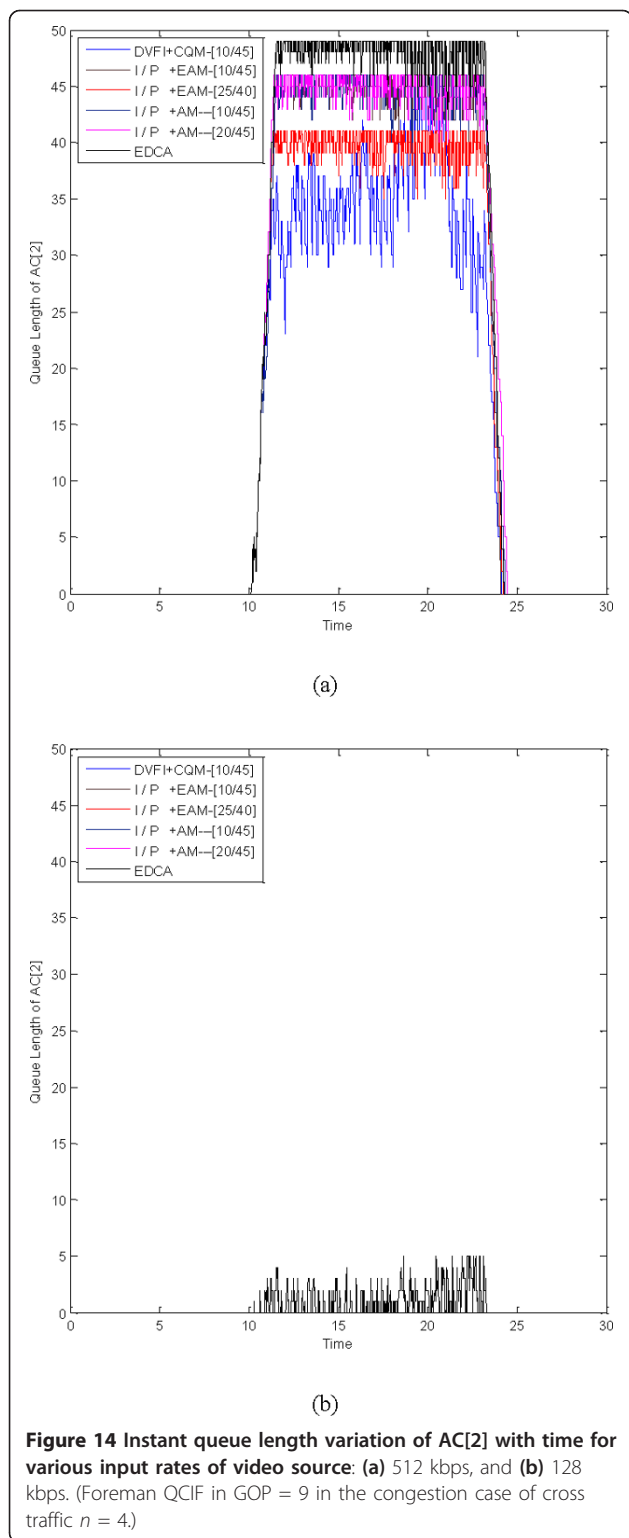
4.5. Performance evaluation

To understand more about the performance issue, PSNR-based performance evaluations can provide a much more intuitive way for comparison. To fully support the superiority of the proposed cross-layer design

(DVFI+CQM), the existing frameworks with various threshold settings (as defined in Section 4.1.2) together with EDCA are compared under the combined test conditions of IR and n (as defined in Section 4.1.1).

Firstly, to see the impact of the GOP length to the PSNR performance, let us re-run the $GOP = 9$ and $GOP = 15$ cases for both Foreman CIF and QCIF. The results are summarized in Figures 16 and 17, where the following messages can be delivered.

- In all the combined test conditions of IR and n , the proposed DVFI+CQM framework outperforms all the existing works with various threshold settings. Besides, EDCA is the worst one, as well expected.
- It is also seen from the time-averaged queue lengths of AC[2] in Figure 15 that both $n = 7$ and $n = 8$ for $IR = 128$ kbps can generate a roughly similar congestion level as both $n = 5$ and $n = 6$ for $IR = 512$ kbps. Thus, both $n = 7$ and $n = 8$ are also added for $IR = 256$ kbps, since it is closer to $IR = 128$ kbps. The general trend in these two IR cases indicates that the performances of the test frameworks start to differ only when $n > 6$ for



$IR = 128$ kbps, $n > 4$ for $IR = 256$ kbps. On the other hand, earlier differences can be seen when $n > 2$ for $IR = 384$ kbps, and $n > 1$ for $IR = 512$ kbps.

• In the case of CIF, the gains of the proposed framework over the existing ones in the case of $GOP = 9$ are somewhat larger than those in the case of $GOP = 15$. However, the gains of both the GOP cases look quite similar. These could well be reasoned by the different distributions of DVFI between these two GOP cases, as aforementioned in Section 4.2 and Figures 10 to 11.

To gain more understanding on the potential correlation of the PSNR-based performances to the DVFI distributions and mean values of lost I and P packets, we propose a novel performance metric called WPR, as defined in (4), to reason the influence by the relative distributions of I and P frames.

$$WPR = \frac{N_{received}(I \& P)}{w_I M_I + w_P M_P} \quad (4)$$

where the numerator $N_{received}(I \& P)$ is total number of received I and P packets, and the denominator $w_I M_I + w_P M_P$ is the sum of the weighted DVFI means of lost I and P packets, with M_I being the DVFI mean of lost I packets and M_P being the DVFI mean of lost P packets, both of which are weighted by w_I and w_P , i.e., the ratio of lost I packets to the total number of lost video packets, and the ratio of lost P packets to the total number of lost video packets, respectively. The basic idea of WPR is that it presents a better video transmission quality when $N_{received}(I \& P)$ is large, and $w_I M_I + w_P M_P$ is small. This idea makes a perfect sense, and it is not proposed to replace PSNR, but to provide a tool to observe how well both DVFI and CQM perform in the proposed cross-layer framework. Figure 18 shows a clear correlation between PSNR and WPR in the case of Foreman QCIF and $GOP = 9$. It is seen that WPR follows similar performance patterns as n increases for the various test frameworks in all the IR cases. We have conducted the same tests on the other video sequences, including Foreman CIF, Carphone (QCIF and CIF), News (QCIF and CIF), and Mother-&-Daughter (QCIF and CIF), and similar conclusions have been obtained. These results show a common conclusion: the design principles (recall *Principles 1 and 2*) have been well implemented in the proposed DVFI+CQM framework. Now, let us go one step back and concentrate on the PSNR-based performance evaluations for all these video sequences both in QCIF and CIF, and recall that they are chosen for test since they represent different motion levels: Foreman stands for a *fast* video, Carphone for *medium-fast*, News for *medium-slow*, and Mother-&-Daughter for *slow*. Figures 19, 20, and 21 summarize the performance evaluations for the latter three video sequences in both the QCIF and CIF formats. A consistent message from the performance evaluation of all the test video sequences can be delivered as follows.

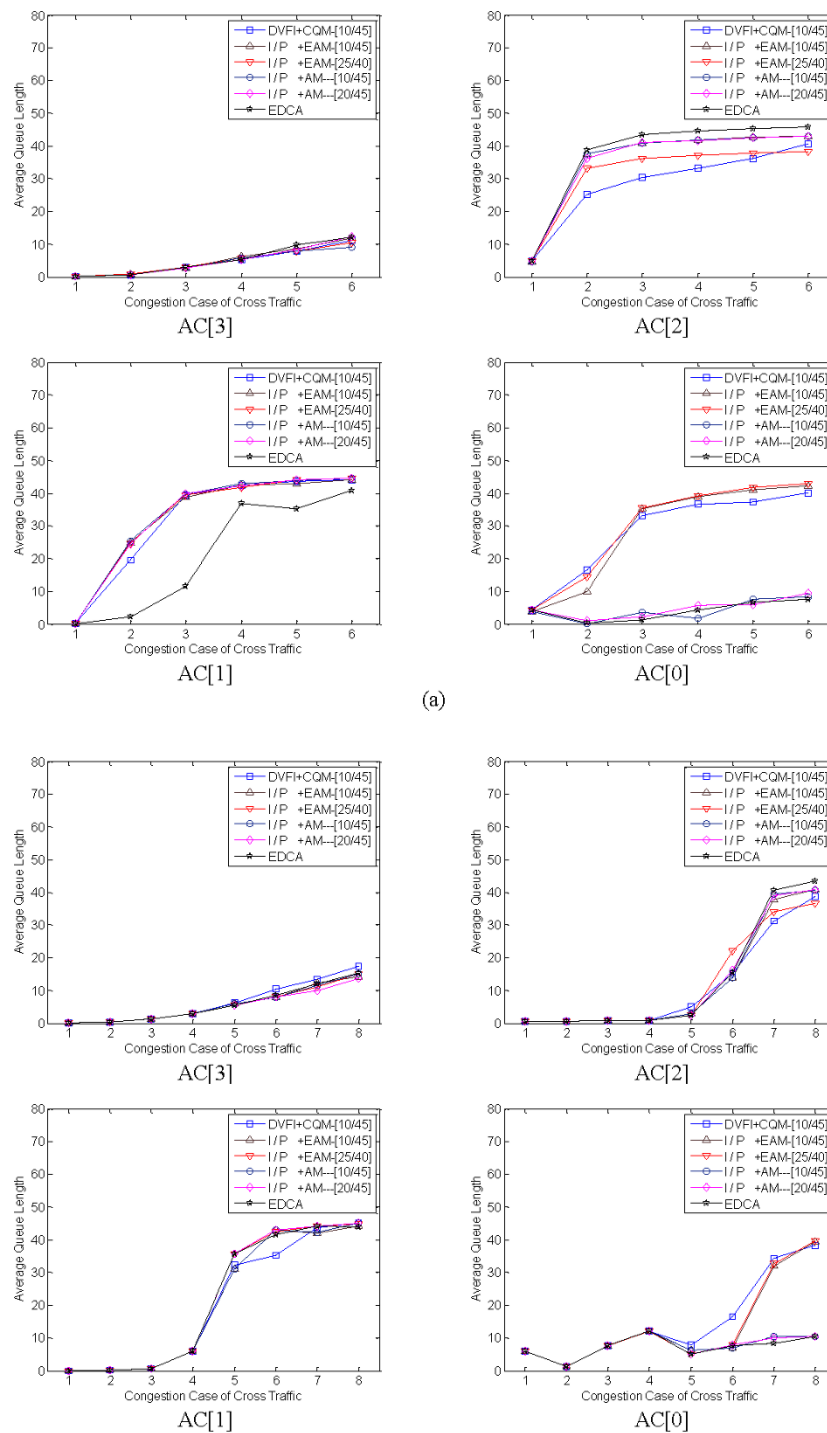


Figure 15 Time averaged queue length of ACs versus congestion case of cross traffic for various input rates of video source: (a) 512 kbps, and (b) 128 kbps. (Foreman QCIF in GOP = 9)

- In general, the proposed DVFI+CQM framework takes the lead quite visibly in all the test video sequences no matter in QCIF or CIF, except that the lead is somewhat smaller in the case of News CIF.

- Among the existing frameworks based on AQM, the smaller value of $threshold_{low}$, i.e., 10, is more preferred in both I/P+EAM and I/P+AM than the adopted values in their own works.

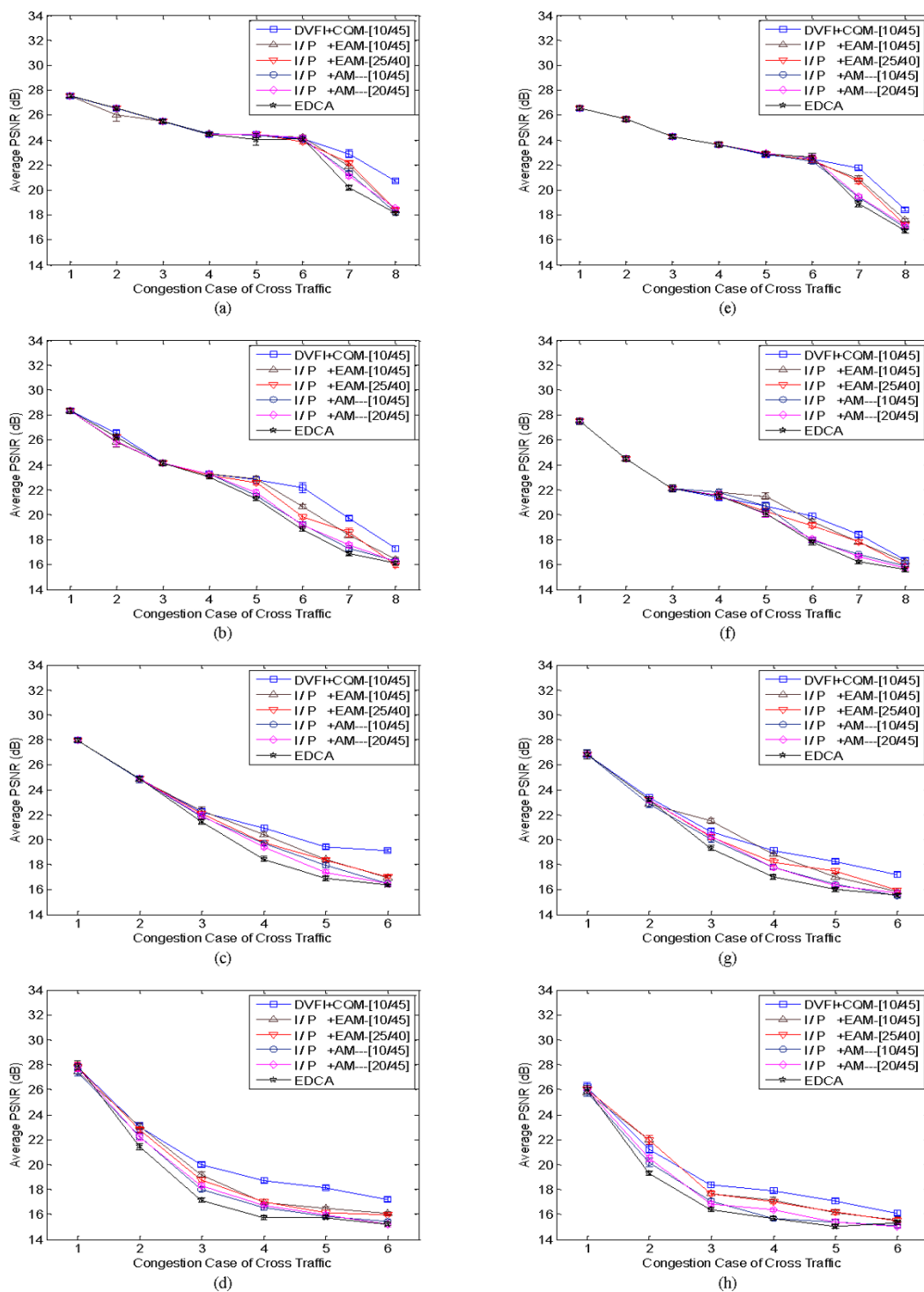


Figure 16 Average PSNR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in GOP = 9: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in GOP = 15: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (Foreman CIF.)

Table 6 further gives the largest PSNR gains (dB) of the proposed DVFI+CQM framework over the existing ones among different congestion cases of cross traffic

for the four input rates of video source ($IR = 128, 256, 384, \text{ and } 512$ kbps) in all the test video sequences both in QCIF and CIF. The largest gains over the best

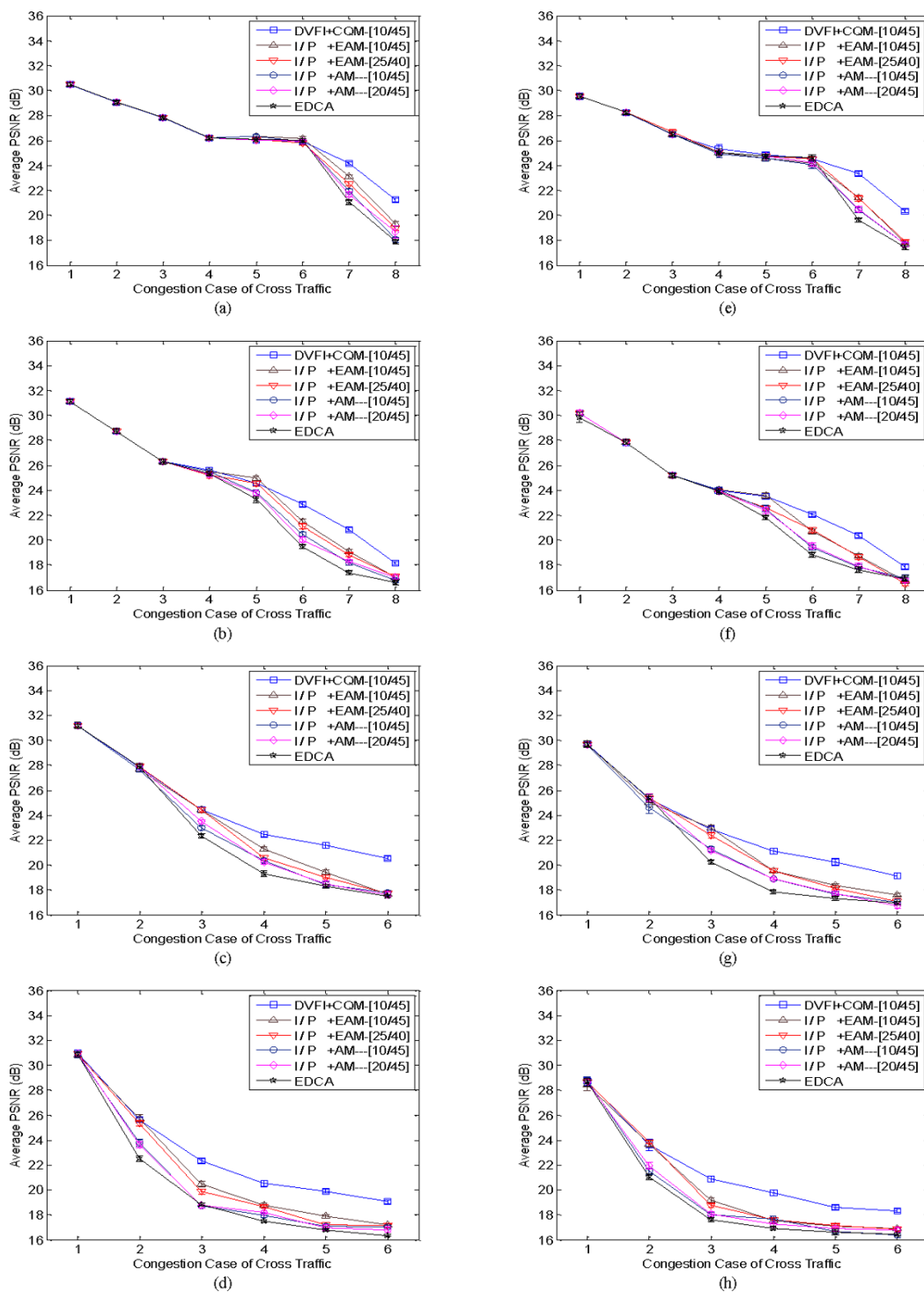


Figure 17 Average PSNR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in GOP = 9: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in GOP = 15: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (Foreman QCIF.)

existing framework I/P+EAM[10/45] range from 0.5 to 3.1 dB, and the ones over EDCA range from 2.2 to 5.3 dB.

5. Conclusions

A novel cross layer design for quality delivery of H.264 video streaming over 802.11e ad hoc networks, called

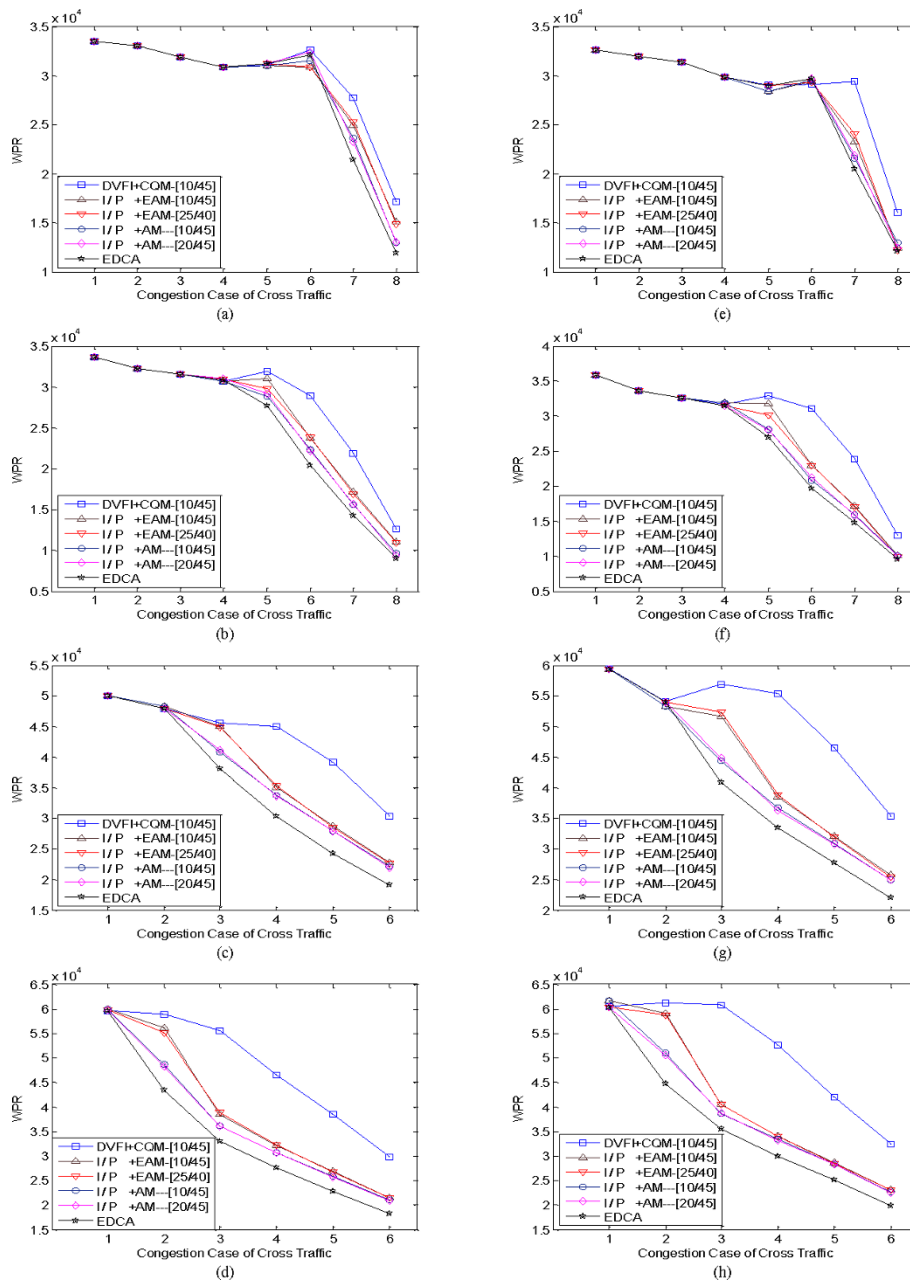


Figure 18 WPR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in GOP = 9: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in GOP = 15: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (Foreman QCIF.)

DVFI+CQM, has been proposed in this article. DVFI is proposed for a precise indexing of the video frame importance from the application layer, while the proposed CQM is a corresponding video packet mapping algorithm among the ACs in the MAC layer, which has multi-branched downward mapping probability functions according to the equal population grouping of DVFI. To support the superiority of this cross-layer framework, extensive tests have been conducted over

various input rates of video source in AC[2] and congestion cases of cross traffic in the other ACs. These tests also cover eight typical video sequences, i.e., four in QCIF and four in CIF formats, with the level of motion from *slow* to *fast*. The results show a consistent pattern of performance evaluation: the proposed DVFI+CQM framework outperforms the existing ones, and the I/P +EAM[10/45] framework takes the second place while EDCA is always the loser, as well expected. According

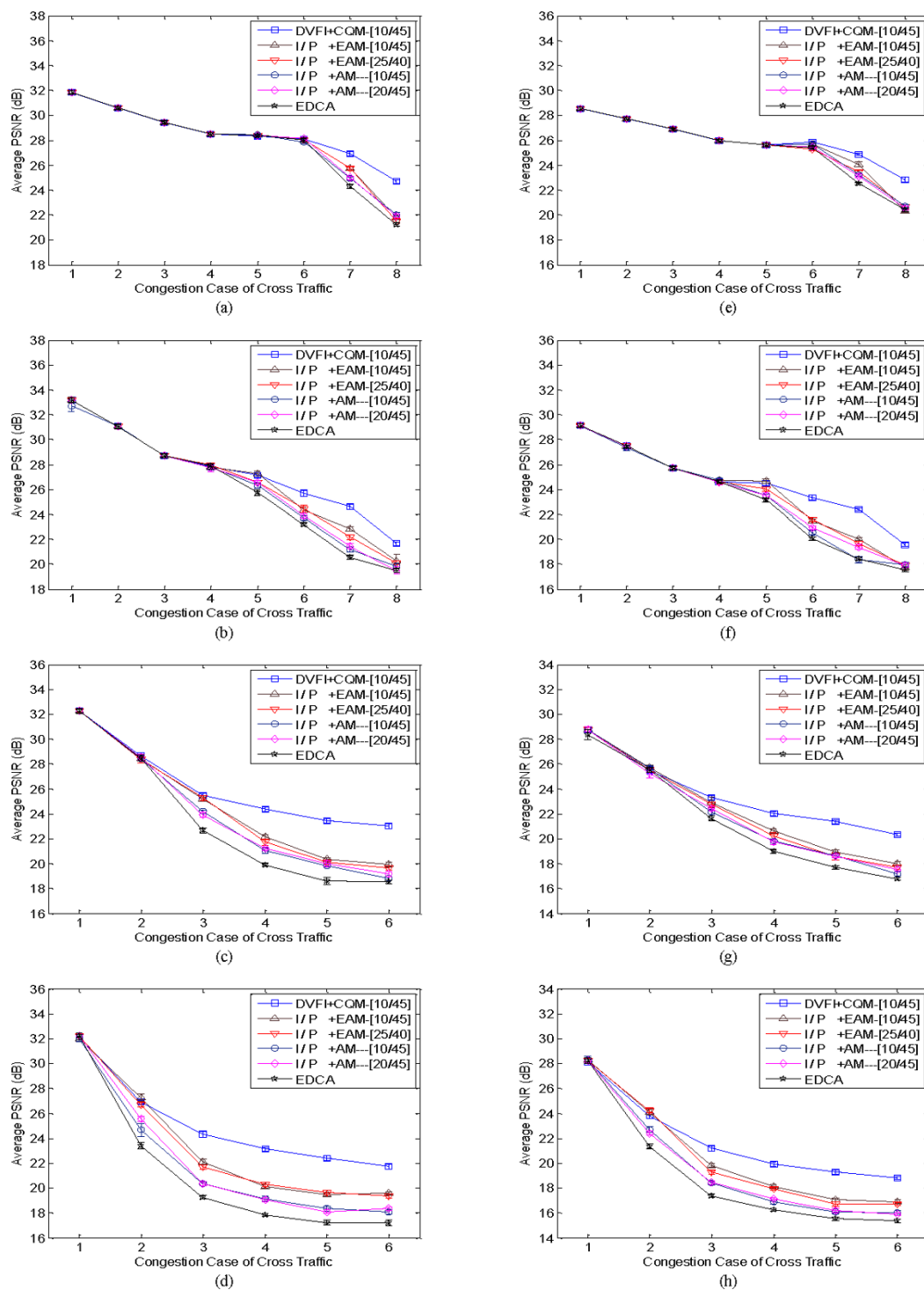


Figure 19 Average PSNR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in QCIF: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in CIF: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (Carphone QCIF and CIF in GOP = 9.)

to the summary results of Table 6, the largest gains of the proposed framework over I/P+EAM[10/45] range from 0.5 to 3.1 dB, and the largest gains over EDCA

range from 2.2 to 5.3 dB. These results indicate that the design principles in the proposed DVFI+CQM framework are successful and robust.

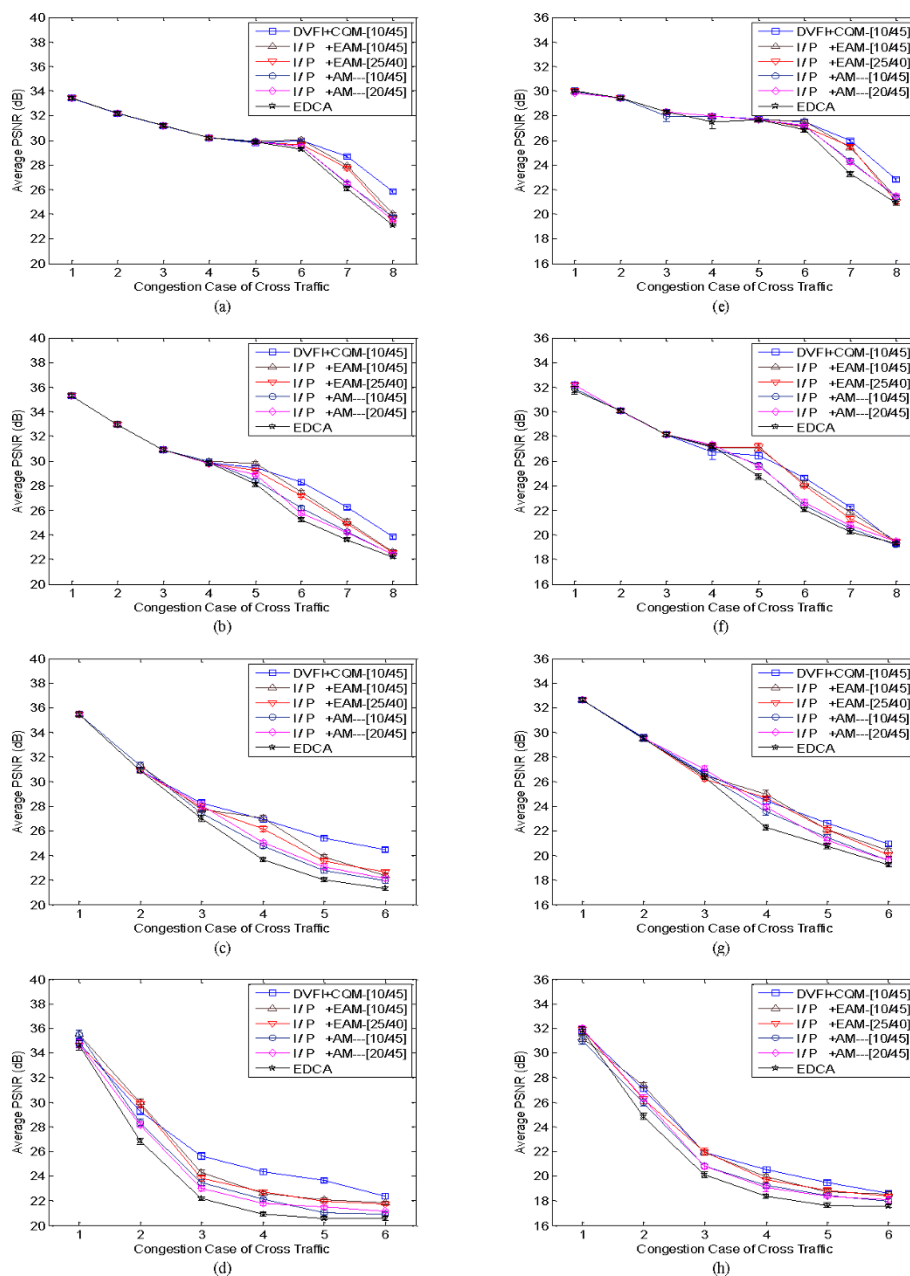


Figure 20 Average PSNR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in QCIF: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in CIF: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (News in QCIF and CIF in GOP = 9)

Recall that this framework is designed for video streaming over 802.11e, and *Principle 1* for the CQM algorithm in the MAC layer requires to keep the most important video packets in AC[2] as many as possible to make the best use of the AC[2] resources. It is obvious that this principle is not suitable for real-time or interactive video applications which would impose some queue delay limit on the use of AC[2]. This could be a good issue for future works. The essence of this issue

lies in the fact that AC[2] can no longer support as many video packets as possible, and more video packets need to be mapped to the other ACs. Thus, from the application layer, it may be helpful to have a more precise importance scheme directly at the video packet level, instead of inheriting the importance from the video frame level, so that one can really keep more important video packets in AC[2]. On the other hand, from the MAC layer, one should try to find a more

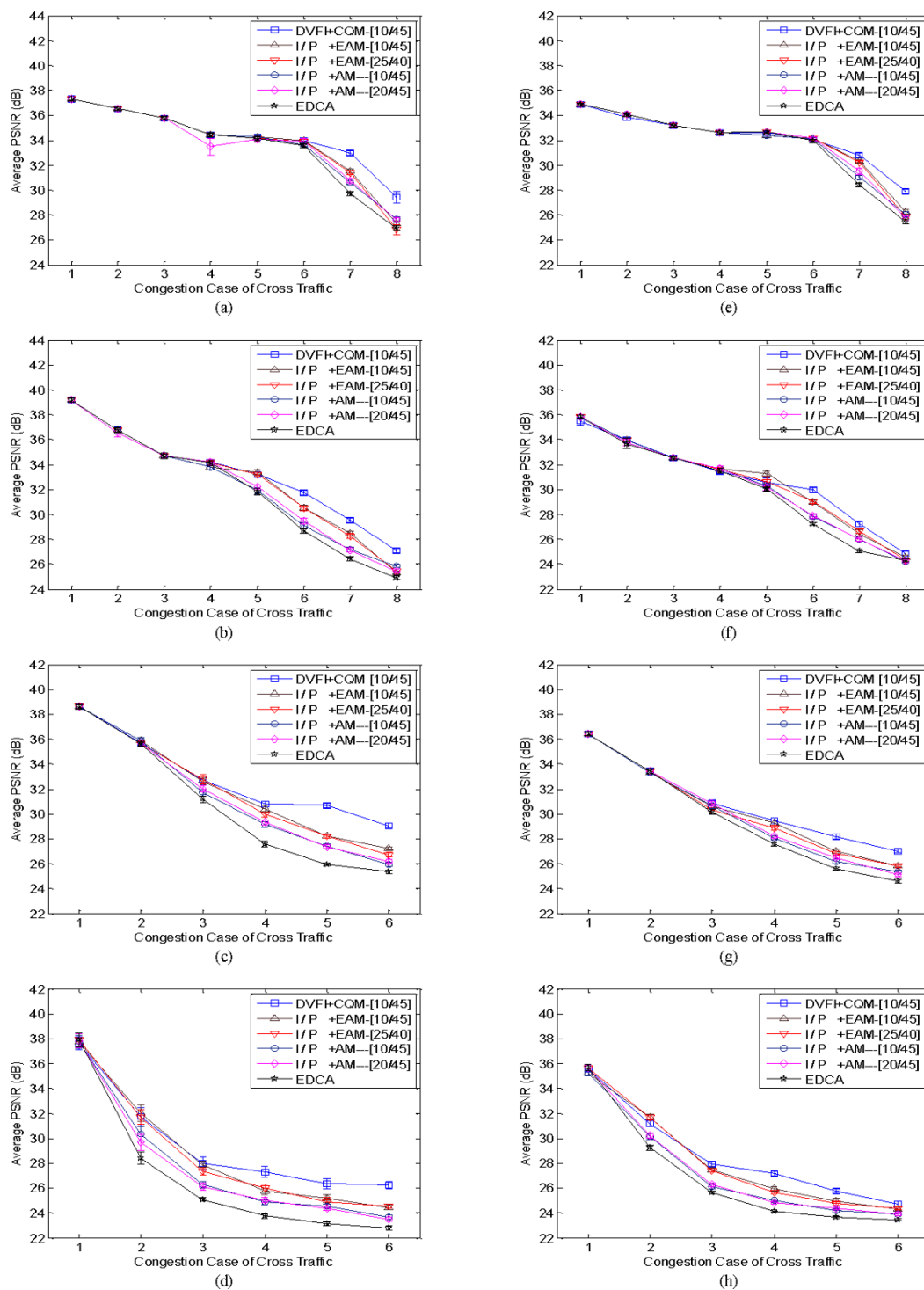


Figure 21 Average PSNR-based performance evaluation of different cross-layer frameworks versus congestion case of cross traffic for various input rates of video source, with the left half in QCIF: (a) 128 kbps, (b) 256 kbps, (c) 384 kbps, (d) 512 kbps, and the right half in CIF: (e) 128 kbps, (f) 256 kbps, (g) 384 kbps, (h) 512 kbps. (Mother-&Daughter QCIF and CIF in GOP = 9)

Table 6 Maximum PSNR gains (dB) of the proposed DVFI +CQM framework over the existing works among different congestion cases of cross traffic for four input rates of video source

Input rate (kbps)	QCIF				CIF			
	128	256	384	512	128	256	384	512
Foreman								
*Gain over								
V/P+EAM[10/45] (dB)	1.9	1.8	2.9	2.0	2.4	1.6	2.2	1.8
V/P+EAM[25/40] (dB)	2.3	2.0	2.8	2.7	2.3	2.3	2.1	2.0
V/P+AM[10/45] (dB)	3.2	2.7	3.2	3.6	2.5	2.9	2.6	2.3
V/P+AM[20/45] (dB)	2.6	2.8	3.1	3.6	2.2	3.0	2.6	2.2
EDCA (dB)	3.3	3.5	3.3	3.5	2.7	3.3	2.8	3.0
Carphone								
*Gain over								
V/P+EAM[10/45] (dB)	2.9	1.8	3.1	3.0	2.5	2.4	2.4	2.2
V/P+EAM[25/40] (dB)	3.2	2.4	3.4	2.8	2.2	2.8	2.8	2.6
V/P+AM[10/45] (dB)	2.6	3.4	4.2	4.0	2.1	4.1	3.2	3.2
V/P+AM[20/45] (dB)	2.8	3.2	3.9	4.3	2.3	3.0	2.8	3.1
EDCA (dB)	3.5	4.1	4.9	5.3	2.4	4.0	3.7	3.8
News								
*Gain over								
V/P+EAM[10/45] (dB)	1.8	1.2	2.1	1.7	1.4	0.5	0.6	0.7
V/P+EAM[25/40] (dB)	2.2	1.3	1.8	1.8	1.8	0.9	0.9	0.8
V/P+AM[10/45] (dB)	2.2	2.1	2.6	2.6	1.6	2.2	1.4	1.3
V/P+AM[20/45] (dB)	2.4	2.5	2.4	2.6	1.7	2.0	1.4	1.5
EDCA (dB)	2.7	3.1	3.4	3.5	2.7	2.6	2.2	2.2
Mother-&Daughter								
*Gain over								
V/P+EAM[10/45] (dB)	2.1	1.8	2.4	1.8	2.4	1.6	2.2	1.8
V/P+EAM[25/40] (dB)	2.6	1.6	2.5	1.7	2.3	2.3	2.1	2.0
V/P+AM[10/45] (dB)	2.4	2.6	3.3	2.6	2.5	2.9	2.6	2.3
V/P+AM[20/45] (dB)	2.2	2.4	3.3	2.8	2.2	3.0	2.6	2.2
EDCA (dB)	3.3	3.1	4.7	3.6	2.7	3.3	2.8	3.0

*Gain over = PSNR Gain of the proposed cross-layer framework DVFI+CQM over the existing ones.

efficient AQM not only for AC[2], but also for other ACs. For example, appropriately utilizing AC[3] for video packets without causing too much delay to the voice traffic might be a good way.

Acknowledgements

This study was supported by Taiwan National Science Council under grants NSC 99-2219-E-155-005 and 100-2221-E-155-064.

Competing interests

The authors declare that they have no competing interests.

Received: 28 July 2011 Accepted: 23 February 2012

Published: 23 February 2012

References

1. IEEE Std. 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Std (1999)

2. G Bianchi, Performance analysis of the IEEE 802.11 distributed coordinated function. *IEEE J Sel Areas Commun.* **18**(3), 535–547 (2000). doi:10.1109/49.840210

3. IEEE Std. 802.11e-2005 (Amendment 8 to IEEE Std 802.11-1999), medium access method (MAC) quality of service enhancements. IEEE Std (2005)

4. S Takeuchi, K Sezakit, Y Yasuda, Dynamic adaptation of contention window sizes in IEEE 802.11e wireless LAN, in *Proc 5th Int Conf Information, Communications and Signal Processing (ICIS'05), Bangkok, Thailand* 659–663 (2005)

5. L Gannoune, S Robert, Dynamic tuning of the contention window minimum (CW_{min}) for enhanced service differentiation in IEEE 802.11 wireless ad hoc networks, in *Proc 15th IEEE Int Symposium Personal, Indoor and Mobile Radio Communications (PIMRC'04), Barcelona, Spain.* **1**, 311–317 (2004)

6. L Gannoune, S Robert, N Tomar, T Agarwal, Dynamic tuning of the maximum contention window (CW_{max}) for enhanced service differentiation in IEEE 802.11 wireless ad hoc networks, in *Proc 60th IEEE Vehicular Technology Conf (VTC'04), Milan, Italy.* **4**, 2956–2961 (2004)

7. H Takabi, AH Moghadam, A Khonsari, Hybrid adaptation of the maximum contention window (CW_{max}) and minimum contention window (CW_{min}) for enhanced service differentiation in IEEE 802.11 wireless ad hoc networks. *Int J Comput Sci Netw Secur.* **6**(12), 281–290 (2006)

8. A Andreadis, R Zambon, QoS enhancement with dynamic TXOP allocation in IEEE 802.11e, in *Proc 18th Annual IEEE Int Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece*, pp. 1–5 (2007)

9. A Andreadis, R Zambon, QoS enhancement for multimedia traffics with dynamic TXOP limit in IEEE 802.11e, in *Proc 3rd ACM Workshop on Personal (Q2SWinet'07), Crete Island, Greece*, pp. 16–22 (2007)

10. N Cranley, M Davis, Video frame differentiation for streamed multimedia over heavily loaded IEEE 802.11e WLAN using TXOP, in *Proc 18th Annual IEEE Int Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece*, pp. 1–5 (2007)

11. JO Kim, H Tode, K Murakami, Service-based rate adaptation architecture for IEEE 802.11e QoS networks, in *Proc IEEE Global Telecommunications Conf (GLOBECOM'05), St. Louis, Missouri, USA.* **6**, 3341–3345 (2005)

12. VA Siris, C Courcoubetis, Resource control for the EDCA mechanism in multi-rate IEEE 802.11e networks, in *Proc 2006 Int Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), Niagara-Falls, Buffalo, NY*, pp. 419–428 (2006)

13. 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.* **33**(1), 107–114 (2006)

14. 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. *Springer Telecommun Syst.* **42**(2), 353–362 (2009)

15. N Chilamkurti, S Zeadally, R Soni, G Giambene, Wireless multimedia delivery over 802.11e with cross-layer optimization techniques. *Multimedia Tools Appl.* **47**(1), 189–205 (2010). doi:10.1007/s11042-009-0413-6

16. J Watkinson, *The MPEG Handbook*, 2nd edn. (Focal Press, Burlington, MA, 2004)

17. Information technology - Coding of audio-visual objects-Part 2: Visual, ISO/IEC 14496-2 (2004)

18. Advanced video coding for generic audiovisual services. ITU-T Rec. H.264 (2010)

19. H Schwarz, D Marpe, T Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans Circ Syst Video Technol.* **17**(9), 1103–1120 (2007)

20. M Flierl, B Girod, Multiview video compression. *IEEE Signal Process Mag.* **24**(6), 66–76 (2007)

21. G Sun, W Xing, D Lu, An novel loss protection scheme for H.264 video stream based on frame error propagation index, in *Proc 3rd Int Conf Communications and Networking in China (ChinaCom'08), Hangzhou, China*, pp. 820–824 (2008)

22. M Davcevski, T Janevski, Analysis of IEEE 802.11e QoS in multimedia environment, in *Proc 7th Int Conf Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS'05), Nis, Serbia and Montenegro.* **1**, 45–48 (2005)

23. M Thottan, MC Weigle, Impact of 802.11e EDCA on mixed TCP-based applications, in *Proc 2nd Annu Int Wireless Internet Conf. (WICON'06), Boston, USA*, pp. 1–9 (2006)

24. J Majkowski, FC Palacio, Dynamic TXOP configuration for Qos enhancement in IEEE 802.11e wireless LAN, in *Proc 2006 Int Conf Software, Telecommunications and Computer Networks (SoftCOM'06), Split-Dubrovnik, Croatia*, pp. 66–70 (2006)
25. Cisco IOS Software Releases 11.2, Weighted random early detection on the Cisco 12000 router http://www.cisco.com/en/US/docs/ios/11_2/feature/guide/wred_gs.pdf. Accessed 15 February 2011
26. C-H Mai, Y-C Huang, H-Y Wei, Cross-layer adaptive H.264/AVC streaming over IEEE 802.11e experimental testbed, in *Proc of the 71st IEEE Conference on Vehicular Technology (VTC 2010), Taipei, Taiwan*, pp. 1–5 (2010)
27. Video Trace Library (US Arizona State University, Arizona 2010) <http://trace.eas.asu.edu/yuv>. Accessed 5 May 2010
28. EvalVid (Germany TKN Telecommunications Networks Group, Berlin 2010) http://www.tkn.tu-berlin.de/menue/softhardware_components/software/experimental_code/evalvid_-_a_video_quality_evaluation_tool-set. Accessed 10 July 2010
29. K Fall, K Varadhan, The ns manual. <http://www.isi.edu/nsnam/ns>. Accessed 6 April 2011
30. FFmpeg Team, FFmpeg software. <http://ffmpeg.org>. Accessed 15 February 2011

doi:10.1186/1687-1499-2012-59

Cite this article as: Lai and Liou: A novel cross-layer design using comb-shaped quadratic packet mapping for video delivery over 802.11e wireless ad hoc networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:59.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
