Efficient MAC for Real-time Video Streaming over Wireless LAN

EILWOO BAIK, University of California, Davis
AMIT PANDE, University of California, Davis
PRASANT MOHAPATRA, University of California, Davis

Wireless communication systems are highly prone to channel errors. With video being a major player in Internet traffic and undergoing exponential growth in wireless domain, we argue for the need of a Video-aware MAC (VMAC) to significantly improve the throughput and delay performance of real-time video streaming service. VMAC makes two changes to optimize wireless LAN for video traffic: (a) It incorporates a Perceptual-Error-Tolerance (PET) to the MAC frames by reducing MAC retransmissions while minimizing any impact on perceptual video quality; and (b) It uses a group NACK-based Adaptive Window (NAW) of MAC frames to improve both throughput and delay performance in varying channel conditions. Through simulations and experiments, we observe 56-89% improvement in throughput and 34-48% improvement in delay performance over legacy DCF and 802.11e schemes. VMAC also shows 15-78% improvement over legacy schemes with multiple clients.

Categories and Subject Descriptors: H.5.1 [**Multimedia Information Systems**]: Video

General Terms: Design, Performance

Additional Key Words and Phrases: IEEE 802.11 standards, Video transmission, Wireless LAN

1. INTRODUCTION

Although advancements in wireless and cellular networks have improved network throughputs, the network still throttles the performance of video and high end applications. Cellular networks have migrated to 4G standards in many countries, yet it is impossible to cater to the ever-increasing demands of video applications using 4G alone. Most companies such as AT&T deploy WiFi hot-spots in public places such as Starbucks coffee or stadiums to migrate excessive data traffic to wireless networks. Apart from this, wireless networks are also used in enterprise and home networking.

Wireless LANs are fundamentally designed for convenient and adhoc pairing with access point for data traffic in a lossy environment. Their LAN resources get quickly exhausted in the presence of high bandwidth applications and varying channel conditions. However, the convenience of pairing with Access Point (AP) and obliviousness to video traffic properties can be the major performance issues for wireless networks.

Surprisingly, the 'last mile' connection again becomes an issue to provision all these services smoothly to end users. The last hop, usually wireless and shared among multiple clients, is the most crowded and saturated part of the network. The normal wireless access point(AP, 802.11g/n/ac) can theoretically support 54M/300M/1.3Gbps but the actual performance is significantly less due to protocol overheads. The network performance and video streaming rate is exponentially reduced by presence of multiple users and poor channel conditions. Figure 1 and 2 shows our experiments. Although the data rate of 802.11g is 54Mbps, we can achieve only around half of it, 24Mbps even in the good conditions. For example - 1080p Youtube video of 'The Dark Knight trailer' needs a download rate of 10 Mbps. Only two viewers can be provisioned at an AP (802.11g) to watch

Author's addresses: Eilwoo Baik, Computer Science Department, University of California, Davis; Amit Pande, Computer Science Department, University of California, Davis; Prasant Mohapatra, Computer Science Department, University of California, Davis One Shields Avenue Davis, CA 95616;

DOI: http://dx.doi.org/10.1145/0000000.0000000

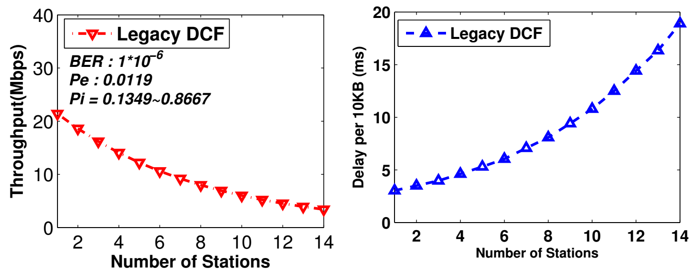


Fig. 1. Throughput and Delay of Legacy Wireless AP(802.11g) : P_e(Error Prob.), P_i(Idle Prob. to send), BER(Bit Error Rate)

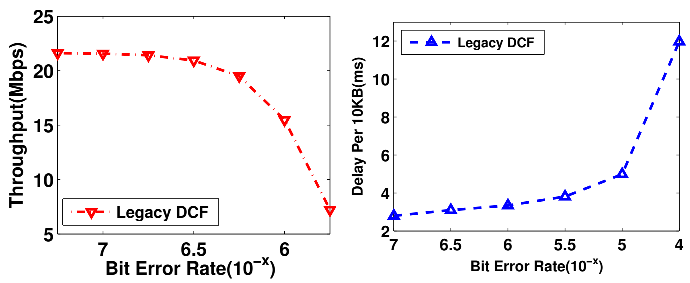


Fig. 2. Throughput and Delay of Legacy Wireless AP(802.11g) : Increasing BER

such movie even in very good channel conditions! The throughput decreases while delay increases exponentially. This motivates the design of a new scheme that provides graceful degradation of network resources for video content services.

With increased proliferation of smart phones and portable devices, there is an impending need to allow all these devices to be connected transparently with the network, providing high performance computing and delivering enhanced real time multimedia. The next 5 years are projected to provide unabated mobile video adoption in diverse applications such as TV broadcast, video chats, video-on-demand and rich media services. Mobile video traffic contributed to 51% of aggregated mobile data traffic at the end of 2012, and will account for 86% of entire traffic by the end of 2016. Traffic from wireless devices will exceed traffic from wired devices by 2014 [CiscoSystem 2011]. Thus, it is desired to optimize video delivery in wireless networks.

As a medium access method, DCF (Distributed Coordination Function) is popularly used in AP, which is a contention-based scheme by using random-backoff duration to obtain medium access. However, it does not support QoS and IEEE 802.11e provisions EDCA (Enhanced Distributed Channel Access) which can improve the performance and support QoS (Quality of Service) to some degree. Although it offers service-class prioritization for video traffic, it is still agnostic to video characteristics. It doesn't consider the low delay requirements of high quality real-time video traffic and doesn't exploit the fact that video packets are resilient to some degree of error without perceptual degradation of delivered video quality.

In this paper, we design a novel scheme that is implemented on top of 802.11e, called Video-MAC (VMAC) for real-time video-aware packet delivery in MAC layer. First, we propose a Perceptual Error Tolerance (PET) scheme in MAC layer that reduces redundant retransmissions of lost packets if they are inconsequential to the human eye. We use the inability of human eye to measure slight losses in reconstructed video to define a threshold for acceptable frame losses in the MAC layer. Loss of MAC frame corresponds to loss in I/P/B frames in videos, each of which has a different impact on perceptual quality. Our scheme accounts for this aspect and assigns different weights to these losses.

We also propose a scheme based on a group NACK transmission to flexibly cope with fluctuating network conditions. This minimizes the signaling overhead not only in good conditions, but also in bad condition. Further, we sub-partition TXOP[1] into windows, which are adapted to size according to channel conditions. We combine them to build a group NACK-based Adaptive Window (NAW) scheme. The main contributions of VMAC are as follows:

— We propose a Perceptual Error Tolerance (PET) mechanism to adapt the retransmissions in MAC according to the desired perceptual quality. The MAC frame losses are weighed according to their impact on perceptual quality, and retransmissions are only triggered when overall losses are beyond a perceptual threshold.
— We derive the weights for different types of MAC frame losses (corresponding to I/P/B losses) using extensive experiments on a prototype selective-frame-drop *middleware*.
— We propose a group NACK-based mechanism, combined with Adaptive Window (NAW) scheme to improve the protocol efficiency for video transmissions.
— The $window$ to be introduced in the proposed NAW has the variable length to cope with varying network conditions by using a Multiplicative-Increase Tolerant-Multiplicative-Decrease algorithm.

In our implementation, we observe 56-89% improvement in throughput and 34-48% improvement in delay performance over legacy 802.11g (DCF; Distributed Coordination Function) and 802.11e schemes. VMAC also shows 15-78% improvement over legacy schemes with multiple clients.

VMAC can be implemented as a dedicated video access point such as Google Chromecast [Chromecast 2014], AppleTV [AppleTV 2014] and home video set-top-boxes, or it can be integrated to traditional Access Points (APs) with a slight update in protocol. In the first case, the dedicated video access point streams video traffic using VMAC. In the second scenario, all traffic (from all clients) seek resources from AP. Then, the normal traffic will be routed via regular 802.11 MAC protocol while video traffic will be moved to VMAC. VMAC can be integrated as a module to Wi-Fi Multimedia (WMM) certified APs which are enabled for EDCA and TXOP. VMAC needs a simple modification in both sender and receiver. Channel sensing remains same for VMAC and non-video traffic goes the usual way as in EDCA. Live video streaming services, live video chat as well as on-demand video can be streamed using VMAC for throughput and delay improvements.

The rest of paper is organized as follows: Section 2 gives a brief review of related works. Section 3 gives background of 802.11 standards for video delivery and understanding of frame-packaging in video codecs. Section 4 gives motivation behind VMAC while Section 5 and 6 details the design elements of our scheme. In Section 7, we discuss the performance of VMAC over existing schemes. Section 8 concludes with directions for future work.

2. RELATED WORK

There are many efforts on development and advancement for the WLAN protocols. However, these approaches are limited to MAC parameters [Doefexi et al. 2003], signal condition [Pavon and Choi 2003], coverage [Sendra et al. 2011] and number of users [Roshan and Leary 2004]. In addition, there are few studies which directly apply video properties of application layer into MAC layer [Chen et al. 2010].

Inan et. al. [Inan et al. 2006] propose a scheduler for multimedia QoS in 802.11e with parameters traffic classification, packet size, interval, direction. The proposed scheduler can improve performance, but there are no fundamental changes to reduce the protocol overhead. Majkowski and Palacio [Majkowski and Palacio 2006] propose dynamic TXOP configuration for QoS enhancement based on average number of packets allocated in AC queue. However, this scheme cannot achieve good results when most users are in the same video service category. Most works [Siris and Courcoubetis 2006; Li et al. 2006; Majkowski and Palacio 2006] on TXOP of 802.11e invest and focus on only TXOP size/duration not on transmission procedure.

[1]TXOP or Transmission Opportunity is introduced in 802.11e and allows sending multiple MAC frames per transmission.

Bi [Bi 2012] presents a strategy for service differentiation by using packet loss rate for 802.11 wireless LAN. Unfortunately, this paper does not consider the impact of lost packets on video quality. Without considering video properties, it can bring more severe side effects in end user QoE [Huszák and Imre 2010]. Ksentini et al. [Ksentini et al. 2006] propose a MAC modification to improve video transmission for H.264 videos. However, they differentiate H.264 video partitions at MAC layer and classify them to different access priorities in 802.11e. Thus, the performance of this scheme is limited to the performance of 802.11e Block ACK scheme. Bucciol et al. [Bucciol et al. 2007] use a perceptual ARQ, however it is unclear how distortion thresholds can be created for each packet in a video, without even considering the impact of I/P/B frame packaging. Existing MAC based approaches focus on application delay constraints of video delivery but are quite agnostic to video quality [Chen et al. 2010; MacKenzie et al. 2009; She et al. 2011]. In selectively dropping video packets, the most significant point is not in finding allowable number/threshold of video packet-drops, but in figuring out which video packets can be dropped depending on the video properties in the given network conditions. Chen et al. [Chen et al. 2010] consider the number of packets dropped within a GOP to limit packet retransmissions. But they do not consider frame-type and its non-uniform impact on video quality. She et all [She et al. 2011] propose a scheme that drops only delayed packets in MAC layer. Their approach does not optimize MAC itself, but throttles the input traffic by eliminating the packets which are not able to make the receiver's application layer in time. It is also limited to real-time video traffic. In event of a sporadic network congestion, the scheme will fail to deliver required video quality, whereas, as we will show later, our scheme works at MAC level and adapts itself to channel conditions. MacKenzie et al. [MacKenzie et al. 2009] considers packet dropping and packet mapping schemes. The packet mapping schemes give higher access level to important video data while packet dropping schemes drop packet pro-actively based on some heuristics. Moreover, the results are shown under only one video resolution (CIF, 352*288) without network variations. None of the approaches tend to improve the MAC performance itself.

Application layer approaches focus mainly on video buffer management and control [Balan et al. 2003; Dua and Bambos 2007], but ignore retransmission issues that significantly contribute to performance of wireless networks. Gangadharan et al. [Gangadharan et al. 2011] propose a frame work of variable buffer size with lossy video quality. Their approach shows tradeoffs between buffer sizing and frame dropping strategy. Although it implies better application buffer usage, it leads to wastage of network resources. Feamster and Balkrishnan [Feamster and Balakrishnan 2002] propose an application layer scheme that uses different network layer schemes (TCP and UDP) for I packets and P/B packets respectively. Performing selective retransmissions in application layer [Feamster and Balakrishnan 2002] will incur higher costs due to redundancy in lower layer retransmissions. Sen et al [Sen et al. 2010] design 'Approximate' communication system for media application. They also consider the video frame type with each priority in PHY. Such work is complementary to our approach.

Most of the existing works for video streaming are based on two approaches; the first approach are from video-buffer management, video frame control, etc in application layer. The approaches cannot contribute to increasing efficiency of wireless resources, because the video packets arrived in application layer means the wireless network resources (in MAC/ network layer) are already wasted. The second approaches are based on only MAC parameters such as signal power, strength, interference, etc. These approaches do not consider the properties of video so that they cannot support the video data when video connections becomes majority in the WiFi network. From this issue, we propose the Video-aware MAC (VMAC), where we apply the properties using in application layer into MAC layer so that VMAC can control video stream in MAC in advance and reduce the significant network metrics like delay, throughput, jitter. We implement perceptual tolerance at MAC layer, saving volumes of retransmissions and improving transmission efficiency by introducing our proposed schemes at 802.11e QoS MAC operations to boost performance and reduce delay for realtime streaming video services. Plus, ours is the first work to consider an adaptive window inside TXOP to achieve better performance tradeoff between 802.11e Burst ACK and Block ACK mechanisms.

NOTE: The nomenclature 'frame' refers to MAC layer frames in this paper. When we discuss about video frames, they are explicitly mentioned.

3. MOTIVATION

Video data has higher throughput requirements and lower delay deadlines (particularly for real-time streaming or video chat applications) than normal network traffic. The resilience of video data to some degree of packet losses has been known to research community, but it has not been leveraged to achieve better coordination and performance in lower layers.

3.1. Perceptual Video Property in MAC

The final evaluator of video content to be transmitted over wireless network is the human consumer. Visual perception is the ultimate QoE (Quality of user Experience) for the received video content. Human visual system can only recognize 10$\sim$12 frames per second (fps) in a video [Read and Meyer 2000]. The frame rates of 24 to 30 fps are progressive formats and widely used for producing video contents, giving a smooth temporal experience to human viewer. It is quite well known fact that our visual system can tolerate some quality loss of video due to persistence of vision in visual cortex [Moorthy et al. 2012]. Existing works consider frame dropping in application layer in terms of buffer management [Gangadharan et al. 2011], or controlling for late-arrived packets. Some papers consider dropping packets in MAC layers, but they make decisions based on network parameters [Chen et al. 2010; She et al. 2011] without consideration of video properties.

The goal behind our scheme is to exploit perceptual resilience of videos in MAC layer to reduce retransmissions. Although UDP connection ignores lost/erroneous and out-of-time packets, MAC layer makes upto 4 attempts to retransmit MAC frame. The continuous retransmission in lossy wireless LAN causes service degradation in delay-sensitive video traffic flows. Our approach is to provide MAC layer with some degree of Perceptual Error Tolerance (PET). For example, A PSNR[2] value above 37 usually implies perfect perceptual reconstruction at receiver, although there may be losses in network transmission [Klaue et al. 2003]. Mapping this tolerance to MAC level frame losses allows us to alleviate the network from the burden of retransmitting those extra bits which don't significantly contribute to perceptual quality, and hence allows better goodput for realtime video applications. More users can be served, with higher satisfaction rate, than the existing standards.

3.2. Tradeoff between Burst ACK and Block ACK Schemes

In order to support QoS and protocol efficiency, 802.11e standards introduce Access Category (AC) for QoS and the concept of TXOP for protocol efficiency, called as Burst ACK scheme. In Burst ACK, multiple MAC frames are sent together within one TXOP duration, each requiring an ACK to indicate successful transmission. A TXOP is a bounded time interval during which a station can send as many frames as possible. The TXOP duration is varied according to application requirements (voice/ video/ data). This improves the overall throughput and delay performance over legacy DCF scheme. However, Burst-ACK scheme still requires ACK for every MAC frame in TXOP. This can achieve quick recovery for frame loss/error in a bad channel, but it can turn into protocol overhead in good channel condition.

The Block ACK (BA) scheme in 802.11e further improves the performance by alleviating the need of individual ACK for each MAC frame transmission. Instead, only a single group ACK message is exchanged at the end of TXOP duration. Removal of ACK for each MAC frame can dramatically decrease the overhead of ACK exchanges, thereby increase net throughput. However, if there is any lost frame in TXOP, it cannot be retransmitted immediately due to absence of individual ACKs. This may lead to increased delay and jitter, which is crucial for real-time video streaming applications in bad channel. Although delay overheads are of few ms, it can lead to substantial degradation

[2]PSNR is a widely used video quality metrics. A low value ($\leq$ 20) indicates poor quality transmission while a higher value indicates higher video reconstruction quality.

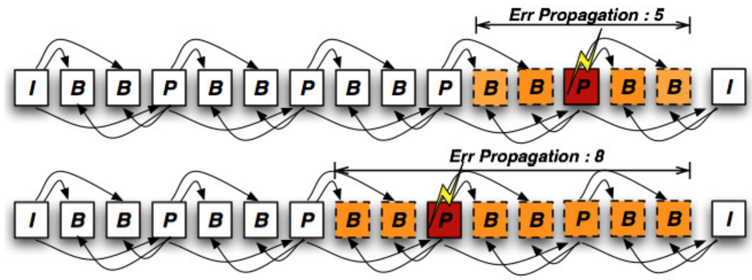


Fig. 3. Error Propagation in a GOP in the even of P frame loss (M=15, B=2)

in user quality in end-end video chat applications. *It is necessary to have flexible tradeoffs between two schemes according to the network conditions*.

3.3. Our Approach

We first design a Perceptual Error Tolerance (PET) scheme in MAC layer. There are three types of video frames (I/P/B frames) and each has different impact on perceptual quality. The loss of MAC frame corresponding to I-type video frame leads to most significant perceptual quality degradation. Lesser degradation is experienced by a P or B frame loss. Through extensive experiments and theory, we define a different penalty for the loss of MAC frame when it corresponds to I or P or B video frame. Next, we add the penalty over a moving time window. In case of a MAC frame loss, we determine whether or not that loss corresponds to significant perceptual loss by summing the penalty over current window. Only when it exceeds a pre-determined threshold, we trigger a selective feedback mechanism, as explained next. Finally, we implement this idea into MAC layer to reduce retransmissions. By avoiding retransmissions for MAC frame losses not amounting to perceptual degradation, PET can boost overall network performance.

Unlike existing 802.11 standards, our scheme uses group-NACK, which improves the throughput in good channel conditions. The size of window is adaptive so that it is able to take advantages from both burst ACK scheme and BA scheme depending on channel conditions. Burst scheme is strong at immediate recovery of frames in high loss/error scenarios, but leads to wastage of time, especially in good channel conditions (where feedback is redundant). On the contrary, BA can transmit data without waiting for inter MAC frame waiting time, while it is vulnerable to expeditious packet recovery in unstable network condition. In order to integrate two schemes, we introduce a scheme which can adapt the window size inside TXOP.

We propose NAW (NACK based Adaptive Window) scheme to reduce the protocol overhead of MAC layer while improving efficiency. NAW introduces a flexible window within TXOP. A group of MAC frames are transmitted together in a single window and a single group-NACK is required for them. This group of frames is called as ‘Window”. In a given TXOP, the size of window is adapted according to the network conditions. In good channel conditions, the performance is close to Block ACK scheme while the performance is close to Burst ACK scheme in case of high losses. We use TCP-like algorithm to vary the size of window.

A sender is able to transmit MAC frames upto the size of the current window with a single group-NACK response. If the transmission is successful, the window size is increased exponentially. If it fails, the window size is maintained, or decreased in case of consecutive failure. Unlike the TCP congestion control, the window size is not reduced by a single failure. If the window size is reduced by only intermittent frame loss/error event (which happens all the time in wireless networks), it will have less transmission efficiency. If network condition becomes worse (successive frame drops), the window size is decreased. The window can adapt its size to the fluctuating wireless network conditions. This is performed within a given TXOP, as shown in the Figure 7(4) later. Our VMAC scheme has these two components: NAW and PET for efficient and flexible video content delivery.

4. PERCEPTUAL ERROR TOLERANCE (PET)

In this section, we define a scheme for perceptual error tolerance in video streaming. The goal is to boost network performance when applying perceptual video tolerance in MAC layer. With this, we also explore tradeoff between MAC frame drop event and its impact on visual quality.

The generation of video packets is defined by the factors such as quality (bit rate), resolution, Group of Picture (GOP) size and frame packaging scheme. Thus, certain frames have a higher impact on video quality than others. We want to figure out *how much frame loss at MAC layer is acceptable for good perceptual delivery?*

All modern video coding standards, starting from H.261, use a hybrid video coding in what they call as Video Coding Layer (VCL). A Network Abstraction Layer (NAL) formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by particular transport layers (such as Real Time Transport Protocol) or storage media. VCL processes video frames into GOP and classifies the frames with as I, P and B type frames which are coded in a specific manner to save overall bandwidth usage [Richardson]. Drop of a frame containing I/P/B information will lead to different perceptual quality degradation (due to predictive coding in video codec).

4.1. Impact of frame type in packet loss

Error in a single I frame in a GOP brings propagates to entire frames of associated GOP. If a P frame is erroneous, the all associated P or B frames to be reconstructed based on it are damaged. Figure 3 shows two examples of the effect of P frame being damaged. If last P frame in a GOP is lost, the number of damaged frames is 5 including the lost packet itself. However, loss of the second-to-last P frame affects 8 frames. Due to error propagation, loss of a prior P frame damages subsequent P frames also. Error propagation of P frame can be different depending on the position of the lost P frame within a GOP. B frame is not used for references of the other frame types. Thus, loss of B frame does not affect other frames. The number of P frames in a GOP can be derived from two parameters usually used to specify a GOP: N is for the size of GOP, and M for the number of B frames between I or P frame as follows :

$$N_P = \frac{N}{M+1} - 1$$

The fraction of erroneous frames (in GOP) by error in a P frame ($\mathbb{D}$) can be given as below :

$$\begin{aligned}
\mathbb{D}_{N_P} &= (2M+1) \times \frac{1}{N} \\
\mathbb{D}_{N_P-1} &= (3M+2) \times \frac{1}{N} \\
&\cdots\cdots \\
\mathbb{D}_1 &= ((N_P+1) \cdot M + N_P) \times \frac{1}{N}
\end{aligned}$$

Average loss caused by a P frame loss is calculated as expected number of distorted frames ($\mathbb{D}$)

$$\begin{aligned}
\mathbb{E}(\mathbb{D}_P) &= \frac{1}{N_P} \cdot \left(\sum_{i=1}^{N_P} \mathbb{D}_i \right) \\
= \frac{1}{N_P \times N} & \left(M(\frac{N_P(N_P+1)}{2} + N_P) + \frac{N_P(N_P+1)}{2} \right) \\
\Rightarrow \mathbb{E}(\mathbb{D}_P) &= \frac{M(N_P+3) + N_P + 1}{2N}
\end{aligned}$$

Thus, we observe that dropping different frame-types has different impact on video quality.

$$\lambda_I \left(= \frac{N}{N}\right) = 1$$
$$\lambda_P = \frac{M(N_P+3)+N_P+1}{2N}$$
$$\lambda_B = \frac{1}{N} \tag{1}$$

Thus, dropping I frame impacts entire GOP, P has effect on λ_P fraction of frames and B is limited to one frame.

4.2. Experiment Setup and Implementation

To investigate the impact of video frame-type on the perceptual video delivery we used the following setup.

We set up a video streaming server and a client using VLC media player. It uses RTP/ RTCP over UDP for video streaming. However, selective dropping of I/P/B frames at MAC layer is not possible with existing software. We implement and insert a *middleware* between sender and receiver which can selectively drop packets at a desired packet loss rate. It sniffs video byte stream, detects video frame type in MAC frames and selectively drops a MAC frame based on I/P/B frame-type. Video frame type, position of frame-begining and slice informations are presented at video header in the payload. Thus, the middleware needs to decode only header information to obtain video frame type, and find position where a new video frame begins if a single packet is overlapped over two consecutive video frames. The middleware inspects packet header, not full packet information so, it does not require any additional workloads and overhead in processing and classifying video frame types. After the detection, the middleware apply the given rules and policies to the MAC frames. The video codecs supported are MP4/MPEG2Video and file container is mp4/mpg. In our setup, the middleware sits at MAC of sender.

4.3. Evaluation

Figure 4 shows the examples of images captured in our experiments. It can be visually seen how I MAC frame loss has maximum damage. However, we can also observe that different I frame losses have different impact on video quality. Same for P and B frames. 0.5% loss in I-frame has no noticeable visible distortion, but around 1% I-frame drop shows broken big lines in the middle of display. In the case of P-frames, there is no recognizable distortions until 3%. Above 3%, P-frame drop leads to blocking and blurring effects on the display. Around 5% drop of B-frame leads to loss of pixels and stretched image at the bottom of Figure 4 (h).

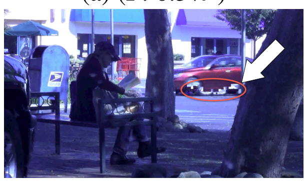
Figure 5 shows the impact of selective MAC frame drops. PSNR values are reported for a representative video sequence ($Oldwoman$). The impact of a I frame loss spreads across entire GOP while the impact of P frame is less spread over the GOP and the impact of a B MAC frame is limited to one video frame. The width of red-dotted line indicates GOP size of video $Oldwoman$. B or P frames can have more stronger impact than losses of I-frame even if the duration is shorter than P and B. I-frame always have bigger and longer impacts than B or P within the same GOP. However, there are various motion speeds and contents in a video, and this means different GOP can have different impacts. Thus, I-frame in a GOP of fast motion can be smaller impacts than P/B frame in the different GOP of slow motion. However, I frame always bring $longer$ propagation than P/B frame because the size of GOP (fixed value) is not changed in the middle of video playtime. In Figure 5, the green line around frame number 620 present B-frame and it shows bigger impact than I-frame which is around frame number 950. However, P/B video frame always have shorter duration than that of I video frame. (Compare width of P/B with that of I-frame).

As we proposed earlier, PET allows the receiver to not report MAC frame losses if they are within the threshold of perceptual tolerance. We ran experiments with different videos and found


(a) (I : 0.5%)


(b) (I : 1%; Recognizable)


(c) (I : 5%; Strong distrotion)


(d) (P : 0.5%; very small dist. in car)


(e) (P : 3%; small dist. in car)


(f) (P : 5%; over entire display)


(g) (B : 1%; Not recognizable)


(h) (B : 3%; Not recognizable)


(i) B : 5% ; thick blur. at bottom))

Fig. 4. Impacts of different drop rates of each frame type; Example of a video ($Oldwoman$)

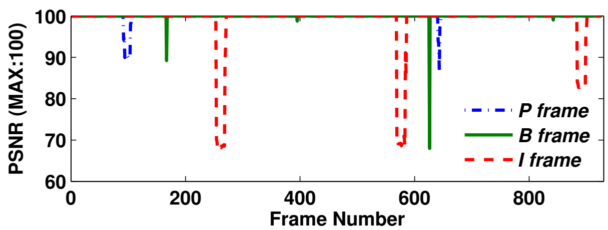
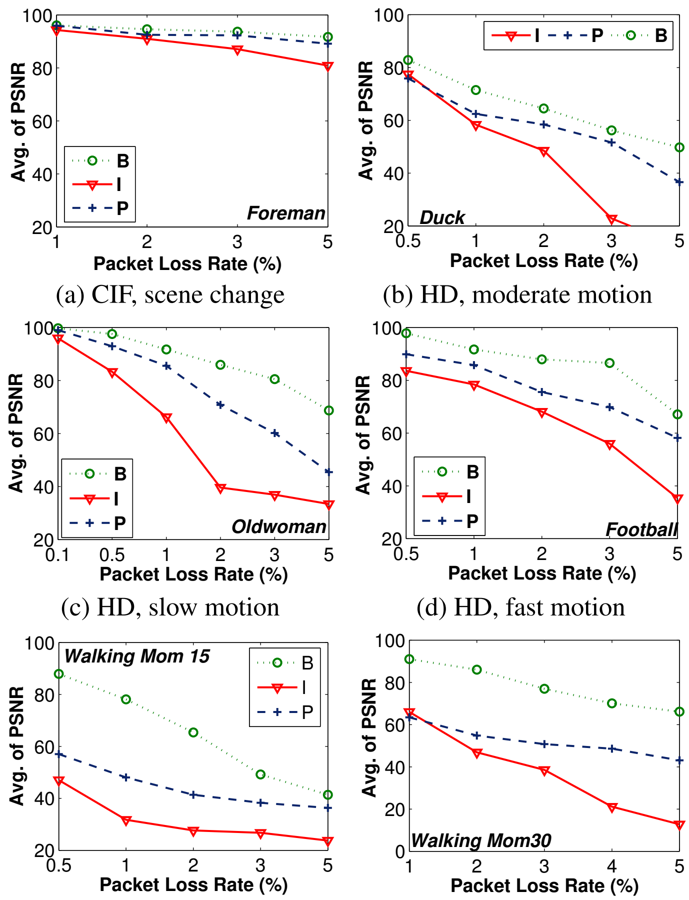


Fig. 5. Effect of dropping individual I/P/B MAC frames on sample video. I frame loss has a lasting impact while B frame loss has impulsive impact on perceived video quality. P frame loss is also spread over multiple video frames. ($Oldwoman$ video HD, GOP(15,2), MAC frame drop rate is set to 0.5% on each frame type).

that when a MAC frame is lost, it corresponds to a distortion in the video frame which the lost MAC frame belongs to, and subsequently leads to quality loss : Damage in I video frame leads to stronger damages and longer damage propagation across over entire GOP than that in P or B video frame. The impact of B video frame is the smallest since B frame is not referred and used to other frames. Figure 6 shows one of our experiment examples to explain impacts depending on the video frame type in a GOP. The details of video contents and properties are reported in the Table I.

Figure 6 shows the variation in perceived video quality with varied channel conditions. The x axis shows packet loss rate at MAC layer while y axis shows average PSNR value of aligned frames. Selective dropping MAC frames belonging to each video frame type (I/P/B) are performed in the middleware and it can be observed that different videos suffer different degradation under same network conditions. Videos with low resolution (CIF 352×288 in (a) for foreman video) are more robust to packet loss compared with (b) a full HD (1920×1080) video. Figure 6 (b$\sim$d) are three videos with different motion-speed contents but similar bit rates. These cases report that the strength of motion-speed in a video can have different tolerances on video quality. (e) is a high quality HD video with aesthetic details. Thus, there is significant drop in video quality with packet loss rate for detailed or high quality videos as compared with regular videos (c-d). (e) and (f) are the same video



(a) CIF, scene change (b) HD, moderate motion

(c) HD, slow motion (d) HD, fast motion

(e) HQ, HD, GOP size 15 (f) HQ, HD, GOP size 30

Fig. 6. Evaluating the impact of individual MAC frame losses for different videos.

with different GOP sizes of 15 and 30. It can be observed that smaller GOP size leads to higher degradation in video quality with packet loss rate.

We summarize these observations from the experiments with following explanations :

Quality: A high quality video has lots of high frequency information, therefore it requires a large bit rate compared to normal video. For example - a single I video frame for '$Walking\ mom$' video splits into 273 MAC frames, while only 25$\sim$50 MAC frames are used for other videos in our dataset. Another observation is that even a single MAC frame loss causes stronger impact in higher resolution video rather than in lower resolution. A high resolution (HD) video frame splits into one or more MAC frames, while even one MAC frame may represent several video frames in a low resolution. A MAC frame loss in low resolution simply causes temporal jitter without spatial distortions while it may lead to spatial distortions and hence lower quality for HD videos.

Resolution: Low resolution videos tend to have higher resilience to MAC frame losses than that of high resolution videos. In the case of low resolution videos such as Foreman, a single I frame splits to only 4$\sim$5 MAC frames, while a single I frame splits to 25-30 MAC frames in HD videos of moderate bit rates (3 $\sim$ 5 MBps). One or some losses of MAC frames leads directly to video frame loss(es) in low resolution videos, while a high resolution video has distortions in a video frame rather than video frame loss(es). Distortions in video frames (I or P) propagate within the range of GOP, and this bring more degradation in user QoE than skipping some video frames due to packet

Table I. Details of sample videos(Codec : MPEG4/H264; Container MP4)

Name	Length	Resolution	GOP(N:M)
Old Woman	30 sec	1920×1080	15:2
Adventure	30 sec	720×576	15:2
Foreman	10 sec	352×288	10:2
Duck	30 sec	1920×1080	15:2
Football	10 sec	1920×1080	12:2
Walking mom	30 sec	1440×1080	15:2, 30:3
Highway	80 sec	352×288	10:2

Table II. PSNR threshold and corresponding drop rates for I, P and B video frame types)

Scheme	Target PSNR Range	Threshold			
		Value \ Frame type	I	P	B
VMAC37	$\overline{37}$	$\bar{\rho}$	0.3	1.7	4
		σ	0.0483	0.560	1.063
VMAC32	$\overline{32}$	$\bar{\rho}$	0.5	2.5	6
		σ	0.0805	0.659	1.328

losses. Figure 6 (a) and (b) depict that a high resolution video experiences more impacts of packet loss than a low resolution video. The other cases of HD of (c), (d), (e) and (f) shows sharp quality degradation than (a).

GOP size: The size of GOP affects the propagation range of video distortion. Figure 6(e-f) show the impact of frame drop for different frame sizes. Errors in B frames are not propagated to other video frames because B frame is not referred by other frames for prediction. However, I frame type are referred by all frames within GOP for prediction and P frame as used by a few frames, so they are more vulnerable to packet loss rate than B (in that order). Loss in I frame is more harmful to video quality when GOP size is large (see Figure 6(e-f)).

Motion Higher-motion videos consume more MAC frames to contain information. The increased number of MAC frames means that each MAC frame loss contributes to lesser portion of video frame. In the same condition of resolution, a MAC frame in a slow-motion takes larger potion in a video frame than the portion of a fast-motion video. This means that the MAC frame loss causes larger distortion in the video frame of a slow-motion video than a fast one, and this distortion propagates to the other frames. Thus, The slow-motion videos are more vulnerable to MAC frame losses than higher-motioned ones. This fact can be observed in Figure 6 (c) and (d).

Frame-type Significant impact was found when we tried to correlate the impact of drop of I/P/B frames with perceptual quality in received videos. It is neither possible to exactly determine video quality/resolution at network level, nor to find a linear relationship in these variations with MAC frame loss events. Therefore, we take the average across a pool of videos (we choose HD videos) to obtain exact thresholds.

We next define perceptual thresholds and translate them to packet loss rate at MAC layer as follows:

(1) PSNR is the most widely used video quality metric. A PSNR value greater than 37 is found to give no-perceptual loss to human observer while a value $\sim$ 32 means good quality visual experience [Klaue et al. 2003]. We select a suitable perceptual threshold ($PSNR_{thres}$)

(2) We take a pool of videos and use our middleware to selectively drop I/P/B MAC frames from the video at different loss rates. We next use the tool presented in [Chan et al. 2010] to measure PSNR of reconstructed videos. The tool presented in [Chan et al. 2010] realigns videos in absence of time-stamp to help us perform PSNR evaluation. We map $PSNR_{thres}$ to individual weights for I/P/B frame errors ($\delta_I(i)$, $\delta_P(i)$ and $\delta_B(i)$ respectively).

(3) We average the I/P/B frame error weights over a pool of video traces to get final values of δ_I, δ_P and δ_B.

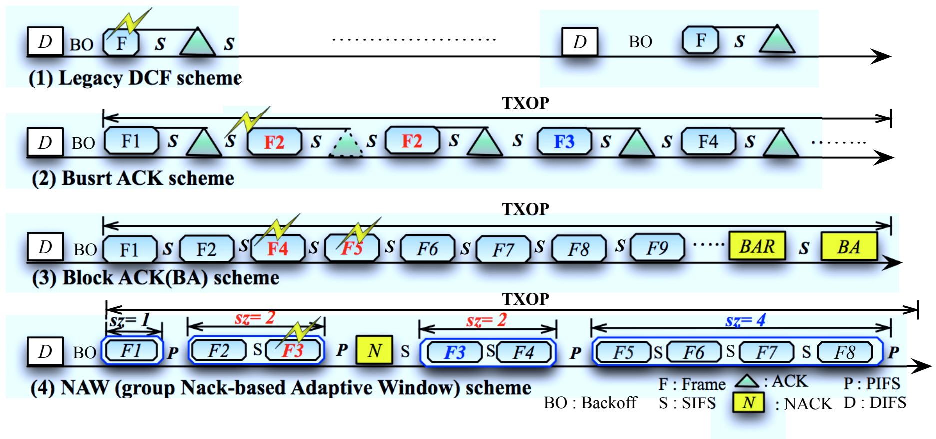


Fig. 7. Comparison of NAW with existing schemes. Legacy DCF transmits single MAC frame in a channel access. 802.11e Burst ACK sends n MAC frames in a burst while Block ACK scheme sends n MAC frames together with a single ACK. NAW is based on group-NACK and the frames are grouped into 'Window'. Its size is adaptive in nature depending on channel conditions.

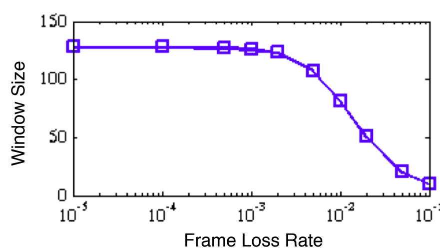


Fig. 8. Plot showing adaptation of window size with changing loss rate of MAC frames

(4) When an expected MAC frame is not received at the receiver, the receiver first determines the frame type of the lost MAC frame based on the video frame type of the previous and next video frame in the GOP sequence (as available). Next, it increases the weight depending on whether the video frame type is I, P or B. If the aggregate of weights for PET window (size 100) is above 1, a NACK is sent for current packet, otherwise, no NACK is sent. The threshold is calculated over a window (of length, say 100 MAC frames) after which the previous packet loss is not aggregated.

We derived the threshold values of acceptable packet loss rate for each video frame type under the conditions of $PSNR_{thres} = 37$ and $PSNR_{thres} = 32$. The results are given in Table II. It can be seen that the ratio of γ_B/γ_I is close to λ_I/λ_B from Equation 1 (M=2 and N varies from 10 to 15 for most content, averaging around 12, $N_P = 2$). The ratio γ_P/γ_I is close to $\alpha\lambda_I/\lambda_P$ where $\alpha \simeq 3.5$ instead of theoretical assumption of 1. The impact of P frames loss is lesser than what we had estimated in theoretical model which accounted for full frame loss instead of MAC frame loss. This is because, MAC layer losses may not lead to full-quality losses due to partial reconstruction of frames from other MAC layer packets. The obtained priorities to be applied in our middleware are in agreement with the theoretical model depending on the video frame type. We use these thresholds in our experiments.

5. NACK-BASED ADAPTIVE WINDOW

5.1. Selective group-NACK Response

ACK is traditionally preferred over NACK because NACK-based scheme can lead to a poor performance in cases where NACK is also lost subsequent to a MAC frame loss. In this case, a sender will transmit future frames without knowledge of a frame loss. Or the user may never issue a NACK if the packets destined to him were all lost. This is not the usual case for high-volume video traffic. After receiving out of order data, a receiver sends the NACK again and the sender transmits the previous data again. In order to overcome this issue, we introduce group-NACK unlike the traditional NACK response. TXOP is divided into several windows and each window contains one or more than one MAC frame. For each window, a single group-NACK is transmitted to the sender. Unlike the loss of traditional NACK, loss of group-NACK can be handled in two ways :

(1) If the loss is below the perceptual threshold, the lost group-NACK can be consolidated into PET and ignored

(2) If the loss is greater than the threshold, group-NACK is triggered in the next window.

These combination of PET and group-NACK of window in TXOP can dramatically maximize the efficiency in video data communication in 802.11e MAC and performance in realtime video streaming services without video quality degradation.

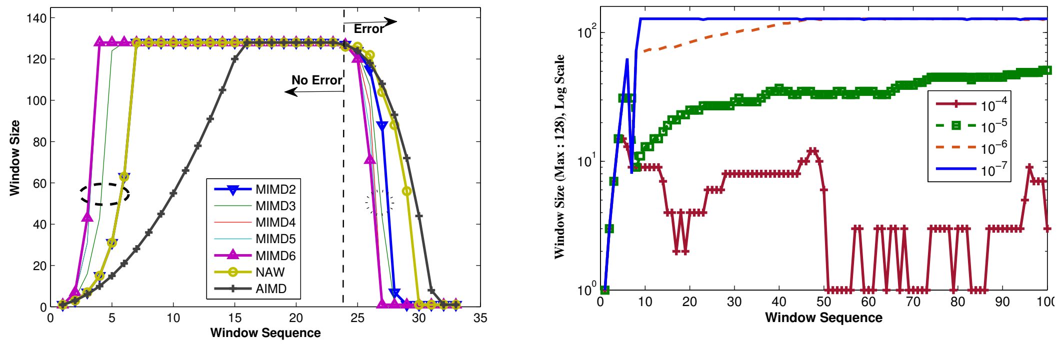
5.2. Adaptive Window in TXOP

We use the concept of 'Window' within TXOP. Multiple MAC frames in the current window are transmitted and expect a single group-NACK reply. The size of window is adaptive varying from 1 (minimum) to TXOP length depending on channel conditions. After transmitting frame(s), a sender should wait PIFS interval to receive NACK from its receiver. If the sender does not receive NACK, it assumes that the packet was well-received. If there is(are) frame loss(es), a receiver sends NACK to inform the sender with information (bit-mark) of dropped MAC frame(s). The duration of PIFS is bigger than SIFS (Short Inter Frame Space) and shorter than DIFS (DCF Inter Frame Space). Unlike BA and burst ACK scheme, a sender waits for larger time interval (PIFS instead of SIFS) to process the NACK from a receiver. We choose PIFS (Point coordination function IFS) over SIFS to give receiver enough time to process NACK for entire window.

We discussed the tradeoff between Block ACK and Burst ACK schemes in 802.11e earlier. In poor channel conditions, Block ACK leads to increased delay while Burst ACK causes redundancy and lower throughput (in good channel conditions). We break TXOP into smaller and variable blocks, which we call as 'Window'. Figure 7.(4) illustrates the structure of window within a TXOP. The system starts with a window size of 1, increases it to 2 with subsequent transmission. In case of frame loss, NACK is issued by a receiver which leads to retransmission of the lost MAC frame ($F3$ of (4) in Figure 7) in next window. Because of MAC frame loss, the window size is not incremented in this transmission. In next transmission, window size of 4 is used and so on.

The size of window is adaptive, varied using a Multiplicative-Increase Tolerant-Multiplicative-Decrease algorithm which has some resemblance to TCP congestion control algorithm. Packet loss is more likely to happen in the bursty manner than isolated losse [Borella et al. 1998]. Multiplicative-increase helps our algorithm to quickly reach suitable window size. We also perform multiplicative-decrease to reduce window size so that lost MAC frames can be recovered immediately without waiting for another TXOP. However, it is triggered only after 2 times of consecutive packet losses in order to avoid inefficiency due to sparse and intermittent packet losses.

The idea is explained as follows: Window size (sz) is initialized to 1. In next round, Window size is multiplied by 2 with every successive transmission reaching a maximum value equal to largest multiple of 2 less than TXOP length ($sz \leq 2^{\lfloor \log(sz_{TXOP}) \rfloor}$). In case of an unsuccessful transmission, sz is scaled by a factor of two. However, given the lossy nature of wireless network, stand-alone MAC frame losses are not considered. MAC frame losses over more than two consequent windows lead to reduction in window size. This helps to prevent window size from dramatic decrement due to intermittent packet loss and from hasty increment in continuous/dense packet loss/error conditions.



(a) AIMD, MIMD ($2 \sim 6$) and NAW (b) Actual examples of window size in different BERs

Fig. 9. Variations of Window size

The lost MAC frames are recovered in next window. The variations in window size reflect the state of wireless channel. Figure 8 shows the general adaptation of (fixed) window size depending on MAC frame loss rate, obtained using simulation over 10^5 frames.

MIMD in NAW : As shown in the Figure 9 (a), any multiplicative incremental rates does not show big difference to reach full window size in no error/ loss environment. However an additive increment takes more than 15 steps to reach maximum window size. Each step causes additional delay of the inter frame space between windows and the number of not-transmitted MAC frames. Thus, NAW choose multiplicative increment in increasing window than additive increment for quick response for no error/loss environment. Figure 9 (a) shows how window is decreased in the case of continuous error/loss occurrence, shown in after step of 25. Unlike the existing multiplicative decrement (MD), MD in NAW show smoother drops in decreasing window at consecutive error/losses conditions. This helps NAW cope with efficiently intermittent errors/losses.

The figure 9 (b) ($log\ scale$ in Yaxis) depicts the real variation of window size depending on the different error rates. While the window approach the maximum size quickly at 10^{-7}, it takes more time to reach the max at 10^{-6}. After the window touches at the max, it does not decrease the window size sharply for intermittent error/loss. Thus, our proposed scheme is able to get as good performance as Block ACK(BA) at good network conditions. On the contrary, NAW is cautious in increasing at high error rates of 10^{-5} or 10^{-4}. Reckless try to increase window size in bad network conditions casues unnecessary redundant retransmissions, and it leads to waste of window. At the highest error rate, the window behavior becomes similar to Burst ACK scheme to cope with recovering high errors/losses.

Therefore, our proposed scheme can adopt its behavior depending on the channel conditions in order to process immediate recovery for lost frames. This is illustrated in the results shown in Figure 7. The flexible length of window can have the both of advantages of the burst ACK and BA scheme. The algorithm is detailed in Algorithm 1 and 2. The lost/erroneous MAC frames can be easily recovered in the next window and NAW does not need to wait for another TXOP interval like BA scheme.

6. PERFORMANCE

We measure the performance of VMAC over existing schemes in this section.

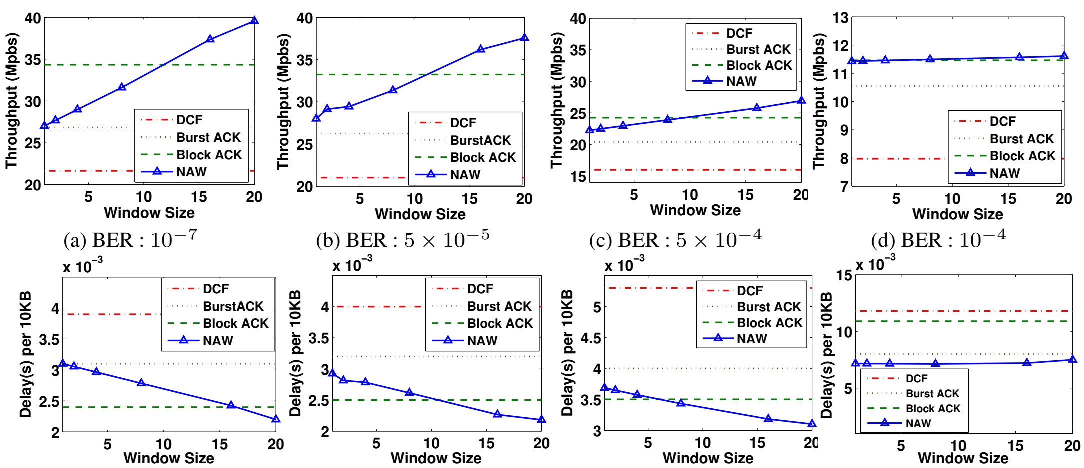
6.1. Setup

The details of parameter settings for the simulations are given in Table III. The length of TXOP was restricted to 3 ms as specified in 802.11e standards. We use 802.11g which restricts the theoretical data bound to 54 Mbps. BER was varied in the range 10^{-4} to 10^{-7}.

ALGORITHM 1: Sender side modifications for VMAC (NAW) algorithm

(A) Sender side modifications for VMAC (NAW) algorithm;

sz : Current NAW Window Size;
while *frames to send* **do**
  wait for PIFS interval ;
  if *NACK(t)* **then**
    if *#NACK(t)=1 & #NACK(t-1)=0* **then**
      $sz = sz$;
    else
      $sz = sz/2$;
    end
    Select NACKed frames for next Window
  else
    $sz = sz \times 2$;
  end
  Transmit sz MAC frames in a single Window;
end



(a) BER : 10^{-7} (b) BER : 5×10^{-5} (c) BER : 5×10^{-4} (d) BER : 10^{-4}

(e) BER : 10^{-7} (f) BER : 5×10^{-5} (g) BER : 5×10^{-4} (h) BER : 10^{-4}

Fig. 10. Practical throughput and delay performance depending on different window sizes

6.2. At Fixed Window Size

First of all, we show the performance of NAW scheme for different fixed window sizes and various channel conditions. Figure 10 gives throughput and delay when using variable window size within given TXOP. We can observe how different window sizes give different performance when we use different network conditions. When the bit error rate is very low ($\sim 10^{-7}$) small window size leads to performance very close to burst ACK scheme while large window size (> 12) gives a performance comparable to Block ACK scheme. The improvement over Block ACK is mainly due to NACK instead of ACK being used in our scheme. We can see consistent better performance for different window size in good channel conditions. At poor channel conditions (such as the case in Figure 10(d) where bit error rate is 10^{-4}), we find that a larger window size doesn't lead to higher throughput or lower delay. Figure 10 (e)-(h) show the performance of delay with each different fixed window size. In poor channel conditions Burst ACK has smaller delay than Block ACK scheme, but our scheme is able to outperform both.

ALGORITHM 2: Receiver side modifications for VMAC (NAW+PET) algorithm

(B) Receiver side modifications for VMAC (NAW+PET) algorithm;

```
wait for PIFS interval ;
T=0 % Threshold variable; n % Current frame number;
W(n) % perceptual loss for frame n ; sz % Current NAW Window Size;
for all packets within NAW Window do
    if frame not received then
        % Apply Perceptual Error Tolerance;
        Determine frametype;
        if frametype==I then
            T=T+1/δ_I ;
            W(n)=1/δ_I;
        end
        if frametype==P then
            T=T+1/δ_P ;
            W(n)=1/δ_P;
        end
        if frametype==B then
            T=T+1/δ_B ;
            W(n)=1/δ_B;
        end
        if W(n − 100) ≠ 0 then
            T=T-W(n-100);
        end
        if T > 1 then
            notify NACK;
            T=0;
            W=0;
        end
    end
    Aggregate NACK notifications and send;
    if Actual window size > sz then
        % implies loss of NACK information;
        sz = sz × 2;
        Aggregate NACK(t-1) notifications with NACK(t);
    end
    if NACK(t) then
        % Update NAW Window size;
        if #NACK(t)=1 & #NACK(t-1)=0 then
            sz = sz;
        else
            sz = sz/2;
        end
    else
        sz = sz × 2;
    end
end
```

Table III. Parameters

Param	Values
Type of traffic	Video
Protocol	RTP/RTCP
Video packet length	1316 * 8(bits)
Total frame length	1370 * 8(bits)
TXOP length	3ms
CWmin	16
Contention time	(CWmin-1)/2
Transmission prob.	2/(CWmin-1)
Bit error rate(γ)	$10^{-7} \sim 10^{-4}$
Frame error rate	$0.1349 \sim 0.8667$
Data rate	54Mbps
Basic rate	6Mbps
Time slot	9 μsec
AIFSN (video)	2

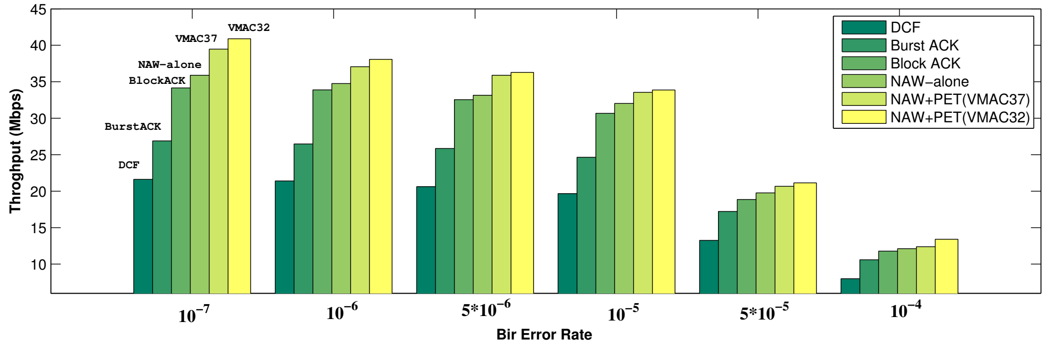
6.3. Evaluation in varying channel conditions

The overall performance of VMAC over existing schemes is shown in Figure 11. There is a consistent performance gain of 60-80% while using VMAC over legacy schemes. Delay gain is also consistent at 35-45% over legacy scheme of 802.11g. By default, VMAC sets the PSNR threshold to 37. We have also shown the results for case where PSNR threshold is set to 32 (good but not best video quality). VMAC37 has more strict conditions (lower drop-rates shown in Table II) than that of VMAC32, so VMAC32 can show better performance consistently. We can see that VMAC outperforms other schemes in all channel conditions. As the bit error rate increases, total throughput naturally decreases, however, VMAC 37 are able to show better results in bad channel condition of $2 * 10^{-4}$. Particularly in poor channel conditions, it pays to set $PSNR_{thres}$ to smaller values such as 32. This gives significant improvement of over 10% in throughput and delay gain over VMAC 37. The delay performance in Figure 11 shows tussle between Block ACK and Burst ACK in varying channel conditions. We have measured end-end delay at application layer (Vcodec buffer) for these measurements and not the packet level delays. This is more practical because application layer delay is more closer to user experience or observation than lower layer delay. In bad channel condition, Burst ACK can have shorter delay than Block ACK. Even if BA can receive more MAC frames than Burst ACK in the MAC layer, burst ACK can pass the MAC layer-receiving buffer packets earlier to the application layer while Block ACK has to wait for next TXOP to recover the loss or erroneous frames.

Unlike Block ACK, VMAC can immediately recover the error/loss in the next window without any delay to wait for next TXOP. Let us say the sequence number of the current window is (n). If frame losses happen, the sender sends NACK to request retransmissions in the next window $(n + 1)$. Even if the lost frames are skipped retransmitting in the next window $(n + 1)$ due to loss of the NACK, the lost frames are recovered quickly in the window of $(n + 2)$. The adaptive window size and use of NACK in VMAC, along with PET, allows it to achieve significant gains in delay as well as throughput performance. VMAC with different thresholds (32/37) can ignore retransmissions for video frames. Thus, the outperformance can be different depending on the video content properties and configurations. Table IV presents the average of actual PSNR achieved and 95% confidence interval. We can see that VMAC is able to ensure desired quality levels to most videos. This experiments are done in the video pool and parameters mentioned in Table I, II and III. The bit error rate (γ) can be categorized : [Very good $\leq 10^{-7} \leq$ Good $\leq 10^{-6} \leq$ Bad $\leq 10^{-5} \leq$ Very poor $\leq 10^{-4}$].

6.4. Evaluation with multiple clients

In presence of multiple users, throughput and delay degrade quickly because of channel contention. We consider a scenario where multiple clients are making parallel video connections and competing for network resources. VMAC doesn't interfere with channel access mechanism of WiFi therefore



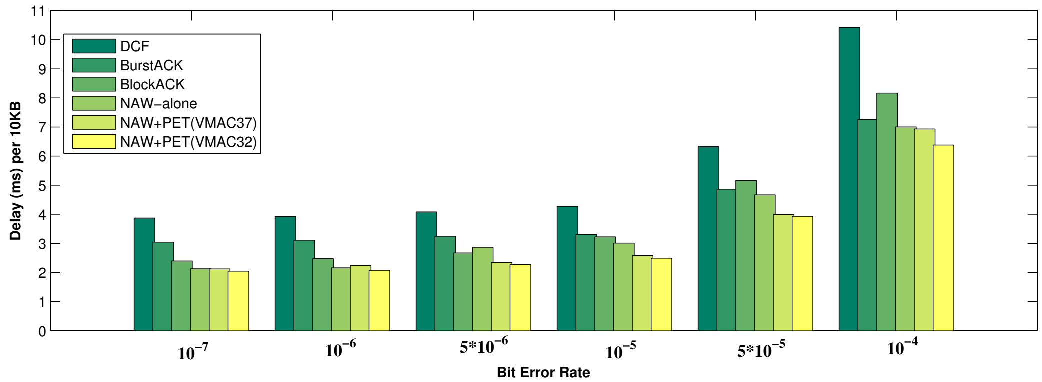


Fig. 11. Performance of VMAC in different channel conditions

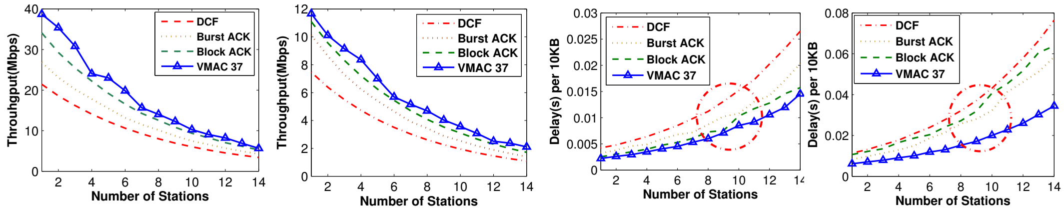
Table IV. Target PSNR vs Actual measured PSNR

Type	Target PSNR	Actual-Measured PSNR	
		Avg.	95% Confidence Interval
VMAC37	∼37	40.0874	[38.1617 - 46.3639]
VMAC32	∼32	32.9025	[31.6798 - 34.1251]

it also degrades in performance with multiple clients. However, it can be seen that the degradation is graceful. VMAC also outperforms the other schemes when increasing the number of clients. For 14 clients, we have throughput gain of almost 67% at BER 10^{-6} and 100% at BER of 10^{-4} in comparing legacy scheme (DCF). The delay gain also follows a similar trend. The significant improvement in delay performance at high error rates is attributed to PET scheme which reduces the retransmission overhead.

6.5. Exact Perceptual Tolerance (EPT)

In above results, we averaged the perceptual tolerance of different video contents to achieve a static value for the PET threshold of I, B and P frames. It is quite possible for video servers (such as Netflix or Youtube) to pre-compute and supply these threshold values for individual videos. In that case, VMAC will be able to use exact threshold value for the videos and maintain consistent visual performance for all content types. In the realtime video streaming services, the video session information by SDP(Session Description Protocol) is delivered to receivers for specific video and session information such as media type, format, and all associated properties. VMAC threshold information can also be delivered to VMAC receivers. Threshold information includes only a few numbers for PET rate so it does not add any overhead in network. Using such scheme, EPT can offer



(a) Throughput at BER : 10^{-6} (b) Throughput at BER : 10^{-4} (c) Delay at BER : 10^{-6} (d) Delay at BER : 10^{-4}

Fig. 12. Throughput and Delay performance for multiple stations

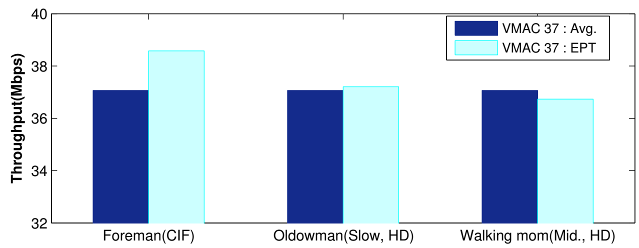


Fig. 13. Comparing performance of VMAC with EPT at BER(10^{-6})

degree of freedom to users to set limits on received video quality. Figure 13 compares performance of existing VMAC with designated perceptual tolerance of each video. VMAC 37 with EPT gives a slightly different throughput than the case when averaged thresholds are used.

7. DISCUSSION & CONCLUSIONS

In this paper we presented VMAC, an attempt to optimize video delivery over wireless network by redesign of MAC protocol. We proposed PET to control retransmissions in MAC depending on the video perceptual quality degree and given conditions, thus leading to save network resources. Secondly, NAW scheme has been proposed to improve the protocol overheads through tradeoff between Burst ACK and Block AC schemes. We demonstrated 56-89% improvement in net throughput and 34-48% improvement in delay performance over legacy schemes, 802.11g and Burst ACK and block ACK schemes of 802.11e. VMAC is defined for video-dedicated traffics by using group-NACK, and PET scheme are all based on error-resiliency of video data. It may be possible to perform similar optimizations for other scalable error-resilient applications such as some image or audio codecs. Varying the perceptual threshold can be helpful in poor networking conditions where reducing PSNR threshold can reduce stalls in video playback while reducing video quality.

REFERENCES

AppleTV. 2014. AppleTV. (2014). https://www.apple.com/appletv/

Anand Balan, Omesh Tickoo, Ivan Bajic, Shivkumar Kalyanaraman, and John Woods. 2003. Integrated Buffer Management and Congestion Control for Video Streaming. In *Proc. of the 10th IEEE International Conference on Image Processing, Barcelona, Spain*.

S. Bi. 2012. Service Differentiation Strategy Based on Packet Loss Rate for 802.11 Wirelesses LAN. In *IEEE Fourth International Conference on Computational and Information Sciences (ICCIS),*. 849–851.

Michael S Borella, Debbie Swider, Suleyman Uludag, and Gregory B Brewster. 1998. Internet packet loss: Measurement and implications for end-to-end QoS. In *IEEE ICPP Workshops on Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing*. 3–12.

P. Bucciol, E. Masala, E. Filippi, and JC De Martin. 2007. Cross-layer perceptual ARQ for video communications over 802.11 e wireless networks. *Advances in Multimedia* 2007, 1 (2007), 1–1.

O. Cabral, A. Segarra, and F.J. Velez. 2009. Implementation of Multi-service IEEE 802.11e Block Acknowledgement Policies. *IAENG International Journal of Computer Science* 36, 1 (2009), 85–93.

A. Chan, K. Zeng, P. Mohapatra, S.J. Lee, and S. Banerjee. 2010. Metrics for evaluating video streaming quality in lossy IEEE 802.11 wireless networks. In *Proceedings IEEE INFOCOM*. 1–9.

Chih-Ming Chen, Chia-Wen Lin, and Yung-Chang Chen. 2010. Cross-layer packet retry limit adaptation for video transport over wireless LANs. *IEEE Transactions on Circuits and Systems for Video Technology,* 20, 11 (2010), 1448–1461.

Google Chromecast. 2014. Google Chromecast. (2014). http://www.google.com/chrome/devices/chromecast/#cc-promo

CiscoSystem. 2011. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011 - 2016. (2011).

A. Doefexi, S. Armour, B.S. Lee, A. Nix, and D. Bull. 2003. An evaluation of the performance of IEEE 802.11a and 802.11g wireless local area networks in a corporate office environment. In *IEEE International Conference on Communications,*, Vol. 2. 1196–1200.

Aditya Dua and Nicholas Bambos. 2007. Buffer management for wireless media streaming. In *IEEE Global Telecommunications Conference, 2007.* 5226–5230.

N. Feamster and H. Balakrishnan. 2002. Packet loss recovery for streaming video. In *12th International Packet Video Workshop*. PA: Pittsburgh, 9–16.

D. Gangadharan, L.T.X. Phan, S. Chakraborty, R. Zimmermann, and I. Lee. 2011. Video Quality Driven Buffer Sizing via Frame Drops. In *IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA),*, Vol. 1. 319–328.

G.R. Hiertz, L. Stibor, J. Habetha, E. Weiss, and S. Mangold. 2005. Throughput and delay performance of IEEE 802.11e Wireless Lan with Block Acknowledgments. In *Wireless Conference 2005-Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*. VDE, 1–7.

Á. Huszák and S. Imre. 2010. Analysing GOP structure and packet loss effects on error propagation in MPEG-4 video streams. In *IEEE International Symposium on Communications, Control and Signal Processing (ISCCSP),*. 1–5.

I. Inan, F. Keceli, and E. Ayanoglu. 2006. An Adaptive Multimedia QoS scheduler for 802.11e Wireless LANs. In *IEEE International Conference on Communications,*, Vol. 11. 5263–5270.

J. Klaue, B. Rathke, and A. Wolisz. 2003. Evalvid–A framework for video transmission and quality evaluation. *Computer Performance Evaluation. Modelling Techniques and Tools* (2003), 255–272.

A. Ksentini, M. Naimi, and A. Guéroui. 2006. Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture. *IEEE Communications Magazine,* 44, 1 (2006), 107–114.

T. Li, Q. Ni, and Y. Xiao. 2006. Investigation of the block ACK scheme in wireless ad hoc networks. *Wireless Communications and Mobile Computing* 6, 6 (2006), 877–888.

Richard MacKenzie, David Hands, and Timothy O'Farrell. 2009. Packet handling strategies to improve video QoS over 802.11 e WLANs. In *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications, 2009*. 1173–1177.

J. Majkowski and F.C. Palacio. 2006. Dynamic TXOP configuration for Qos enhancement in IEEE 802.11e wireless LAN. In *International Conference on Software in Telecommunications and Computer Networks,*. 66–70.

A. Moorthy, L. Choi, A. Bovik, and G. De Veciana. 2012. Video quality assessment on mobile devices: Subjective, behavioral and objective studies. (2012).

J.P. Pavon and S. Choi. 2003. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *IEEE International Conference on Communications,*, Vol. 2. 1108–1113.

A. Ranjitkar and Y.B. Ko. 2008. Performance enhancement of IEEE 802.11s mesh networks using aggressive Block ACK scheme. In *IEEE International Conference on Information Networking,*. 1–5.

P. Read and M.P. Meyer. 2000. Restoration of motion picture film. (2000).

EG Richardson. Ian, 2003, H. 264 and MPEG 4 Video Compression. (????).

P. Roshan and J. Leary. 2004. *802.11 wireless LAN fundamentals*. Cisco Systems.

S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. 2010. Design and implementation of an "approximate" communication system for wireless media applications. *Computer communication review* 40, 4 (2010), 15.

S. Sendra, M. Garcia, C. Turro, and J. Lloret. 2011. WLAN IEEE 802.11 a/b/g/n Indoor Coverage and Interference Performance Study. *International Journal On Advances in Networks and Services* 4, 1 and 2 (2011), 209–222.

James She, Fen Hou, Basem Shihada, and Pin-Han Ho. 2011. MAC-Layer Active Dropping for Real-time Video Streaming in IEEE 802.16 Networks. *GESJ: Computer Science and Telecommunications* 2 (2011), 31.

Vasilios A Siris and Costas Courcoubetis. 2006. Resource Control for the EDCA and HCCA Mechanisms in IEEE 802.11 e Networks. In *IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks,*. 1–6.

APPENDIX

A. BACKGROUD

In this appendix, we briefly review the 802.11e enhancements, their application to video streams and some video properties, which we refer to in later discussion.

Distributed Coordination Function (DCF) is the most common deployment method of Carrier Sense Multiple Access/ Collision Avoidance (CSMA/CA) in IEEE 802.11 WLANs. In DCF, a station checks whether or not the channel is busy and then it tries to transmit data after a time interval measured as DIFS (Distributed Inter Frame Space). If data is received without collision, the station sends acknowledgement (ACK) after SIFS (Short Inter Frame Space) interval to inform the data is well-received. The frame spacings are of three type DIFS, PIFS and SIFS and their values are reported in Table VI.

Unfortunately, DCF supports only best-effort service, which implies that it cannot guarantee any QoS such as minimum bandwidth, jitter, packet delay, etc. Absence of traffic differentiation finally leads to low satisfaction of end users.

A.1. EDCA - TXOP

IEEE 802.11e Task Group (TG) defines Enhanced Distributed Channel Access (EDCA) with 8 different user priority (UP) in order for stations to access differential and distributed channels. UP is defined depending on QoS priority is split into 4 Access Categories (AC) so that the stations can be serviced independently based on the AC. Table V represents the details of AC. AIFS is a interval time, which depends on the priority of AC. The AIFS is derived by $AIFSN[AC] \times slot - time + SIFS$, where AIFSN depends on AC in the Table V.

Packets to be sent are mapped to corresponding AC depending on the different QoS requirements. Higher ACed packets are served first. When the packet are ready to transmit, it follows DCF function.

In legacy DCF mode, only one MAC frame can be delivered in a single channel access, while EDCA can send more than one frames using concept of TXOP. However, each MAC frame should be ACKed by the receiver to inform sender whether each packet is well transmitted. This is referred to as Burst ACK scheme. This traditional ACKing for each transmitted frame can decrease full performance of TXOP method of EDCA.

Table V. Access Category (AC) in IEEE 802.11e)

Access Category (AC)	Type	AIFSN	Max. TXOP
Background	AC_BK	7	0
Best Effort	AC_BE	3	0
Video	AC_VI	2	3.008ms
Voice	AC_VO	2	1.504ms
Legacy DCF	-	2	0

*AIFSN : Arbitration inter-frame spacing

A.2. Block ACK

IEEE 802.11e standard defines a Block ACK (BA) scheme. In original TXOP of EDCA, a sender can send a block of frames to the same destination. However, BA allows the sender to send block of frames without receiving ACK message on each MAC frame of the block. After transmitting a block of frames, a sender delivers a block acknowledgement request (BAR) to obtain information about which frames are well received or not. The receiver responds with a block acknowledgement (BA) to deliver the answer. After receiving the response from the receiver, the sender should defer a DIFS interval time and process a back off time before sensing channel again.

If some frames are not correctly received, they will be transmitted as a part of the very next block. The receiver has to maintain a buffer to insert lost frames when they are received. If every frame in the block is received successfully, the frames transmitted are removed from the queue of

Table VI. Values of Interframe Space in IEEE 802.11)

Standard	Slot Time	DIFS(μs)	PIFS(μs)	SIFS(μs)
IEEE 802.11b	20	50	30	10
IEEE 802.11a	9	34	25	16
IEEE 802.11g	9 / 20	28 / 50	19 / 30	10
IEEE 802.11n (2.4/ 5GHz)	9 / 20	28 / 50	19 / 25 / 30	10 / 16

the sender and then a new block of frames are constructed for the next round of transmission. In order to overcome the protocol overhead from TXOP of EDCA scheme, BA can achieve a better performance in terms of throughput but it also has limitation on delay sensitive traffic like voice and video [Cabral et al. 2009; Hiertz et al. 2005; Li et al. 2006; Ranjitkar and Ko 2008]. Larger TXOP size can bring better throughput but it can also introduce a delay problem on delivery of packets between layers in a receiver side. The original EDCA scheme can process immediate response for a lost frame in bad circumstance since every ACK message is delivered to a sender on every frames in a given TXOP duration.

A.3. Frame packaging in videos

Existing video coding standards such as MPEG2, H.264 AVC (or MPEG-4 rel 10), H.264 SVC and H.265 (or MPEG HEVC) are all based on same skeleton of video coding, essentially characterized by three steps:

(1) **Temporal coding** using frame packaging, motion compensation, estimation and other techniques.
(2) **Spatial coding** using DCT, quantization and other techniques.
(3) **Entropy coding** using arithmetic or huffman coding.

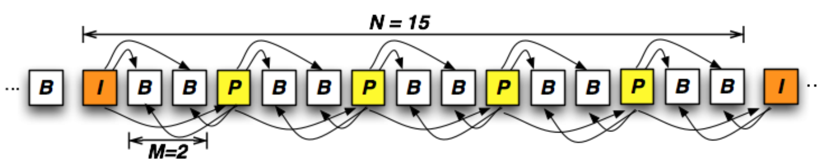


Fig. 14. Example : Pattern of GOP(N=15, M=2) in MPEG Video Stream; (15,2) is most commonly used GOP parameters.

In the frame packaging stage, video frames are classified into three types of frames: I, P and B frame. Each type of frame is encoded by different compression methods according to the dependences with other temporal video frames. The bit stream generated by video encoder consists of the fixed length-pattern of I, P and B frames, which is called as Group of Picture(GOP). From the type of GOP pattern and size, the encoding MPEG file has different properties such as file size, video quality, decoding methods, bitrate of streaming, etc.

— **I Frame** (Intracoded frame); Encoded independently of the other types of frame. I frame is referenced in encoding P and B frames.
— **P Frame** (Predictively encoded frame); P frame is encoded from information of the previous reference frame, I or another P frame.
— **B Frame** (Bi-diretional frame); Encoded from the preceding and succeeding reference frames, I or P frame.

The pattern of GOP is specified by two parameters, N and M. N denotes the total size of GOP; interval of I frame, and M presents the coding ratio of inter-frame; the number of B frames between I or P frame. The Figure 14 shows the example of a single GOP. GOP is a serial pattern of I, P and B frames. Key frame, I is the most important frame, followed by P and B in order. There is one I frame in a single GOP, whereas the number of P and B are decided by parameters, M and N. In Figure 14, there are four P frames, but they have different importance according to their each position. The P frame in beginning of GOP causes more significant distortion than others.