To appear in IEEE ICC 2008

An On-off Queue Control Mechanism for Scalable Video Streaming over the IEEE 802.11e WLAN

Yu Zhang, Chuan Heng Foh, Jianfei Cai
Centre of Multimedia and Network Technology
School of Computer Engineering
Nanyang Technological University
Singapore

Abstract— **In this paper, we study the issue of scalable video streaming over IEEE 802.11e EDCA WLANs. Our basic idea is to control the number of "active" nodes on the channel in order to reduce collisions under heavy traffic conditions. Specifically, we propose a distributed on-off queue control (OOQC) mechanism, which is designed to maintain high network throughput while keeping packet loss due to collision as low as possible. A low priority early drop (LPED) method is also employed to drop the packets at the queue according to packet relative priority index (RPI) provided by scalable video coding. Simulation results show that our proposed OOQC scheme significantly outperforms EDCA in received video quality.**

I. Introduction

IEEE 802.11 based WLANs have become one of the most widely deployed wireless networks due to its attractive cost performance ratio. With its continuously increased data rate, WLANs start to support more multimedia applications. The IEEE 802.11e standard [1] is developed to enhance the MAC mechanisms, among which a contention based enhance distributed channel access (EDCA) is developed to extend the legacy distributed coordination function (DCF). EDCA enchances the MAC layer through introducing multiple access categories (ACs) to serve different types of traffics. Although EDCA provides a powerful platform to support QoS, its parameters need to be fine tuned in order to achieve an optimal network performance.

Recently, a few schemes have been proposed for video streaming over 802.11e EDCA [2], [3], [4], [5]. The basic idea is to give different priorities to different portions of a video bitstream. In particular, Ksentini *et al.* illustrated the idea of mapping different video packets into different EDCA ACs. In our previous independent study [3], we proposed an adaptive framework for optimally mapping prioritized scalable video packets into two EDCA ACs.

Although our previously proposed system in [3] achieves excellent performance for scalable video over EDCA, it uses two EDCA ACs for transmitting video traffic, which is not standard-compliant. In this paper, we focus on using queue control method in the MAC layer to transmit scalable video over one AC (AC2) that is specified by the standard for video traffic. Our basic idea is to control the number of "active" nodes on the channel to reduce collisions under intensive competition. Specifically, we propose a distributed on-off queue control mechanism to adaptively maintain a certain number of "active" nodes on the network so that the network can be operated at high throughput without going into congestion collapse. In addition, we use the low priority early drop (LPED) algorithm to drop the packets at the queue according to packet priority index provided by scalable video coding (SVC). Simulation results show the superior performance of our proposed system.

The rest of the paper is organized as follows. Section II introduces background of our proposal. In section III, we describe our proposed queue control mechanism. Section IV presents the ns-2 simulation results for our proposed mechanism. Finally, conclusions are drawn in Section V.

II. Background

A. Overview of IEEE 802.11e EDCA

In the 802.11e standard, the EDCA mechanism extends the legacy distributed coordination function (DCF) access mechanism to enhance the QoS support in the MAC-layer through introducing multiple ACs to serve different types of traffics. In particular, EDCA defines four queues in one node, and each queue is mapped onto one AC. Each AC is designed to carry particular prioritized traffic. In the protocol, AC_VO (AC3) is designed for voice service, and AC_VI (AC2) is designed for video service, respectively. The left two ACs, AC1 and AC0 are for background service and best effort service. Each AC is assigned a certain contention window (CW), i.e. CW_{min} and CW_{max}, and arbitrary inter-frame space (AIFS) parameters. In general, larger values of CW and AIFS are assigned to a lower priority queue as these parameters lead to a less aggressive access to the common wireless channel. Each AC performs the backoff procedure individually according to its CW and AIFS parameters. If two or more ACs in one node schedule packet transmission at the same time, a virtual collision occurs. The packet with highest priority wins the transmission opportunity (TXOP).

Table I shows the IEEE 802.11e system parameters, whose constants follow the standard [1] whenever specified.

B. Scalable Video Coding

In order to enable the easy adaptation of wireless video streaming, in our previous work [6], we develop a simple scalable video coding scheme based on the integration of the motion compensated temporal filtering (MCTF) [6] and

TABLE I
IEEE 802.11E MAC PROTOCOL SYSTEM PARAMETERS

Access category	AIFSN	CW_{min}	CW_{max}	Queue length	Maximum retry limit, r
AC_VO	2	7	15	25	8
AC_VI	2	15	31	25	8
AC_BK	3	31	1023	25	4
AC_BE	7	31	1023	25	4

JPEG2000. In particular, each color component (YUV) of the original frames is first filtered using MCTF with 5/3 wavelet. MCTF is applied iteratively to the set of low-pass bands in order to provide multiple frame rates in the final scalable bitstream. Through MCTF, we generate the motion vectors and many temporal bands (T-bands). Each T-band can be treated as an individual image. Then, JPEG2000 is used to encode these T-bands into multiple quality layers, each of which has a R-D value. After removing those non-feasible truncation points, optimal bit truncation is performed to reach the given target bitrate. The final video bitstream consists of the MV information generated by MCTF and the JPEG2000 bitstream for each T-band.

III. PROPOSED QUEUE CONTROL SCHEME

In this section, we first revisit the performance influence of the number of nodes on the throughput. This study leads to our proposal of on-off queue control (OOQC) mechanism which aims to achieve high throughput performance by controlling the number of nodes accessing the common IEEE 802.11e channel. OOQC adaptively controls the source rate of SVC video and drops excessive packets from the buffer of each local node to maintain low packet loss rate and high throughput operation of the IEEE 802.11e MAC protocol. While OOQC maintains high performance of the channel, it does not address the unequal loss protection of video packets. Thus, based on the relative priority index (RPI) generated by SVC, we further introduce low priority early drop (LPED) algorithm to protect high priority video packets from dropping due to OOQC.

In our proposed scheme, we employ the 3D wavelet based scalable video codec in [6] for video encoding. The encoded video bitstream is packetized into packets with different priorities as that in [3]. Specifically, the packetization assigns a relative priority index to each packet. The lower the RPI is, the more important the packet is. If a packet with high priority is lost, it will cause significant video quality degradation to the reconstructed video.

The prioritized video packets are sent into a virtual queue (VQ) for further manipulation before passing down to the MAC layer for transmission. The VQ executes the LPED algorithm to monitor the queue length and drop low priority packets when buffer overflow is likely to occur. This early dropping protects future packet arrivals that may carry higher priority values, hence protecting them from finding a full buffer and forcing to be discarded.

Video packets in the VQ are regulated while sending to AC2. Our proposed OOQC mechanism uses "on" and "off" to shape the traffic flow passing from VQ to AC2. OOQC regulates the flow such that the contention in the IEEE 802.11e channel will be not excessive. This control allows the IEEE 802.11e channel to operate at a level close to its highest throughput. Details of our proposed system are given in the following subsections.

A. IEEE 802.11e EDCA Throughput

As our mechanism focuses on video streaming, we utilize AC2 queue in our OOQC design to achieve high throughput operation in the IEEE 802.11e channel. We recognize that the EDCA is a contention based protocol that may suffer from congestion when excessive number of nodes contend for transmissions. To understand the impact of the number of nodes on the EDCA performance, we make use of our earlier developed Markov chain model in [7] to study the EDCA throughput. Here, we provide the important results that are necessary for our discussion.

Our examination follows the model described in [7]. Let $\alpha_{i,j}$ be stationary distribution of the Markov chain at state $\{i, j\}$ (i is the backoff stage index and j is the value in the backoff counter), which can be determined by [7]

$$\alpha_{0,0} = \begin{cases} \frac{\xi}{\iota_r+\kappa}, & r \leq m \\ \frac{\xi}{\iota_m+\kappa+\nu}, & r > m \end{cases} \tag{1}$$

where ξ, ι, κ, and ν are given by

$$\begin{aligned} \xi &= 2(1-2p)(1-p) \\ \iota_r &= W_0(1-(2p)^{r+1})(1-p) \\ \iota_m &= W_0(1-(2p)^{m+1})(1-p) \\ \kappa &= (1-2p)(1-p^{r+1}) \\ \nu &= W_0 2^m p^{m+1}(1-2p)(1-p^{r-m}). \end{aligned} \tag{2}$$

The probability that a AC2 node transmits in a slot can be expressed as

$$\tau = \frac{(1-p^{r+1})\alpha_{0,0}(p)}{1-p}. \tag{3}$$

In general, τ is associated with the collision probability p, which is the collision probability given that a node transmits. Let the number of nodes accessing the network of AC2 be n, the collision probability p can be written as

$$p = 1-(1-\tau)^{n-1}. \tag{4}$$

Solving for (3)-(4) allows the study of AC2 throughput. The determination of the AC2 throughput further needs

$$P_I = (1-\tau)^n \tag{5}$$

$$P_S = n\tau(1-\tau)^{n-1} \tag{6}$$

$$P_C = 1 - P_I - P_S \tag{7}$$

where P_I, P_C, and P_S describe the probabilities that a particular slot is idle, contains a collision, and carries a successful transmission, respectively. The throughput of AC2, $U(n)$, can then be determined by

$$U(n) = \frac{P_S E[P]}{P_I\sigma + P_S T_S + (1-P_S-P_C)T_C} \tag{8}$$

where σ, $E[P]$, T_S and T_C are the duration of an idle slot, the average duration of frame payload, successful transmission

TABLE II
IEEE 802.11E TIMING CONSTANTS FOR BASIC METHOD

Description	Value
Channel bit rate	11 Mbps
Idle slot duration, σ	20 μs
Overhead of Successful transmission duration, T_S	131.8182 μs
Overhead of Collision duration, T_C	93.1818 μs

and collision, respectively. Table II describes the duration constants according to the IEEE 802.11e standard.

With the above performance measure $U(n)$, in Fig. 1, we obtain the throughput of AC2 under different node number n conditioned that there is only AC2 traffic under the parameters listed in Table I and Table II. Since we set the maximum video packet size to 500 byte, we approximate E[P] to 500 bytes. As for the other parameters, we follow the setting listed in [8]. As shown in Fig. 1, the throughput of AC2 ranges from 0.1 to 0.40 when the network size is less than 50. On the curve, a peak value appears when there are around five nodes transmitting. In other words, when the network consists of five nodes contending the common channel, the channel achieves its highest throughput.

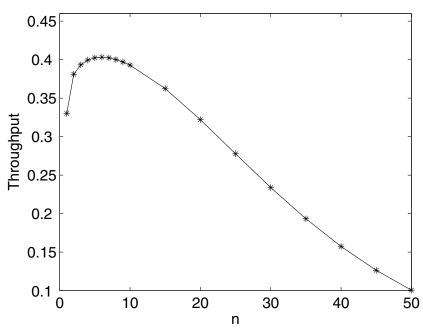


Fig. 1. The throughput in an AC2 node under various network conditions.

The main purpose of OOQC is to regulate the traffic flow from each node such that there is around five nodes contending the channel at any time of operation. It is possible to achieve that if we force some nodes into temporary "inactive" mode while maintaining five nodes "active" in the network. As we can see from Fig. 1, operating around five nodes, not necessary at exactly five nodes, is sufficient to maintain high throughput of the IEEE 802.11e MAC protocol.

B. The On-Off Queue Control Mechanism

To allow the IEEE 802.11e channel to operate at its highest throughput, it is necessary to control the number of the active nodes contending the channel. In a distributed protocol like the IEEE 802.11e MAC protocol, the information of the number of nodes in a network is not readily obtainable. In the literature, Bianchi *et al.* [9] has proposed a method to estimate the number of nodes. Knowing traffic of video streams appears continuously in time, we here seek a simpler method to obtain this information. Each node periodically detects traffic associated with AC_VI, and makes use of the observed MAC addresses to tell the number of simultaneous video streaming nodes. This observation may be easily done by using the NAV information. We then use this observation to control the number of active nodes in the network.

The regulation of traffic flow onto the IEEE 802.11e channel is necessary when the traffic load is relatively high. Our proposed control uses "on" and "off" to regulate the packet flow from the buffer of a local node onto the common channel. Each node consists of a virtual queue (VQ) that holds video packets, which prevents video packets from entering unrestrictedly to the AC2 queue in the IEEE 802.11e MAC layer for transmissions. Associating with the VQ is a sleep counter that forces a node to the "off" state for certain duration. The operation of our proposed OOQC is described as follows.

During the "on" state, each node passes one video packet to the AC2 queue for transmission. When the packet is successfully transmitted, the node is forced to enter the "off" state and the sleep counter is set to S where S is a constant protocol parameter in OOQC. The counter is decremented each time when the node detects a successful transmission from AC2 on the channel. When the counter reaches zero, the node returns to the "on" state to attempt for another video packet transmission and attend the contention again.

The design for S is based on the following analysis. For the OOQC procedure, we first derive a compact Markov chain model in Fig. 2. Considering to control the number of "on" state nodes in the WLAN, the size of S is important to exploit. We study the behavior of a single node with Markov model. We decrease the sleep counter at the end of one successful transmission. The slot time here for the Markov chain is the time interval between two consecutive successful transmission ends. Let δ_i be stationary distribution of the Markov chain at state i, which can be determined by

$$\delta_0 = \frac{1}{1 + \gamma S} \tag{9}$$

where γ is the probability that one node successfully transmits and S is the sleep counter of node.

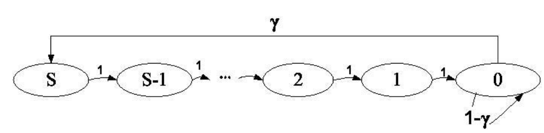


Fig. 2. Markov Chain Model for OOQC.

The quantity δ_0 is associated with the transmitting probability γ which can be written as

$$\gamma = \sum_{i=1}^{T} \left[\frac{\binom{T}{i} \delta_0^i (1-\delta_0)^{T-i}}{1-(1-\delta_0)^T} \cdot \frac{1}{i} \right] \tag{10}$$

where T is the total number of nodes.

Our proposal tries to maintain a certain number of active nodes on the wireless channel, given the total number of nodes to be T. The expectation of the number of "active" nodes, n,

TABLE III

DESCRIPTION OF ON-OFF QUEUE CONTROL MECHANISM

```
Let NIDG denote the set of sensed node id.
Let S denote the sleep counter which the node should experience.
Let n̂ denote the target network size which is 5 in our case.

procedure SuccessfulPacketTransmissionEvent(id)
  // id is the node id of which transmits successfully.
  // i is the node id of all nodes.
  // T is the node number of which are transmitting.
  // M is the maximum supported video bandwidth, we use 3.6Mbps.
  // counter(i) is the sleep counter of node i.
  // R_s is the source rate of one node.
  if id not in NIDG
    T++;
    S = 0.8(T - n̂);
    R_s = Min{Max{M/T, 200kbps}, 800kbps};
    NIDG = NIDG ∪ {id}
  else
   for i in NIDG
    if i ≠ id
     counter(i)--;
    else
     counter(i) = S;
    end if
   end for
  end if
end
```

can be expressed as

$$\Sigma_{i=1}^{T}\left[\frac{\binom{T}{i}\delta_0^i(1-\delta_0)^{T-i}}{1-(1-\delta_0)^T}\cdot i\right] = n \qquad (11)$$

Consequently, the procedure of our OOQC exploits the above results for the sleep counter setting. Equations (9)-(11) form a non-linear system which can be solved numerically. Our particular interest is S which is the sleep counter of one node, given the target of n=5. The solution is illustrated in Fig. 3, where we plot S versus T. Interestingly, we see that the results can be described by a simple function of $S = 0.8(T - 5)$. For implementation simplicity, we use $S = 0.8(T - 5)$ in our simulation.

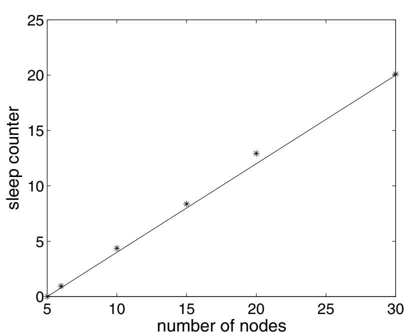


Fig. 3. Sleep counter versus the number of nodes.

The detailed mechanism description is given in Table III.

C. The Low Priority Early Drop Algorithm

OOQC regulates the traffic flow from the VQ to the AC2 queue. The mechanism tries to maintain the highest throughput by mitigating the heavy traffic condition of the IEEE 802.11e channel in order to reduce the channel collision. The cost of this regulation is the packet drops at the VQ. We should make use of source rate control method to reduce packet drop. In TABLE III, we update the source rate R_s of each video node, once a new node is introduced. If we cannot sense one nodes's packet for two updating interval, we consider the node is missing. Then, we reduce T by 1 and update R_s again.

VQ overflow is still inevitable despite using source rate control. As a packet drop is necessary, the packet dropping should be selective rather than random. We exploit the RPI values provided by SVC for prioritized packet dropping. The packet drop occurs when video packets passing down from the application layer find a full buffer at the VQ. Without a control, video packets that are of high priorities may be blocked from entering the VQ. To prevent this from occurring, we introduce the low priority early drop (LPED) algorithm to realize the unequal error protection for prioritized video packets. LPED constantly sorts the packets in VQ according to the RPI. When one incoming packet finds a full buffer, the video packet with lowest priority is dropped to give space for high priority packet.

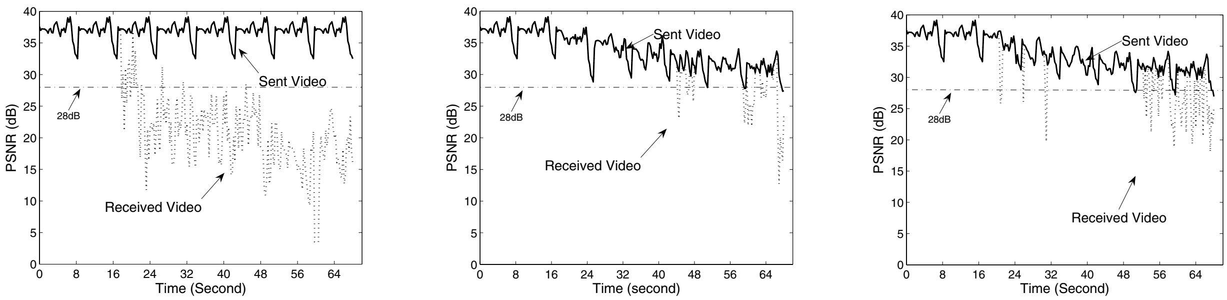
IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed OOQC mechanism through ns-2 simulation with EDCA components [10]. Our simulation uses the environment setting listed in TABLE I & II. The simulation scenario considers an infrastructure 802.11e EDCA wireless network with one access point (AP). Channel data rate is set to 11 Mbps. Each node has one video clip to transmit over the EDCA AC2. The receivers are outside of the WLAN and receives the video via the AP in the WLAN. To simplify the experiments, we measure the video quality at the AP. We use the first 256 frames of the "foreman" CIF sequence as our video source, which is encoded by the 3D wavelet based SVC [6]. The sequence is repeatedly transmitted to have a long time simulation. The maximum packet size is set to 500 bytes. The RPI value ranges from 0 to 63. As the GOP of our video source is 8 frames and the video sequence displays at 30 fps, 8/30 seconds is the finest scale at which the source rate can be updated.

We compare the performance of three mechanisms: EDCA, our proposed OOQC without LPED, and our proposed OOQC with LPED. For our proposed OOQC, the video source rate can be adapted ranging from 200 kbps to 800 kbps, while the initial source rate is fixed to 800 kbps. We take the PSNR of Y component as our video quality measurement. For comparison, the sent video quality is used as a benchmark.

In simulation, At time zero, the first node is added onto the EDCA channel. After that, each new node is added every four seconds until the number of nodes reaches 15.

Fig. 4(a) shows the PSNR performance using EDCA, where the received video quality is measured at node 0. We see that the PSNR dropped below 20 dB after 5th node joining the



(a) (b) (c)

Fig. 4. PSNR performance of received video for (a) EDCA, (b) OOQC with LPED, and (c) OOQC without LPED.

Fig. 5. Sample frame received with OOQC (left) vs. Sample frame received with EDCA (right).

network. This indicates that EDCA cannot support beyond five video nodes running 800 kbps.

Fig. 4(b) shows the performance of our proposed OOQC mechanism with LPED. Our proposed OOQC mechanism with LPED supports up to 14 streaming nodes with over 28 dB for the received PSNR. A small PSNR drop appears when the 12th node joining the network, however, our algorithm recovers the quality after several frame transmissions.

To test also the importance of LPED in performance, we show the PSNR performance of our proposed OOQC mechanism without LPED in Fig. 4(c). It can be observed that the PSNR suffers a drop even when the 6th streaming node joining the network. LPED ensures that high priority packets are protected to avoid significant video quality degradation.

Fig. 5 illustrates the visual quality of a particular received video frame using our proposed OOQC scheme and EDCA. We can see that our proposed OOQC achieves much better visual quality than using EDCA.

V. Conclusion

In this paper, we proposed an on-off queue control mechanism, which works with the source rate adjustment and the lower priority early drop algorithm to achieve better video streaming support. The OOQC utilizes a VQ to control the "active" video node number to around five, which achieves the highest throughput in EDCA in our studied scenario. We employed source rate adjustment and LPED to further avoid overflowing of packets onto the channel. Simulation results showed that our proposed OOQC scheme outperforms EDCA in received video quality.

References

[1] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Method MAC) Quality of Service Enhancements*, IEEE-SA Standards Board Std. 802.11e, Nov 2005.

[2] A. Ksentini, M. Naimi, and A. Gueroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Communications Magazine*, pp. 107–114, Jan. 2006.

[3] C. H. Foh, Y. Zhang, Z. Ni, and J. Cai, "Scalable video transmission over the IEEE 802.11e networks using cross-layer rate control," in *Proc. IEEE ICC'07*, Glasgow, UK, June 2007.

[4] M. van der schaar, Y. Andreopoulos, and Z. Hu, "Optimized schalable video streaming over IEEE 802.11a/e HCCA wireless networks under delay constraints," *IEEE transaction on mobile computing*, vol. 5, no. 6, pp. 755–768, JUNE 2006.

[5] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, "Feedback-based control for providing real-time services wiht the 802.11e MAC," *IEEE/ACM transactions on networking*, vol. 15, no. 2, pp. 323–333, APRIL 2007.

[6] Z. Ni and J. Cai, "Constant quality aimed bit allocation for 3D wavelet based video coding," in *Proc. IEEE ICME'06*, Toronto, ON, Canada, July,2006, pp. 1657–1660.

[7] J. W. Tantra, C. H. Foh, and A. B. Mnaouer, "Throughput and delay analysis of the IEEE 802.11e EDCA saturation," in *Proc. IEEE International Conference on Communications 2005*, Seoul, Korea, May 2005.

[8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[9] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *INFOCOM*, vol. 2, March-April 2003, pp. 844–852.

[10] $http://www.tkn.tuberlin.de/research/802.11e_ns2/$.