electronics

MDPI

Article

A Novel QoS-Aware A-MPDU Aggregation Scheduler for Unsaturated IEEE802.11n/ac WLANs

Cong Lu [1,2], Bin Wu [1,*] and Tianchun Ye [1]

[1] Intelligent Manufacturing Electronics R&D Center, Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China; lucong@ime.ac.cn (C.L.); tcye@ime.ac.cn (T.Y.)

[2] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: wubin@ime.ac.cn; Tel.: +86-8299-5566

Received: 18 June 2020; Accepted: 24 July 2020; Published: 27 July 2020

Abstract: Improving the quality of service (QoS) performance to support existing and upcoming real-time applications is critical for IEEE 802.11n/ac devices. The mechanisms of the media access control (MAC) layer, including the aggregate MAC protocol data unit (A-MPDU) aggregation, greatly affect the QoS performance in wireless local area networks (WLANs). To investigate the impact of the aggregation level on the QoS performance for real-time multimedia applications, a novel end-to-end delay model for the unsaturated settings is proposed in this paper. The presented model considers the gathering procedure of packets, queuing behaviors, and transmissions using the RTS/CTS (request to send/clear to send) mechanism on error-prone channels. Based on the model, a novel QoS-aware A-MPDU aggregation scheduler for IEEE802.11n/ac WLANs was shown to obtain better QoS performance with lower latency and less packet loss, a larger capacity to hold higher data rates, and more working nodes. The validation of the proposed model and the promotion of the proposed scheduler are well benchmarked by ns-3.

Keywords: IEEE802.11n/ac; A-MPDU; aggregation level; QoS-aware; end-to-end delay; unsaturated; collision

1. Introduction

Currently, IEEE 802.11n/ac wireless local area networks (WLANs) are widely used to bear Internet traffic for real-time multimedia applications, such as live video sharing, voice over internet protocol (IP), and online games on electronic devices. These real-time multimedia applications necessitate a stringent level of quality of service (QoS) performance, including low end-to-end delay and small packet loss rates. According to [1], the average end-to-end delay should be less than 100 ms, and the packet loss rate should be less than 0.1% for real-time applications. Emerging applications, such as virtual reality/augmented reality, industrial surveillance, and cloud computing require even tighter QoS constraints. Therefore, it is critical to improve the QoS performance in WLANs to support existing and upcoming real-time applications.

The scheduling mechanism of the media access control (MAC) layer greatly affects the QoS performance in WLANs. In IEEE 802.11n/ac WLANs, the performance of the MAC layer is enhanced via the aggregate MAC protocol data unit (A-MPDU) aggregation and block acknowledgment (BlockACK) [2]. When a new aggregation procedure is launched, up to 64 cached packets are aggregated into a new A-MPDU. For real-time voice and video traffic, because it is not guaranteed that there are adequate packets in the buffer all the time (i.e., the WLANs work under unsaturated conditions), the timely processing of cached packets and the construction of larger A-MPDU frames are contradictory. If more packets are to be aggregated, the cached packets should wait for new packets to arrive,

which increases the waiting time of existing packets [3]. On the contrary, lower aggregation levels require more frequent attempts to transmit packets, which increases the collisions and results in a longer delivering time together with the queuing time, as described in [4]. Consequently, to obtain the optimal aggregation level, i.e., the right number of aggregated packets and the best QoS performance, it is challenging but meaningful to compromise among the additional waiting time, improved collisions, and MAC efficiency.

To investigate the performance of real-time multimedia applications in IEEE802.11n/ac WLANs with A-MPDU aggregation, we proposed a novel end-to-end delay model. The proposed model considered the gathering procedure, queuing behaviors, and transmissions using the RTS/CTS (request to send/clear to send) mechanism on error-prone channels under the unsaturated settings. Based on the proposed model, we present a novel QoS-aware A-MPDU aggregation scheduler for IEEE802.11n/ac WLANs to determine the optimal aggregation level, with which better QoS performance and enhanced system capacity were achieved. The major contributions of the proposed scheduler are listed below:

- The model is based on the aggregation level-dependent collision probability considering the gathering procedure.
- The average and variance of the access delay for transmissions using RTS/CTS mechanisms on error-prone channels are discussed and the effect on the queuing behaviors is given.
- A model for the end-to-end delay, gathering delay, queuing delay, and collisions is developed.
- An algorithm that can search for the optimal aggregation level more quickly by narrowing the candidate range of aggregation levels is proposed.

The remainder of the paper is organized as follows. In Section 2, the background and related work are introduced. In Section 3, we discuss the analytical model to evaluate the end-to-end delay with the aggregation level. In Section 4, we present the QoS-aware A-MPDU aggregation scheduler for IEEE802.11n/ac WLANs. In Section 5, we illustrate the validation of the proposed model by comparing the analytical and simulation results and the progress of the proposed scheduler through comparison with other schedulers. We conclude the paper in Section 6.

2. Background and Related Works

2.1. The A-MPDU Aggregation and Black Acknowledgement

The format of the A-MPDU frame is shown in Figure 1. The A-MPDU aggregation mechanism obtains multiple MSDU frames from the upper layer first. Then, the MPDU delimiter, MPDU header, and frame check sequence (FCS) fields are appended to each MSDU frame to form an MPDU sub-frame. Finally, these sub-frames are aggregated to form an A-MPDU frame and the end identifier (end of file, EOF) is added to the end of the A-MPDU frame. Due to the use of greater bandwidth and the multi-input multi-output (MIMO) technique in 802.11n/ac WLANs, the maximum aggregation level is usually determined by the size of the aggregation window, W. From the perspective of the recipient, the processing of each sub-frame is performed independently. The position of each sub-frame can be located by the MPDU delimiter, and the FCS field can be used to confirm whether there is a transmission error in each sub-frame.

The format of the BlockACK frame is shown in Figure 2. The BlockACK mechanism also supports the separate confirmation of sub-frames in the A-MPDU frame using the bitmap field. Due to the existence of channel errors, some sub-frames in the A-MPDU may be transmitted incorrectly. After receiving the A-MPDU frame, the recipient will send a BlockACK frame if there is at least one correctly-received sub-frame. Then, the originator only retransmits the unacknowledged sub-frames.

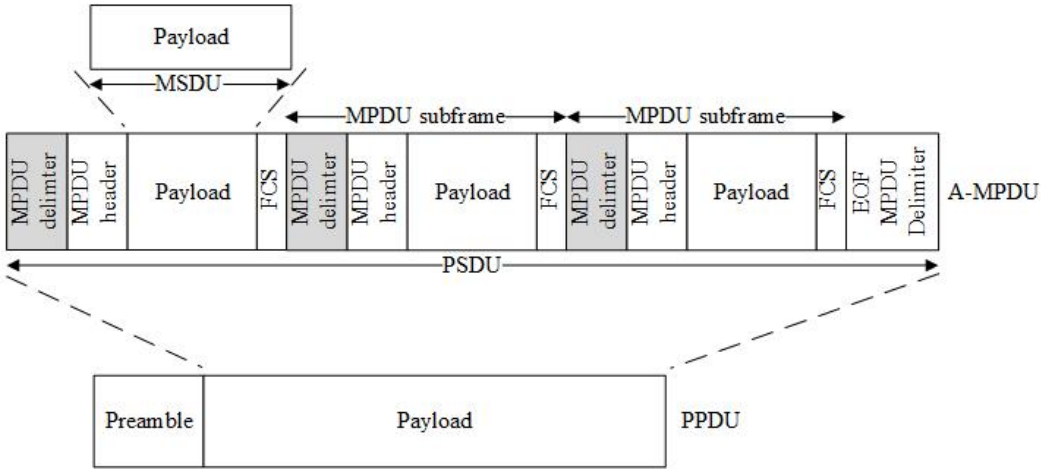

Figure 1. The aggregate media access control (MAC) protocol data unit (A-MPDU) aggregation mechanism.

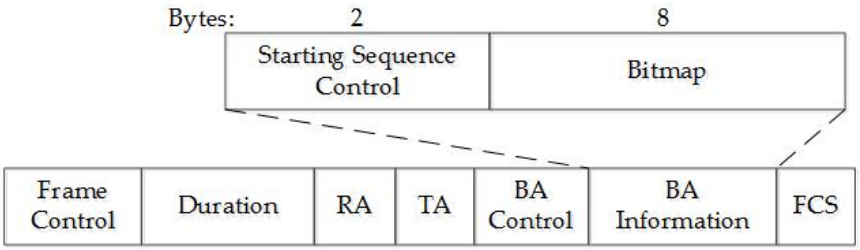

Figure 2. Frame format of block acknowledgment (BlockACK; BA).

2.2. The RTS/CTS Mechanism

The RTS/CTS is a mechanism to protect data packets from collisions. Before transmitting the A-MPDU frame, the originator sends an RTS frame to request the transmission, and the recipient replies with a CTS frame if it receives the RTS frame correctly. Since RTS and CTS frames include a time field and indicate the duration of the current transmission to all neighboring nodes receiving them, these nodes do not access the media during the duration of this transmission. Therefore, no conflicts occur during the transmission of A-MPDU frames. If a collision occurs during the RTS frame transmission, the recipient cannot receive the correct RTS frame and will not reply with the CTS frame. After waiting for the CTS frame time-out, the originator will restart the RTS/CTS operation.

2.3. Related Works

For IEEE802.11 WLANs, extensive published papers have improved QoS performance by modifying the media access parameters, such as the arbitration inter-frame space number (AIFSN), transmission opportunity (TXOP) limits, and contention window (CW) size. In [5], the proposed dynamic AIFSN adaptation scheme modified the AIFSN values based on the current network conditions to enhance the performance of voice and video. In [6], an immediate dynamic TXOP hybrid coordinator function controlled channel access (HCCA) plus scheduling algorithm was proposed to achieve the improved network performance by monitoring the transmission duration. In [7], the QoS performance was improved by adaptively scaling the CW size according to the number of active stations in each access category (AC). Zawia et al. [8] researched the performances of different IEEE 802.11 access schemes using the dynamic contention windows approach. Tian et al. [9] established a theoretical model that described a deadline-constrained MAC protocol with QoS differentiation and evaluated the QoS performance.

Various published papers also analyzed and improved QoS performances with aggregation in IEEE802.11n/ac. Many of the proposed aggregation schedulers focused on achieving maximum

throughput while meeting the QoS constraints. The A-MPDU aggregation scheduler for voice traffic introduced in [4] decided the aggregation size based on the timing report measured by the real-time transport protocol (RTP)/real-time transport control protocol (RTCP) layer and the timing information of previous transmissions to obtain a larger throughput with the QoS constraints fulfilled. In [10], the proposed retransmission scheme maximized the throughput by fully utilizing the aggregation levels that met the end-to-end delay constraints based on the end-to-end delay model extended from the Bianchi model. Lee et al. [11] calculated the mean delay of the MPDUs for each queue under a saturated condition and obtained the optimal aggregation level to enhance the throughput with the delay constraints satisfied.

Certain aggregation schedulers constructed A-MPDU frames based on the urgency of the packets reaching the buffer header to reduce the latency for multimedia applications. In [3], the influence of the frame aggregation was investigated, and an aggregation scheduler was proposed based on the investigation. For voice and video applications, the proposed scheduler transmitted all cached packets with the same AC with or without aggregation once the medium was obtained. In [12], a dynamic frame aggregation scheduler was proposed to construct the aggregated frame by selecting packets from different ACs with the lowest urgency delay to provide QoS satisfaction to real-time services.

Some existing theoretical models analyzed the performances of IEEE802.11n/ac WLANs with A-MPDU aggregation. In [13], a two-dimensional Markov chain model was provided for studying the saturated throughput performances with the packet loss caused by the channel errors. Mansour et al. [14] expounded an analysis model that conformed to the existing A-MPDU retransmission scheme to analyze the impacts of A-MPDU and BlockAck technology on saturation throughput performance. Seytnazarov et al. [15] proposed a Markov chain model for the aggregation size to evaluate the influence of the BlockACK window operation under error-prone channels and derived the throughput and the access delay for the saturated networks. Karmakar et al. [16] introduced a theoretical model for unsaturated networks to analyze the impacts of channel errors on throughput and channel access delay. In that model, each MPDU in an A-MPDU was treated separately in terms of the transmitting error and collision without considering the RTS/CTS mechanism. In [17], an unsaturated analytical model for the throughput and end-to-end delay in the error-free channel was proposed. The similarity of these unsaturated models is that their collision probabilities were estimated based on whether the buffer was empty without considering the aggregation level.

Although all the above proposals analyzed or improved the QoS performance from different perspectives, none of them discussed the relationship between aggregation levels and the collision probability for real-time multimedia applications, analyzed the queuing behaviors under error-prone channels, or compromised between the timely processing and the reduction of collisions when A-MPDU aggregation was adopted. To overcome those problems, we propose a novel end-to-end delay model that considers the gathering procedure of packets, queuing behaviors, and transmission using the RTS/CTS mechanism on error-prone channels under the unsaturated settings. Based on the proposed model, a QoS-aware A-MPDU aggregation scheduler for IEEE802.11n/ac WLANs is presented to achieve better QoS performance and enhanced system capacity.

3. Methods

First, some terminology and assumptions used in the following parts are introduced: The end-to-end delay is defined as the time interval between when a packet arrives at the buffer of AC and when it is given to the upper layer by the recipient in the WLAN. The grouped A-MPDU is defined as the A-MPDU for the initial transmission of the access procedure and determines the cached packets served in each access procedure. To simplify the analysis, we assume here that: (1) there are N stations that transmit multimedia traffic with the same average arrival rate, average packet size, and AC; (2) under the guarantee of the back-off mechanism, the collision probability of each station is considered to be constant and identical by statistical average; its effect on simplifying the modeling and

calculation has been examined in [18]; and (3) the buffer size of each AC is considered to be enough because the load of the unsaturated network is light.

3.1. The End-to-End Delay

The end-to-end delay is divided into three parts, as Figure 3 shows. The first stage is the gathering procedure, where the cached packets wait in the buffer until the number reaches the given aggregation level. The second part is the queuing procedure, where the cached packets are aggregated into the grouped A-MPDUs and these A-MPDUs are inserted into the TxQueue while waiting to access the media. The gathering procedure is different from the queuing procedure because the waiting packets are absent, even though the TxQueue is empty. The third one is the access procedure, where all the sub-frames of the fetched grouped A-MPDU are delivered to the recipient. Therefore, the end-to-end delay, D_{e2e}, can be calculated by

$$D_{e2e} = D_{gather} + D_{queue} + D_{access}, \tag{1}$$

where D_{gather} represents the time between the arrival instant of a packet and the arrival instant of the final packet in the grouped A-MPDU frame, D_{queue} means the queuing delay in the TxQueue, and D_{access} is the delay of delivering all the subframes in the grouped A-MPDU frame.

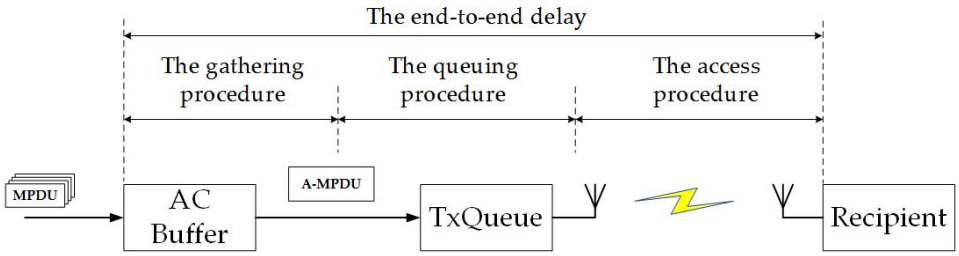

Figure 3. The constructions of the end-to-end delay.

When the aggregation level is set to L and the average arrival rate of the packets is λ, the average of D_{gather} is calculated by

$$E[D_{gather}] = \frac{L-1}{2\lambda}. \tag{2}$$

A video stream is usually characterized as periodically-delivered packet clusters. However, the arrival behavior of the grouped A-MPDU is not equivalent to that of the video stream. A grouped A-MPDU is generated when the number of cached packets exceeds the set aggregation level. Thus, the randomness of the packet quantity in each cluster results in the random arrival of grouped A-MPDUs. Moreover, when video streams from different sources are delivered simultaneously in one station, the packet cluster is not ensured to arrive periodically. Therefore, considering a general case, it is more appropriate that the arrival model of the grouped A-MPDU is assumed by the Poisson arrivals.

According to [19], the queuing behavior of the grouped A-MPDU is modeled by the M/G/1 queue. The arrival model of the grouped A-MPDU frame is described by parameter $\frac{\lambda}{L}$ and the service time is characterized by the average and the variance of the access delay, D_{access}.

According to the Pollaczek–Khinchin formula [20], the mean value of D_{queue} is calculated by

$$E[D_{queue}] = \frac{\lambda \cdot (var[D_{access}] + (E[D_{access}])^2)}{2(L - E[D_{access}] \cdot \lambda)}. \tag{3}$$

3.2. The Access Delay

Prior to analyzing the access delay, some characteristics of transmissions using the RTS/CTS mechanism under the error-prone channels were derived. The packet error rate (PER) in the access procedure is represented by e.

Figure 4 shows an example of the access procedure. When the transmission of an A-MPDU is finished, errors may occur in some sub-frames due to the channel error. Those failed sub-frames indicated by the BlockAck frame are aggregated into a new A-MPDU frame and retransmitted again. Consequently, during the access procedure, multiple A-MPDU retransmissions may be required, and the number of sub-frames in each A-MPDU retransmission is varied. We define the retransmission stage as the order of A-MPDU retransmissions in the access procedure.

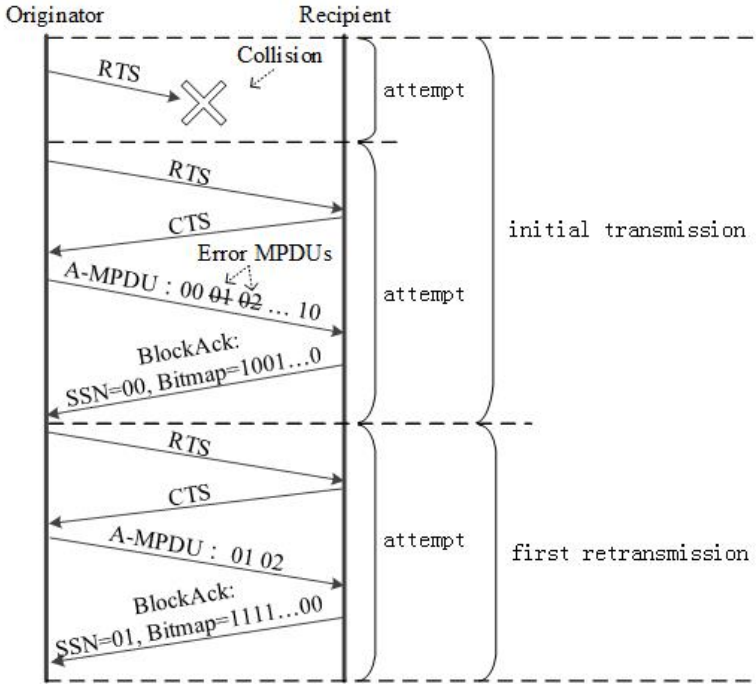

Figure 4. An example of the access procedure under the error prone channels. RTS/CTS (request to send/clear to send).

When the number of sub-frames in each retransmission stage is considered, the state diagram is shown as Figure 5. The state ℓ means there are ℓ subframes scheduled to be transmitted in the retransmission stage. Let $h_{i,j}$ be the transition probability that there are i sub-frames to be retransmitted in the next retransmission stage, after the transmission of an A-MPDU frame containing j sub-frames is finished.

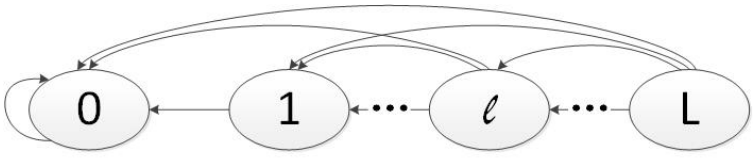

Figure 5. The state diagram for the number of sub-frames in each retransmission stage.

The distribution of the number of sub-frames in the s-th retransmission stage is calculated by:

$$\vec{\alpha}_s(e) = (\alpha_{s,0}(e) \ \ldots \ \alpha_{s,L-1}(e) \ \alpha_{s,L}(e))^T = \begin{cases} \mathbf{H}^s \cdot \vec{\alpha}_0 & s > 0, s \in \mathbf{Z}, \\ (0 \ \ldots \ 0 \ 1)^T & s = 0, \end{cases} \tag{4}$$

where **H** is the one-step transition probability matrix, which is defined as follows:

$$\begin{aligned} \mathbf{H} &= \{h_{i,j}\}, for \ \ i \in [0, L], j \in [0, L], \\ h_{i,j} &= \begin{cases} 1 & i = 0, j = 0, \\ \frac{\binom{j}{i}(1-e)^{j-i} \cdot e^i}{1-e^j}, & j > i, \\ 0 & j \leq i. \end{cases} \end{aligned} \tag{5}$$

Let $S(e)$ be the maximum retransmission stage that is equal to s whose $\alpha_{s,0}(e)$ is asymptotically equal to 1.

To characterize the retransmission stage of an arbitrary A-MPDU transmission where the PER is e, a discrete-time Markov model is estimated. Let s(t) represent the retransmission stage corresponding to the A-MPDU frame transmitted at time t. The state diagram for the model is shown in Figure 6, where the state "s" represents the s-th retransmission of the access procedure and the state "0" indicates that the initial transmission is the initial state. Let $q_{i,j}$ be the transition probability from the state j to the state i.

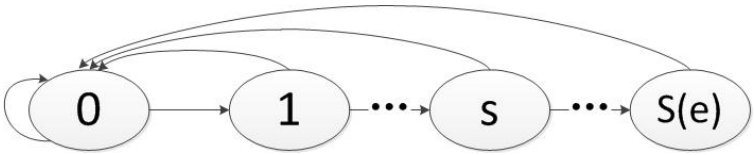

Figure 6. The state diagram for the retransmission stage corresponding to the A-MPDU frame transmitted at time t.

The stationary distribution can be solved by the following equation:

$$(\pi_0(e) \ \pi_1(e) \ \ldots \ \pi_s(e) \ \ \pi_{S(e)}(e))^T = \lim_{m \to \infty} \mathbf{Q}^m \cdot (1 \ 0 \ \ldots \ 0 \ 0)^T, \tag{6}$$

where **Q** is the one-step transition probability matrix, which is defined as follows:

$$\begin{aligned} \mathbf{Q} &= \{q_{i,j}\}, for \ \ i \in [1, S(e)], j \in [1, S(e)], \\ q_{i,j} &= \begin{cases} \alpha_{j,0}(e), & i = 0, \\ 1 - \alpha_{j,0}(e), & i = j + 1, \\ 0 & others. \end{cases} \end{aligned} \tag{7}$$

Therefore, the stationary distribution of the number of sub-frames in an arbitrary A-MPDU frame is obtained by:

$$\begin{aligned} \vec{\alpha}_\infty(e) &= (\alpha_{\infty,1}(e) \ \ldots \ \alpha_{\infty,L-1}(e) \ \alpha_{\infty,L}(e))^T, \\ \alpha_{\infty,i}(e) &= \begin{cases} \sum_{s=1}^{S(e)} \pi_s(e) \frac{\alpha_{s,i}(e)}{1-\alpha_{s,0}(e)}, & i \neq L, \\ \pi_0(e), & i = L. \end{cases} \end{aligned} \tag{8}$$

Because stations in the WLAN face different channel conditions, the bit error rate (BER) which reflects the characteristics of the physical channels is measured individually in each station. The PER of the n-th station, e_n, is derived by

$$e_n = 1 - (1 - ber_n)^{(len_{MACheader} + len_{payload})}, \tag{9}$$

where $len_{MACheader}$ is the length of the MPDU header and $len_{payload}$ is the length of the payload. The ber_n is the BER measured in the n-th station.

To obtain a universal description of characteristics of transmissions in the WLAN, the general distribution of the number of sub-frames in the s-th retransmission stage is calculated by:

$$\vec{\alpha^*}_s = (\alpha^*_{s,1} \ \dots \ \alpha^*_{s,L-1} \ \alpha^*_{s,L})^T = \frac{1}{N}\sum_{n=1}^{N} \vec{\alpha}_s(e_n), s = [1, S], \tag{10}$$

where $S = max[S(e_n)]$ is the maximum retransmission stage in the WLAN.

The general distribution of the number of sub-frames in an arbitrary A-MPDU frame in the WLAN is calculated by:

$$\vec{\alpha^*}_\infty = (\alpha^*_{\infty,1} \ \dots \ \alpha^*_{\infty,L-1} \ \alpha^*_{\infty,L})^T = \frac{1}{N}\sum_{n=1}^{N} \vec{\alpha}_\infty(e_n); \tag{11}$$

P_e is the average PER of the WLAN, which is calculated by

$$P_e = \frac{1}{N}\sum_{n=1}^{N} e_n, \tag{12}$$

In each attempt of transmitting an A-MPDU, the attempt fails if a collision is encountered during the transmission of the RTS frame or if all the sub-frames of the A-MPDU are lost due to channel error. In this situation, the originator doubles the CW size when the maximum window size is not reached and resumes attempts until an attempt of the A-MPDU is successful or the retry limit is reached. When the transmission of an A-MPDU is finished, the CW size is reset to the minimum window size.

The attempt of an A-MPDU is said to be successful when the BlockACK/ACK frame corresponding to the A-MPDU is received by the originator. Thus, the backoff probability and the successfully-transmitted probability of each attempt are defined as below, respectively:

$$p_{bo,l} = (1-\gamma)P_e^l + \gamma, \tag{13}$$

$$p_{st,l} = (1-\gamma)(1-P_e^l), \tag{14}$$

where ℓ is the number of sub-frames in the A-MPDU, and γ is the general collision probability.

The Zhao's model for an unsaturated IEEE 802.11 DCF network [21] is extended to analyze the back-off procedure for the multimedia traffic. The core vision of the model is to use a scaled version of the saturated attempt rate to approximate the unsaturated one. The main expansion here is to consider the impacts of aggregation and transmission under the error-prone channels. According to [21], the average attempt rate per slot for each node when the TxQueue is not empty is defined as:

$$\beta_c = \frac{R(\gamma)}{X(\gamma)}, \tag{15}$$

where $R(\gamma)$ and $X(\gamma)$ are redefined as the average quantity of attempts and the average backoff time (in slots) in the access procedure. As the retransmission stage in the access procedure varies from 0 to S, $R(\gamma)$ and $X(\gamma)$ are calculated by:

$$R(\gamma) = \sum_{s=0}^{S}\sum_{l=1}^{L} \alpha^*_{s,l} \cdot R_l, \tag{16}$$

$$X(\gamma) = \sum_{s=0}^{S}\sum_{l=1}^{L} \alpha^*_{s,l} \cdot X_l, \tag{17}$$

where R_l and X_l are the average quantity of attempts and the average back-off time (in slots) in the transmission of A-MPDU frame containing ℓ sub-frames. Therefore, R_l and X_l are calculated by:

$$R_l = p_{bo,l}^K \cdot K + \sum_{k=1}^{K} p_{bo,l}^{k-1} \cdot p_{st,l} \cdot k, \tag{18}$$

$$X_l = p_{bo,l}^K \sum_{u=1}^{K} b_u + \sum_{k=1}^{K} p_{bo,l}^{k-1} \cdot p_{st,l} \cdot \sum_{u=1}^{k} E[b_u], \tag{19}$$

where b_u is the back-off time (in slots) at the u-th back-off stage ($1 \leq u \leq K$), and K is the retry limit of one transmission.

Considering the impact of the aggregation level, the attempt rate is scaled by the probability that the TxQueue is not empty, pa, which is calculated by:

$$pa = \min(1, \frac{\lambda}{L} \cdot X(\gamma) \cdot T_C), \tag{20}$$

where T_C is the average time for each reduction of the back-off counter, which is given by

$$T_C = \sigma + p_{bs}(1 - p_{tr})T_{cl} + p_{bs} \cdot p_{tr} \sum_{l=1}^{L} \alpha_{\infty,l}^{*}((1 - P_e^l)T_{sc,l} + P_e^l \cdot T_{ls,l}). \tag{21}$$

The parameters are defined as follows: σ is the duration of one slot time. $p_{bs} = 1 - (1 - \gamma)^{\frac{N}{N-1}}$ is the probability that the slot is busy. $p_{tr} = N(1 - \frac{\gamma}{p_{bs}})$ is the probability that only one A-MPDU frame is transmitted among the contending nodes on the condition that the slot is busy. T_{cl} is the collision duration when the RTS/CTS mechanism is enabled. $T_{ls,l}$ is the duration when all ℓ sub-frames transmitted have failed. $T_{sc,l}$ is the duration when the transmission of the A-MPDU frame containing ℓ sub-frames is successful.

With the assumption that the buffer size is infinite, the general attempt rate β is acquired by:

$$\beta = pa \cdot \beta_c. \tag{22}$$

The relationship between the general collision probability and the general attempt rate is indicated by

$$\gamma = 1 - (1 - \beta)^{N-1}. \tag{23}$$

Consequently, the general collision probability can be solved and the relationship between the aggregation level and the collision probability is constructed.

Finally, because the only difference between the saturated and the unsaturated cases is characterized by the attempt rate, we can analyze the access delay based on [22]. Our contribution is that the following analysis covers A-MPDU aggregation and transmission under the error-prone channels.

$$\begin{aligned}
&\eta = (N-1)\beta(1-\beta)^{N-2},\\
&\theta_2 = (\gamma - \eta)T_{cl} + \eta \textstyle\sum_{l=1}^{L} \alpha_{\infty,l}^{*}((1 - P_e^l)T_{sc,l} + P_e^l \cdot T_{ls,l}),\\
&\theta_1 = \sigma + \theta_2,\\
&\theta_3 = (\gamma - \eta)(T_{cl} - \theta_2)^2 + \eta \textstyle\sum_{l=1}^{L} \alpha_{\infty,l}^{*}((1 - P_e^l)(T_{sc,l} - \theta_2)^2 + P_e^l(T_{ls,l} - \theta_2)^2).
\end{aligned}$$

The average and the variance of the delay of an A-MPDU transmission containing ℓ sub-frames are calculated by:

$$E[D_l] = \sum_{k=1}^{K} \frac{p_{bo,l}^{k-1} \cdot p_{sc,l}}{1 - p_{bo,l}^K}((k-1)E[T_{fl,l}] + T_{sc,l} + \theta_1 \sum_{u=1}^{k} E[b_u]), \tag{24}$$

$$var[D_l] = \sum_{k=1}^{K} \frac{p_{bo,l}^{k-1} \cdot p_{st,l}}{1-p_{bo,l}^{K}}((k-1)var[T_{fl,l}] + ((k-1)E[T_{fl,l}] + T_{sc,l} - E[D_l] + \theta_1 \sum_{u=1}^{k} E[b_u])^2 + \sum_{u=1}^{k}(\theta_3 \cdot E[b_u] + \theta_1^2 \cdot var[b_u])), \quad (25)$$

where $T_{fl,l}$ is the duration when the transmission of A-MPDU containing ℓ sub-frames fails. $T_{fl,l}$ equals T_{cl} when a collision happens or $T_{ls,l}$ when all sub-frames are lost.

The average and the variance of the access delay are calculated by:

$$E[D_a] = \sum_{s=0}^{S} E[D_s], \quad (26)$$

$$var[D_a] = \sum_{s=0}^{S}(\alpha_{s,0}^{*}(E[D_s])^2 + \sum_{l=1}^{L} \alpha_{s,l}^{*} \cdot (var[D_l] + (E[D_l] - E[D_s])^2)), \quad (27)$$

where $E[D_s] = \sum_{l=1}^{L} \alpha_{s,l}^{*} E[D_l]$ means the average time in the s-th transmission.

4. Algorithms

In this section, the proposed aggregation scheduler is introduced. In the proposed scheduler, a timer is initiated when a new packet reaches the buffer and deals with the circumstance that no new packet arrival occurs. This circumstance happens when the cached packet is a request frame or the application is terminated. Therefore, the timer limitation is usually desired to be much larger than the arrival interval of previous packets. As shown in Figure 7, when a new packet arrives, whether an A-MPDU is forming and whether the TxQueue is full are checked. If neither is true, the number of cached packets is checked afterward. In case the number is not less than the optimal aggregation level derived from the proposed algorithms or the timer is expired, the AC will construct a grouped A-MPDU with the cached packets and queue the grouped A-MPDU into the TxQueue to wait for the access procedure. Otherwise, the AC waits for a new packet arrival. After the grouped A-MPDU is queued into the TxQueue, one A-MPDU procedure is finished and the next one is initiated from the forming and TxQueue checking.

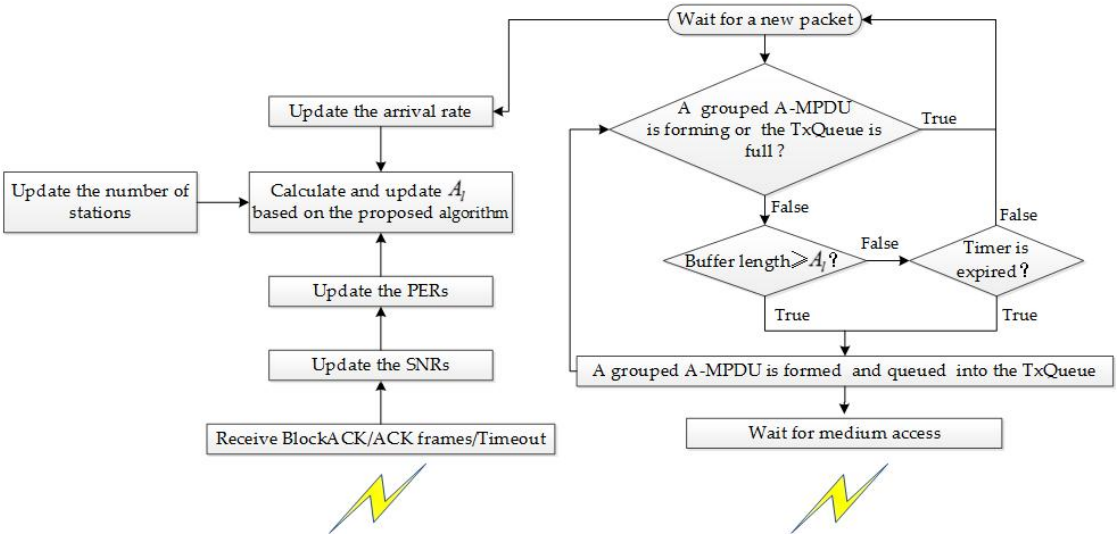

Figure 7. The procedure of the proposed A-MPDU aggregation scheduler.

$$A_l = argmin_L \; E[D_{e2e}(\lambda, BERs, L, N)], \tag{28}$$

$$\begin{aligned} s.t. \quad & 1 \leq L \leq W, \\ & pa < 1, \\ & \gamma^K < Thr_{loss}. \end{aligned}$$

The kernel of the aggregation scheduler is to search for the optimal aggregation level, A_l, to achieve the best QoS performance. As Equation (28) shows, the optimal aggregation level with which the minimum end-to-end delay is achieved is updated according to the packet arrival rate, the packet error rate (PER), and the number of active stations.

The constraints of the optimized problem are the limitations according to the IEEE802.11n/ac protocol or set to ensure the stable operation of the applications.

The first constraint limits the candidate aggregation level, L, according to the protocol. W denotes the aggregation window size.

The second constraint prevents the congestion in the WLAN. When pa is equal to 1, more than one grouped A-MPDU arrives at the TxQueue during each access procedure. That means that the current system's transmission capability cannot support the stable operation of applications at this arrival rate. Therefore, this constraint is important for the normal operation of applications.

The third constraint restricts the packet loss rate. Thr_{loss} is the QoS constraint for packet loss rate. There are two reasons for packet loss: the back-off times reach the retry limit of one transmission and the delay exceeds the service time in the WLAN. That a packet is discarded because it exceeds the packet lifetime seldom happens when the congestion is prevented by the second constraint. Therefore, only the packet loss that occurs due to reaching the retry limit is limited here.

The major hurdle of the proposed algorithm is its huge computational load. If the algorithm is conducted according to Equation (28), the calculation is relatively complicated because the calculation of the end-to-end delay is performed for W times. Moreover, the derivation of the distributions $\vec{\alpha}_s(e)$ and $\vec{\alpha}_\infty(e)$ requires multiple matrix iterations, which also increases the computation complexity. Therefore, three ideas were adopted to reduce the computation load. Two of them narrow the range of candidate aggregation levels and the last one offloads the computation load of matrix iterations:

(1) Those aggregation levels whose pa is equal to 1 can be removed in advance. As the back-off procedure is executed in units of A-MPDU and the overhead of idle and conflicting slots is independent of the aggregation levels, the average service time per packet is shortened when the aggregation level increases. Consequently, pa is in descending order according to Equation (20), and we can locate the smallest aggregation level whose pa is less than 1 in advance through the binary search method.

(2) According to Equation (2), the gathering delay ascends with the increase of the aggregation levels. Therefore, before the end-to-end delay is calculated, the gathering delay is compared with the best end-to-end delay at that point. If the gathering delay is greater than the currently best end-to-end delay, it is clear that the end-to-end delay containing the gathering delay is greater.

(3) According to the model, $\vec{\alpha}_s(e)$ and $\vec{\alpha}_\infty(e)$ only depend on the PER and the candidate aggregation level which are limited values. Therefore, the derivation of these distributions is preferred to be implemented offline and the results are stored in a look-up table with all possible PER and L. When calculating the end-to-end delay, the algorithm can fetch the distributions from the table by indexing with given PER and L.

The pseudo-code of the Algorithm 1 is as follows.

Algorithm 1 Searching for the optimal aggregation level.

Require: λ { The packet arrival rate.}
Require: Bit error rates (BERs) in all stations{The BERs in the channel can be estimated using the signal to noise ratio (SNR).}
Require: N{The number of active stations in the wireless local area network (WLAN).}
Ensure: A_l {The optimal aggregation level.}

```
procedure LowerBoundSearch(λ, BERs, n)
    L_min ⇐ 1;
    Calculate pa with aggregation level L = L_min, using Equation (20);
    if pa == 1 then
        left ⇐ 1;
        L_min ⇐ 64;
        Calculate pa with aggregation level L = L_min, using Equation (20);
        if pa < 1 then
            right ⇐ W;
            while right - left > 1 do
                L_min ⇐ (left + right)/2;
                Calculate pa with aggregation level L = L_min, using Equation (20);
                if pa < 1 then
                    right ⇐ L_min;
                else
                    left ⇐ L_min;
                end if
            end while
        end if
    end if
    return L_min
end procedure
procedure OptimalAggregationLevelSearch(λ, BERs, n, L_min)
    L ⇐ L_min;
    γ is calculated with aggregation level L, using Equation (23);
    while γ^K > Thr_loss and L ≤ W do
        γ is calculated with aggregation level L, using Equation (23);
        L++;
    end while
    D_e2e,opt is calculated with aggregation level L, using Equation (1);
    A_l ⇐ L_min;
    Calculate D_gather with new aggregation level L+1, using (2);
    while D_gather < D_e2e,opt and L ≤ W do
        L++;
        D_e2e is calculated with aggregation level L, using Equation (1);
        if D_e2e < D_e2e,opt then
            A_l ⇐ L;
            D_e2e,opt ⇐ D_e2e;
        end if
        Calculate D_gather with new aggregation level L+1, using Equation (2);
    end while
    return A_l
end procedure
```

5. Performance Evaluation

The performance of the proposed model and algorithm was evaluated by the NS-3 simulator [23]. The topology of the experimental network is shown in Figure 8. In the following experiments, there was a user datagram protocol (UDP) client in each station transmitting video-streams to the UDP server, which was operated in the access point (AP).

In the experiments, we simulated a basic video-stream whose parameters are listed in Table 1. Then we constructed the video stream of desired data rate by superposition of multiple basic flows.

Table 1. Parameters for the basic video-stream.

Data Rate	Frame Rate	Average Frame Size
5 Mbps	60 fps	10,341 bytes

In the IEEE802.11ac WLAN established for evaluation, the RTS/CTS was enabled and the number of spatial streams was 4. The buffer size was set to be a large enough value to not cause packet loss, and the packet lifetime parameter was set to 500 ms. To facilitate comparison and discussion, the BER was configurable in the simulations, similarly to [10,13–16]. If not specified, the PHY rate was the modulation and coding scheme (MCS)9, the BER was around 10^{-5}, the number of stations was 10, and the data rate of each video-stream was 20 Mbps. Other settings of the MAC layer are listed in Table 2.

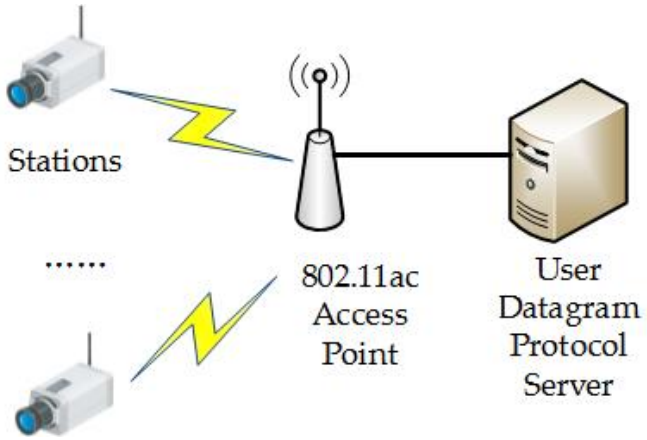

Figure 8. The topology of the experimental network.

Table 2. Parameters for the WLAN system.

Parameter	Value
W	64
retry limit	4
maximum back-off stage	2
minimum contention window (CW) size	8
$len_{MACheader}$	78 bytes
$len_{payload}$	36 + 1472 bytes
$\sigma(The\ slot\ time)$	9 µs
T_{DIFS}	43 µs
T_{SIFS}	16 µs
T_{BACK}	32 µs
$T_{BACKfail}$	76 µs
$T_{PHYheader}$	48 µs
T_{RTS}	42 µs
T_{CTS}	44 µs
$T_{CTSfail}$	76 µs

5.1. The Validation of the Proposed Model and Algorithm

First, the validation of the proposed model and algorithm was conducted with different BERs, arrival rates, and numbers of stations.

In Figure 9, the optimal aggregation levels and the searched ranges under different conditions are shown. The optimal aggregation levels obtained based on Equation (28) are shown as symbols. The dotted lines illustrate the range limited by the two methods adopted in the proposed algorithm. The upper bound is the aggregation level with a gathering delay larger than the optimal end to end delay, and the lower bound is the minimum aggregation level whose *pa* is less than 1.

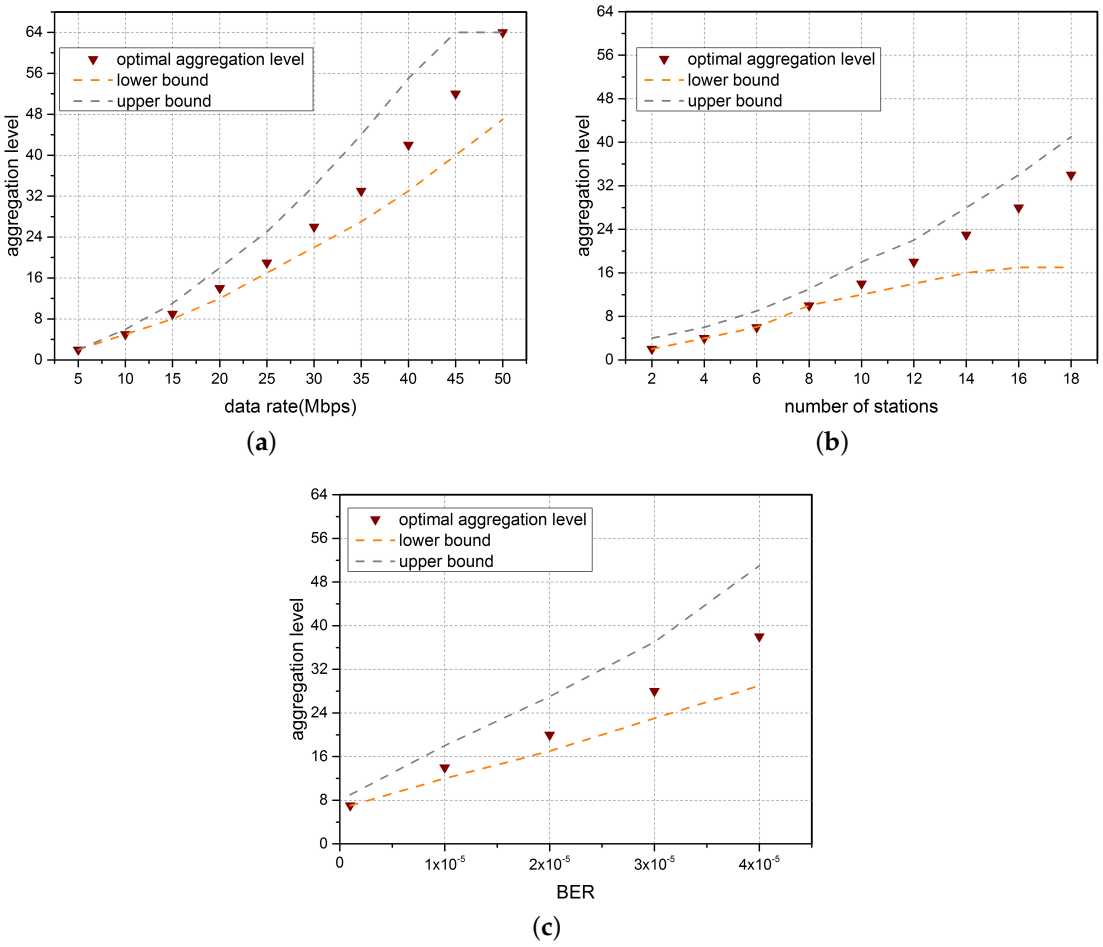

Figure 9. The optimal aggregation levels and the candidate ranges under different conditions. (**a**) The optimal aggregation levels and the candidate ranges under different data rates. (**b**) The optimal aggregation levels and the candidate ranges under different numbers of stations. (**c**) The optimal aggregation levels and the candidate ranges under different bit error rates (BERs).

From the simulation results shown, the optimal aggregation levels obtained based on Equation (28) are always located in the scaled range and the candidate range of the aggregation levels is reduced by 53.1–98.4%. When the working intensity of the system is low, the method that determines the upper bound plays a major role in excluding unqualified candidates. However, as the working intensity increases, the method that determines the lower bound tends to exclude more unfit candidates. Therefore, the adoption of both methods in the proposed algorithm can achieve a considerable reduction in scope under different conditions.

In Figure 10, the end-to-end delay model is evaluated by comparing the analyzed results and the simulated ones under different conditions. The optimal end-to-end delay obtained from the analysis

is shown on the red dot line and the result obtained from the simulation is drawn with the blue line. The error bar indicates the standard deviation of the simulation results. The deviation is increased along with the working intensity.

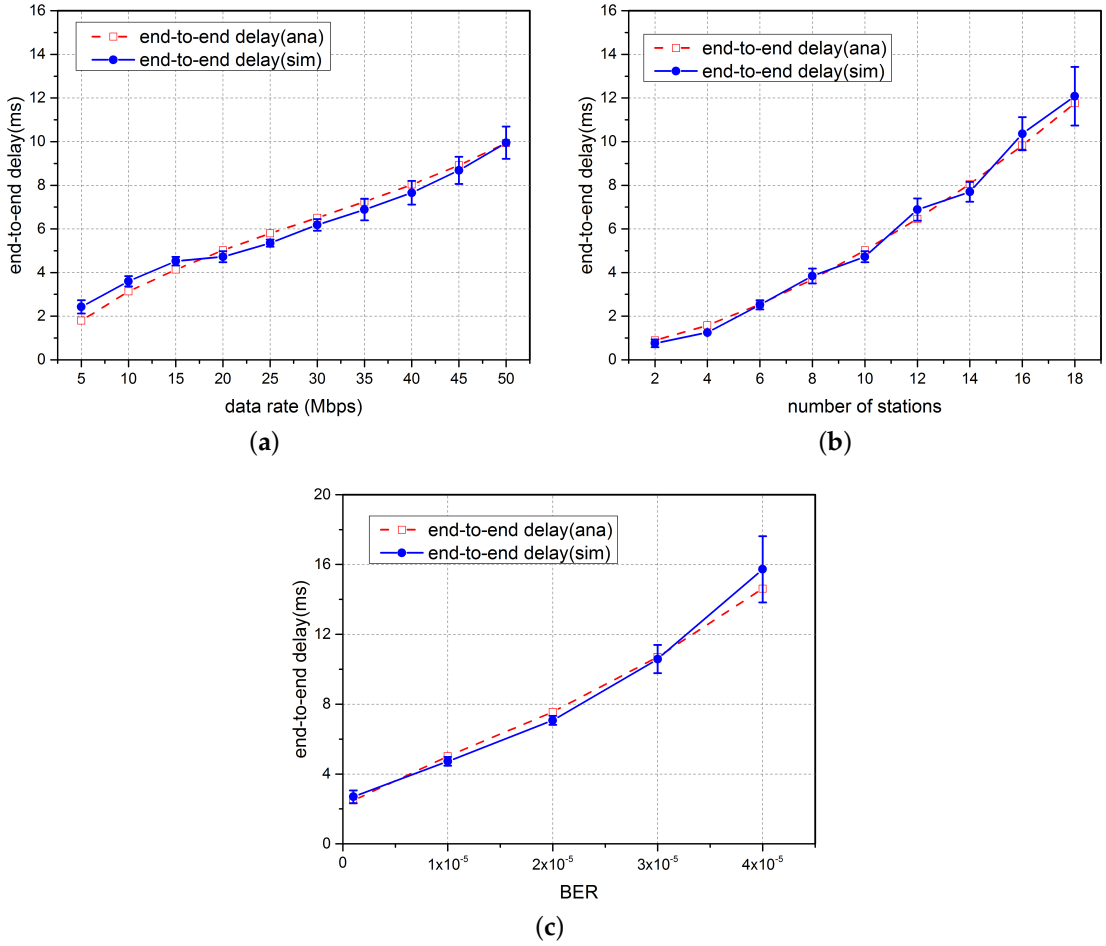

Figure 10. The optimal end-to-end delay under different conditions. (**a**) The optimal end-to-end delay under different data rates. (**b**) The optimal end-to-end delay under different numbers of stations. (**c**) The optimal end-to-end delay under different BERs.

It was observed that the simulation results were slightly better than the analyzed ones for some data rates and BERs. The reason for this phenomenon is that we used the average access delay of grouped A-MPDUs to represent that of packets to reduce the complexity. However, the actual value is smaller because the recipients upload the packets whose sequence numbers are smaller than the failed packets at the end of each transmission.

In general, the analyzed results well match the simulated ones and the trends of both lines are almost coincident. Therefore, the proposed model accurately estimated the end-to-end delay under different conditions.

From Figures 9c and 10c, when the BER is varied, the optimal aggregation level and the optimal end-to-end delay differ greatly. Therefore, it is necessary to analyze the end-to-end delay with the error-prone channels instead of the error-free channels.

5.2. The Performance Evaluation of the Proposed Scheduler

The performance evaluation was conducted by comparing the QoS performances of four different aggregation schedulers. There are three other adaptive aggregation schedulers in the existing literature,

called urgent access aggregation (UAA) [3,13,16], sliding window aggregation (SWA) [14,15], and more packet aggregation (MPA) [4].

The urgent access aggregation (UAA) scheduler serves the cached packets at the first opportunity and adaptively adjusts the aggregation level according to the number of the cached packets.

The sliding window aggregation (SWA) scheduler adopts the sliding window method that constructs an A-MPDU with both the retransmitted packets and the cached packets to enhance the efficiency.

The more packet aggregation (MPA) scheduler always tries to wait for enough cached packets to construct a full size A-MPDU when the QoS constraint is not violated. The full size A-MPDU is aggregated with the maximum aggregation level, W.

The optimal aggregation level (OAL) scheduler constructs the A-MPDU with the optimal aggregation level, which is derived based on the proposed end-to-end delay model and the proposed algorithm.

Figure 11 depicts the end-to-end delay against the data rates of the video-streams. The OAL follows the most steadily-increasing tendency and obtains the best end-to-end delay among the four. This is because the OAL achieves a balance between the collisions and the gathering delay to adapt to different scenarios. When the data rate is low, the MPA is much slower than other schedulers because more time is wasted for gathering the 64 packets due to a longer packet interval. However, the OAL avoids such a situation via adaptively adjusting the aggregation level based on the estimation of the gathering delay. When the data rate increases, the SWA and the UAA grow more rapidly than the OAL and are greatly increased in the end. The reason for the explosion of growth is that the volume of data exceeds the transmission capacity of the station, which causes a tremendous increase in the queuing delay. Owing to the reduction of the collision probability, the OAL achieves both better end-to-end delay and larger system capacity than the SWA and the UAA. When the end-to-end delay is greater than 100 ms, since dropping packets that exceed the survival time can reduce the rising momentum, the performance of end-to-end delay should be considered in conjunction with the packet loss.

Figure 12 illustrates the packet loss rate (PLR) against the data rates of the video streams. It should be noted that the ordinate is drawn non-linearly. The error bar indicates the standard deviation of the simulation results and its length is related to its position. The reasons for the packet loss are that the retry limit is reached when collisions are encountered, and that the packets exceed the survival time in the WLANs. The PLR of the UAA is around 0.1% when the data rate is less than 35 Mbps and larger than 20% when the data rate is greater than 40 Mbps. The multimedia applications are typically unable to work in such circumstances. Similarly, the PLR of the SWA is mostly around 0.1% and grows when the data rate is greater than 50 Mbps. In contrast, the PLRs of the OAL and the MPA are around 0.04% when the data rate is less than 50 Mbps and near 0.1% when the data rate is 55 Mbps. The PLRs of the OAL and the MPA are much more steady and outstanding than other schedulers because collisions are greatly eased. When the data rate is great, less packet loss is also due to the much lower end-to-end delay. Although the PLR of the MPA is smaller than that of the OAL when the data rate is less than 10 Mbps, it is essential to note that the decrease of the loss is at the cost of a longer gathering delay.

Consequently, considering the delay and the packet loss together, compared to other schedulers, the OAL-equipped systems can adapt to scenarios of different data rates and support applications that require higher data rates, lower latency, and less packet loss, simultaneously.

Figure 13 shows the trends of the end-to-end delay when the number of stations is variable. Concerning any given quantity of active stations, the OAL is the most outstanding among the four. For the OAL, the smallest end-to-end delay and the slowest growth trend were obtained because the collisions were reduced with the limited gathering delay. When the number of stations is small, the UAA and the SWA are very close to the OAL because the collisions under both schedulers are light thanks to the backoff procedure. However, when the number of stations is increased, the UAA and the SWA schedulers are boosted with the increase of the collisions. In particular, when the number of stations is larger than 14, the UAA and the SWA schedulers grow because more time is wasted for collisions and a longer queuing delay is needed when the packets congest in the originator. For the

MPA and the OAL, since some stations whose cached packets are less than the given aggregation level are absent from contending for the channel, the collisions are greatly reduced. As the gathering delay was fixed for the MPA scheduler, the end-to-end delay was around 20 ms for different numbers of stations.

Figure 14 depicts the PLR against the number of stations. It should be noted that the ordinate is drawn non-linearly. The error bar indicates the standard deviation of the simulation results and its length is related to its position. The PLRs of the UAA and the SWA are higher than those of the MPA and the OAL due to more severe collisions. The PLR of the UAA is over 0.1% when the number of nodes is greater than eight and the SWA exceeds 0.1% when the number of nodes is greater than 10. In this case, the QoS constraints required for real-time applications cannot be met. Thanks to the well-scheduled mechanism, the MPA and the OAL tended to grow more moderately and the top PLRs were around 0.1% even though there were 20 stations. The PLR of the MPA was the smallest for most cases due to the lowest probability of collisions; however, the gap between the MPA and the OAL was trivial, especially compared to the differences of the end-to-end delay.

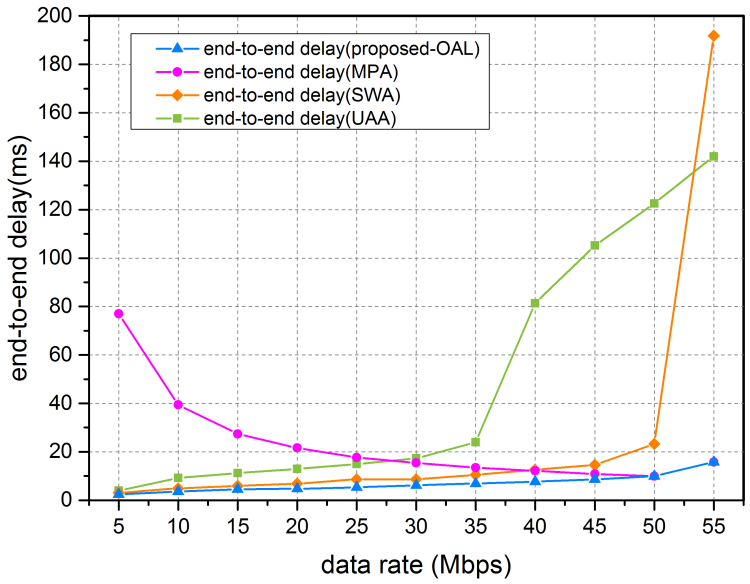

Figure 11. End-to-end delay vs. the data rate of the video-stream.

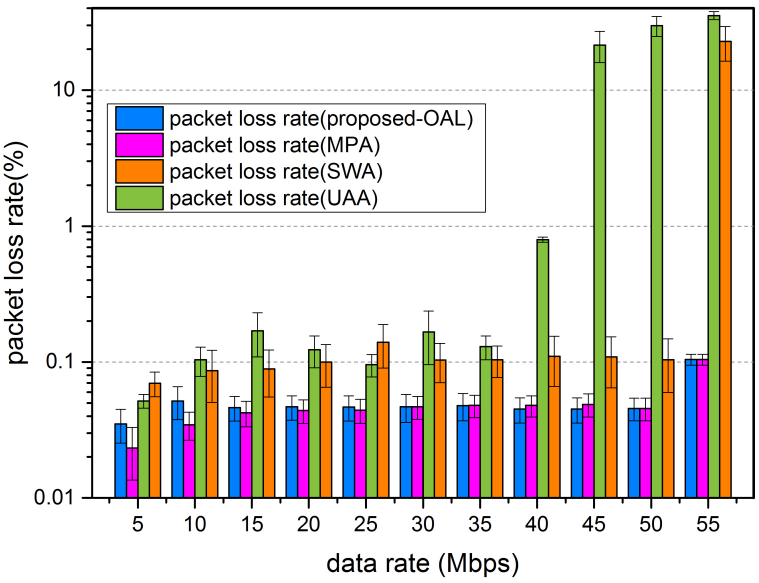

Figure 12. The packet loss rate (PLR) vs. the data rate of the video-stream.

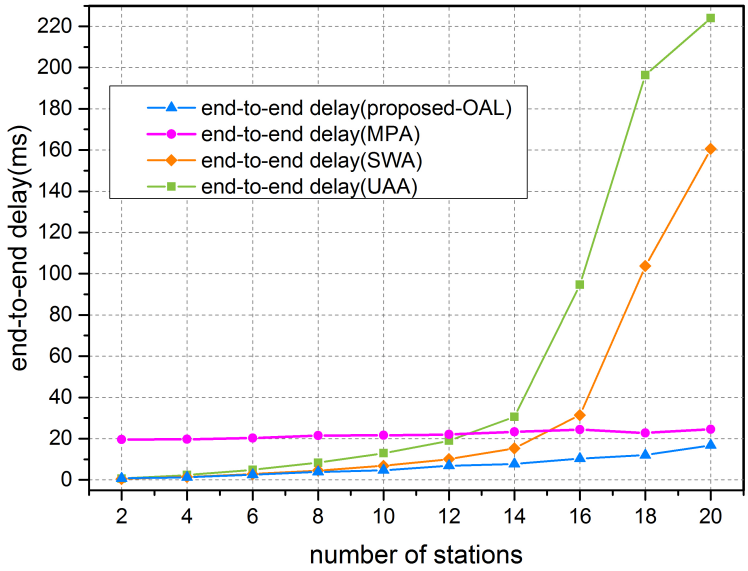

Figure 13. The end-to-end delay vs. the number of stations.

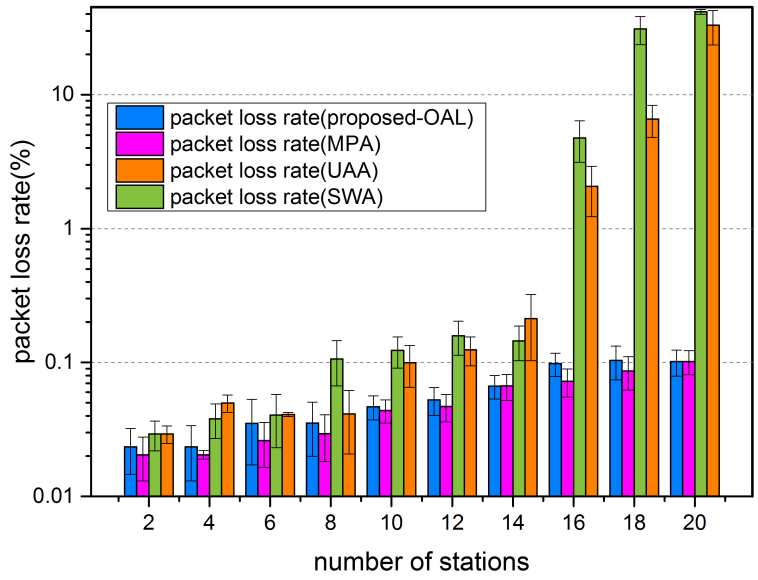

Figure 14. The packet loss rate (PLR) vs. the number of stations.

Consequently, a system equipped with the OAL can adapt to scenarios of different numbers of stations and guarantee the performances of applications that require high data rates, low latency, and low packet loss when the system contains 8–10 more stations.

6. Conclusions

In this paper, we extended the latency model under unsaturated settings and considered the gathering of packets, the queuing behaviors, and the transmissions using the RTS/CTS mechanism under error-prone channels. Then, we proposed a A-MPDU aggregation scheduler to improve the QoS performance of the IEEE802.11n/ac devices. The kernel of the scheduler was an algorithm that traded off between the extra delay of waiting for new packet arrival, the reduction of collisions, and the efficiency of aggregation to obtain the optimal aggregation level. To lower the computational complexity in the algorithm, we adopted two methods to narrow the candidate range and another one to offload the matrix iterations. According to the evaluated results from ns-3, the validation of our

model was confirmed and the proposed scheduler showed stronger adaptability in different scenarios and was able to improve the QoS performance and the system capacity dramatically.

Author Contributions: Conceptualization, C.L. and B.W.; methodology, C.L.; software, C.L.; validation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, C.L. and B.W.; project administration, B.W.; funding acquisition, B.W.; supervision, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Major Science and Technology Program of China grant number 2013ZX03004007.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *ITU-T Y.1541. Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks*; International Telecommunication Union (ITU): Geneva, Switzerland, 2011.
2. *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Networks-Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*; IEEE: Piscataway, NJ, USA, 2016.
3. Hajlaoui, N.; Jabri, I.; Taieb, M.; Benjemaa, M. A frame aggregation scheduler for QoS-sensitive applications in IEEE 802.11n WLANs. In Proceedings of the 2012 International Conference on Communications and Information Technology (ICCIT), Hammamet, Tunisia, 26–28 June 2012; pp. 221–226.
4. Seytnazarov, S.; Kim, Y.-T. QoS-Aware Adaptive A-MPDU Aggregation Scheduler for Voice Traffic in Aggregation-Enabled High Throughput WLANs. *IEEE. Trans. Mob. Comput.* **2017**, *16*, 2862–2875.
5. Coronado, E.; Villalón, J.; Garrido, A. Dynamic AIFSN tuning for improving the QoS over IEEE 802.11 WLANs. In Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015; pp. 73–78.
6. Ruscelli, A.L.; Cecchetti, G. Improving the QoS of IEEE 802.11e networks through imprecise computation. *Int. J. Ad Hoc Ubiquitous Comput.* **2016**, *23*, 152–167.
7. Syed, I.; Shin, S.-H.; Roh, B.-H.; Adnan, M. Performance Improvement of QoS-Enabled WLANs Using Adaptive Contention Window Backoff Algorithm. *IEEE Syst. J.* **2018**, *12*, 3260–3270.
8. Zawia, H.; Hassan, R.; Dahnil, D.P. Enhancement of real-time application IEEE 802.11e using dynamics contention windows approach. *Adv. Sci. Lett.* **2016**, *22*, 1874–1881.
9. Tian, G.; Camtepe, S.; Yu-Chu, T. A Deadline-Constrained 802.11 MAC Protocol With QoS Differentiation for Soft Real-Time Control. *IEEE Trans. Ind. Inform.* **2016**, *12*, 544–554.
10. Qian, X.; Wu, B.; Ye, T.-C. QoS-Aware A-MPDU Retransmission Scheme for 802.11n/ac/ad WLANs. *IEEE Commun. Lett.* **2017**, *21*, 2290–2293.
11. Lee, W.H.; Hwang, H.Y. A-MPDU aggregation with optimal number of MPDUs for delay requirements in IEEE 802.11ac. *PLoS ONE* **2019**, *14*, e0213888.
12. Charfi, E.; Gueguen, C.; Chaari, L.; Cousin, B.; Kamoun, L. Dynamic Frame Aggregation Scheduler for Multimedia applications in IEEE 802.11n Networks. *Trans. Emerg. Telecommun. Technol.* **2015**, *28*, e2942.
13. Hajlaoui, N.; Jabri, I.; Ben Jemaa, M. An accurate two dimensional Markov chain model for IEEE 802.11n DCF. *Wirel. Netw.* **2018**, *24*, 1019–1031.
14. Mansour, K.; Jabri, I.; Ezzedine, T. Revisiting the IEEE 802.11n A-MPDU Retransmission Scheme. *IEEE Commun. Lett.* **2019**, *23*, 1097–1100.
15. Seytnazarov, S.; Choi, J.-G.; Kim, Y.-T. Enhanced Mathematical Modeling of Aggregation-Enabled WLANs with Compressed BlockACK. *IEEE. Trans. Mob. Comput.* **2019**, *18*, 1260–1273.
16. Karmakar, R.; Swain, P.; Chattopadhyay, S.; Chakraborty, S. Performance modeling and analysis of high throughput wireless media access with QoS in noisy channel for different traffic conditions. In Proceedings of the IEEE International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 5–10 January 2016; pp. 1–8.
17. Charfi, E.; Chaari, L.; Kamoun, L. Fairness of the IEEE 802.11n aggregation scheme for real time application in unsaturated condition. In Proceedings of the 2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011), Toulouse, France, 26–28 October 2011; pp. 1–8.

18. Zhao, Q.; Tsang, D.H.K.; Sakurai, T. Modeling Nonsaturated IEEE802.11DCF Networks Utilizing an Arbitrary Buffer Size. *IEEE Trans. Mob. Comput.* **2011**, *9*, 1248–1263.
19. Wang, Q.; Jaffres-Runser, K.; Scharbarg, J.-L.; Fraboul, C.; Sun, Y.; Li, J.; Li, Z. A thorough analysis of the performance of delay distribution models for IEEE 802.11 DCF. *Ad Hoc Netw.* **2015**, *24*, 21–33.
20. Kleinrock, L. *Theory, Queueing Systems*; Wiley-Interscience: New York, NY, USA, 1975; Volume 1.
21. Zhao, Q.; Tsang, D.H.K.; Sakurai, T. A simple and approximate model for nonsaturated IEEE802.11 DCF. *IEEE Trans. Mob. Comput.* **2009**, *8*, 1539–1553.
22. Sakurai, T.; Vu, H.L. Access Delay of the IEEE802.11 MAC Protocol under Saturation. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 1702–1710.
23. NS-3 Network Simulator. Available online: http://www.nsnam.org/ (accessed on 18 June 2020).