CTS-to-self as a Protection Mechanism for the No Acknowledgment Protocol in VoIP WLANs

Anastasios C. Politis and Constantinos S. Hilas

Department of Informatics Engineering
Technological Educational Institute of Central Macedonia
62124, Serres, Greece

Abstract

In modern WLANs, the No Acknowledgment (NoACK) policy implements an acknowledgment-free protocol that is able to boost Medium Access Control (MAC) layer performance. However, due to the lack of a MAC-level recovery, a protection mechanism must be used in conjunction with the NoACK protocol in order to minimize the probability of packet loss. There are mainly two protection mechanisms defined by the IEEE 802.11 standard, namely RTS/CTS and CTS-to-self. While the first is well studied when used to protect the frame exchange with NoACK, the latter has not received any attention as a NoACK protection mechanism. In this paper we provide an analytical study to determine the system efficiency in terms of achieved throughput when these mechanisms are used to protect the acknowledgment suppression protocol in WLANs supporting voice applications. Based on the findings presented, a scheme based on the CTS-to-self mechanism is proposed to aid the application of the NoACK policy in WLANs. The proposed scheme is evaluated and its effectiveness is proven by means of simulation.

Keywords: No Acknowledgment; CTS-to-self; VoIP; WLANs;
1 Introduction

Multimedia applications are extremely popular to internet users nowadays. At the same time wireless connectivity is available to everyone who possesses a broadband internet access. Thus, modern WLANs are expected to support large amounts of ingress and egress multimedia traffic. These WLANs are typically operating in infrastructure mode allowing applications such as VoIP and streaming video to reach users around the globe.

Voice over Internet Protocol (VoIP) and other multimedia traffic have strict QoS requirements. To satisfy these requirements, the IEEE 802.11e amendment \cite{4} introduced a QoS-aware MAC layer to provide prioritized access to real-time over delay tolerant applications. The key contribution of this amendment is the definition of the Enhanced Distributed Channel Access (EDCA) function which implements a class-based differentiation mechanism together with a priority queuing system.

Besides the need of effective multimedia support in modern WLANs, there was also an important issue addressed by the same amendment: the disproportionate increase in data transmission rates and the MAC-layer efficiency. In fact, the MAC-layer functionality seemed to halt the benefits of the high rate Physical (PHY) layers. The gap between the advancement of the two layers started to shrink when IEEE 802.11e defined several mechanisms for MAC-layer efficiency improvement and is continuously closing with the appearance of newer techniques in the latest IEEE 802.11ac amendment \cite{5}.

In this group of enhancements new and optional acknowledgment policies defined in IEEE 802.11e are displayed prominently. The Block Acknowledgment (BlockACK) and the No Acknowledgment (NoACK) are two optional acknowledgment policies that may substitute the traditional two-way acknowledgment method defined in the initial IEEE 802.11 standard \cite{6}. The former is extensively analyzed and seems to be favored by the scientific community. The latter, on the other hand, received little attention by researchers.

According to the latest revision of the IEEE 802.11 standard in 2012 \cite{6}, when the NoACK protocol is selected as an acknowledgment policy, a protection mechanism should be used. There are two basic protection mechanisms defined by the standard for EDCA: the Request-To-Send/Clear-To-Send (RTS/CTS) and the CTS-to-self frame exchange. In the scarce number of research papers investigating the NoACK protocol, RTS/CTS was the selected protection mechanism. Currently, no other work has considered CTS-to-self in this role.

In this paper, the efficiency obtained by CTS-to-self as protection mechanism of the NoACK policy is investigated through an analytical study and compared with RTS/CTS and the standard two-way handshaking protocol (DATA/ACK). Moreover, based on the findings of this study we propose the
The rest of the paper is structured as follows. In Section 2 a description of EDCA and its main efficiency-oriented mechanisms are outlined. Section 3 reviews the related work and in Section 4 our analysis of the NoACK protection mechanisms is provided along with the results obtained regarding the efficiency of these mechanisms. In Section 5 we discuss several practical aspects regarding the application of the NoACK protocol and present our proposal details. In Section 6 we evaluate the effectiveness of the proposed scheme through simulation. In Section 7 we discuss a security-related issue regarding the proposed scheme and the paper is concluded in Section 8 with final remarks.

## 2 EDCA Background

EDCA achieves QoS by utilizing a traffic differentiation technique and a priority queuing system. In this way, delay sensitive multimedia frames receive privileged treatment to the disadvantage of other delay tolerant packets (e.g., FTP and HTTP). This differentiation-priority duplet complies with the laws of the Distributed Channel Function (DCF) which is the MAC-layer access mechanism defined in the original IEEE 802.11 specification. In essence, EDCA employs the basic Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mechanism, yet extends it by fine tuning MAC-layer parameters to achieve prioritized channel access to selected data frames.

This class-based QoS provisioning scheme is achieved by implementing four Access Categories (ACs) in a QoS-aware wireless station (QSTA) or a QoS-aware Access Point (QAP). Each AC is mapped to a different priority transmission queue which runs an instance of the EDCA access function. The access parameters for every priority queue are preset to a specific configuration in order to statistically prioritize real-time over non real-time traffic. This unequal service provisioning is feasible by manipulating Inter-Frame Spaces (IFS) and minimum and maximum contention windows. Real-time service queues are abbreviated as AC_VO and AC_VI, respectively. AC_VO handles voice traffic and owns the highest priority. AC_VI accommodates video frames and has the second highest priority for channel access. Non real-time traffic is serviced by AC_BK (background) and AC_BE (best effort) with the background category having the least chance of channel access.

### 2.1 Transmission Opportunity

A key feature introduced in the IEEE 802.11e amendment is the Transmission Opportunity (TXOP). TXOP is a contention-free period following immediately after the necessary collision avoidance idle gap. A QSTA (or a QAP)
that successfully acquires channel ownership is permitted to transmit multiple frames for a fixed time period. Thus, contention for each individual frame is avoided. The time for this contention-free burst (CFB) is bounded by the $TXOP_{\text{limit}}$ parameter which is AC and PHY specific. The TXOP option is enabled only to multimedia ACs (AC\_VO and AC\_VI) for all PHYs. On the other hand, non real-time ACs are permitted to transmit only one frame per medium access. In this way, real-time frames may experience lower latencies.

2.2 Acknowledgment Policies

The BlockACK and the NoACK policies intend to replace the standard positive acknowledgment protocol (Data/ACK), especially when data packets are transmitted during the TXOP won by a QSTA or a QAP. With the BlockACK policy the recipient can accumulate all acknowledgment frames into a single packet. On the other hand, the NoACK mechanism is less complicated and may be regarded as UDP-like protocol located at the MAC-layer. A frame transmitted by the originator does not require a positive acknowledgment by the recipient, thus all overhead imposed by the positive acknowledgment frames is completely vanished. The NoACK policy may, therefore, be used when channel conditions are extremely good and for error resistant applications (e.g., VoIP and streaming video).

2.3 Protection Mechanisms

The IEEE 802.11 standard dictates that when either the BlockACK or the NoACK policy is used, then a mandatory protection mechanism should be enabled to ensure that no other station transmits during the TXOP. For EDCA, two possible protection mechanisms are available: the RTS/CTS and the CTS-to-self frame exchange.

RTS/CTS was primarily defined to counter the hidden node problem in WLANs while the purpose of the CTS-to-self mechanism was to provide interoperability among IEEE 802.11g and IEEE 802.11b devices that co-exist in the same Basic Service Set (BSS). Both mechanisms perform a frame exchange at the beginning of a TXOP won by an AC, as shown in Fig. 1. This frame exchange allows the setting of the Network Allocation Vector (NAV) at all receiving stations. Since there may be legacy (e.g., IEEE 802.11b) nodes present in the BSS, these frames need to use legacy modulation techniques and lower data rates. The modulation type and the rate at which these control frames are transmitted are chosen among the available mandatory ones, as dictated by the standard [6]. Since IEEE 802.11 legacy nodes become increasingly rare, the CTS-to-self mechanism has no practical use in modern WLANs. On the other hand, RTS/CTS is still frequently used for protecting frame sequences.
Figure 1: Timing structure of three data frames in a CFB with (a) standard DATA/ACK policy, (b) NoACK policy with RTS/CTS protection, (c) NoACK with CTS-to-self protection.

3 Related Work

As stated earlier, the efficiency of the NoACK policy in modern WLANs is lightly investigated. To our knowledge only a handful of research papers are dedicated to this topic and they are briefly reviewed in this Section.

In [8], the authors research the impact that fiber delay has on the performance of the acknowledgment policies in fiber-fed WLANs. Their simulations, however, included the NoACK policy with RTS/CTS as the only protection mechanism.

In [3], the throughput and the packet loss rate obtained when the NoACK mechanism is enabled are investigated through analytical modeling. Based on their findings, the authors propose an adaptive acknowledgment scheme which they claim to be beneficial under different channel environments. However, their analysis did not include CTS-to-self as a protection mechanism. Furthermore, it is assumed that data bursts are always available at the transmitting stations. This is not always the case as shown in [9], especially for VoIP applications.

In our previous work [9], the RTS/CTS was considered as a protection mechanism for the acknowledgment suppression protocol in WLANs supporting voice traffic. In this work it was reported that an efficiency threshold exists under which the NoACK policy is outperformed by the standard DATA/ACK mechanism and a modification to the NoACK protocol was proposed. However, the CTS-to-self frame exchange was not considered in this investigation.

In [1], the authors present an analysis of the NoACK policy when applied to VoIP applications. A model is provided to quantify the benefits of the acknowledgment suppression technique. However, the authors explicitly state
that their inquiry does not include the usage of any protection mechanism, which makes this study unrealistic.

Another work which is relative to our research is the one presented in [2]. Even though the topic of this paper is data broadcasting in WLANs, the mechanism the authors focus on is the CTS-to-self frame exchange. This work suggests the expansion of CTS-to-self in order to be used as protection during frame broadcasting in WLANs. According to the authors, two basic modifications to the IEEE 802.11 MAC process are required: a CTS-to-self frame exchange prior to every data transmission and adjustment of the CTS frame modulation and data rate to the ones used by data transmission. As it will be explained in the following sections, in contrast to [2], our proposal does not require any modifications to the MAC-layer functionality thus it can be easily implemented.

4 Protection Mechanisms Analysis and Simulation

In this Section a simple analysis of the CTS-to-self and RTS/CTS protection mechanisms when used in conjunction with the NoACK policy is provided, followed by numerical and simulation results. The standard DATA/ACK mechanism is also included as a reference.

4.1 Analysis

In our analysis we assume an error-free channel, no hidden terminals, no stations with different PHY technologies and that the only active AC in a wireless node is the AC_VO in which the NoACK policy is enabled. Furthermore, we assume voice codecs producing equally-sized frames in standard intervals (no silence compression).

Both protection mechanisms introduce an overhead during the CFB of an AC as depicted in Figure 1. This overhead is given by:

\[
\begin{align*}
O_{\text{RTS}} &= T_{\text{RTS}} + T_{\text{CTS}} + 2T_{\text{SIFS}}, & \text{RTS/CTS} \\
O_{\text{CTS}} &= T_{\text{CTS}} + T_{\text{SIFS}}, & \text{CTS-to-self}
\end{align*}
\] 

(1)

where, \(T_{\text{RTS}}\) and \(T_{\text{CTS}}\) are the transmission times of the RTS and CTS frames respectively. \(T_{\text{SIFS}}\) is the Short Inter-Frame Space which is PHY specific.

The maximum number of equally-sized data frames, \(n_{\text{max}}\) that may be included in a CFB is derived by the following equation:
CTS-to-self as a protection mechanism for the NoACK protocol

\[ n_{\text{max}} = \begin{cases} \frac{\text{TXOP}_{\text{limit}}}{T_{\text{DATA}} + T_{\text{ACK}} + 2T_{\text{SIFS}}} \, , & \text{DATA}/\text{ACK} \\ \frac{\text{TXOP}_{\text{limit}} - O_{\text{RTS}}}{T_{\text{DATA}} + T_{\text{SIFS}}} \, , & \text{RTS}/\text{CTS} \\ \frac{\text{TXOP}_{\text{limit}} - O_{\text{CTS}}}{T_{\text{DATA}} + T_{\text{SIFS}}} \, , & \text{CTS-to-self} \end{cases} \] (2)

where, \( T_{\text{DATA}} \) and \( T_{\text{ACK}} \) are the transmission times of the data and the positive acknowledgement frames, respectively.

The time needed to transmit the whole burst of frames, \( T_{\text{burst}} \) is given by:

\[ T_{\text{burst}} = \begin{cases} n(T_{\text{DATA}} + T_{\text{ACK}}) + (2n - 1)T_{\text{SIFS}}, & \text{DATA}/\text{ACK} \\ O_{\text{RTS}} + nT_{\text{DATA}} + (n - 1)T_{\text{SIFS}}, & \text{RTS}/\text{CTS} \\ O_{\text{CTS}} + n(T_{\text{DATA}} + T_{\text{SIFS}}), & \text{CTS-to-self} \end{cases} \] (3)

where, \( n = \{1, 2, 3, ..., n_{\text{max}}\} \) and signifies the number of equally-sized frames included in the data burst.

Finally, an estimation of the achievable throughput, \( S \) by AC_VO can be obtained as follows:

\[ T = \frac{nl}{T_{\text{burst}}} \] (4)

where, \( l \) is the size of voice frames measured in bits.

4.2 Numerical and Simulation Results

In order to obtain our results we considered the G.729, G.728 and G.711 voice codecs which are used in many network related studies. The main difference among these codecs is the frame size and the frame inter-arrival time. Furthermore, we considered the IEEE 802.11g as the underlying PHY technology operating at 54 Mbit/s. With this PHY selected, the TXOP limit value according to the IEEE 802.11 standard is set to 1504 usec.

In our calculations the necessary MAC and PHY-layer overhead was also taken into account. The parameters and their values used to obtain our numerical and simulation results are summarized in Table 1.

Due to the relatively old version of the simulation tool that was available (OPNET Modeler version 11.5), including newer PHYs such as IEEE 802.11n
Table 1: PHY and MAC parameters used in numerical and simulation analysis.

<table>
<thead>
<tr>
<th>PHY</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{SIFS}$ = 10,us</td>
<td>Data header length= 240bits</td>
</tr>
<tr>
<td>$T_{PLCP}$ = 20,us</td>
<td>ACK and CTS header length= 112bits</td>
</tr>
<tr>
<td>Data Frame Rate= 54Mbps</td>
<td>RTS header length= 160bits</td>
</tr>
<tr>
<td>Control Frame Rate= 24Mbps</td>
<td>$TXOP_{\text{limit}}$ = 1504,us</td>
</tr>
</tbody>
</table>

Figure 2: Maximum number of equally-sized voice frames included in a CFB for different VoIP codecs.

and IEEE 802.11ac in our simulation tests was not possible at the time of writing this paper. However, these PHYs will be considered in our future research endeavors. Nonetheless, the outcome of a similar experiment on newer versions of the IEEE 802.11 standard is expected to yield analogous results. As shown in Table 1, the control frames (ACK, RTS and CTS) are sent with the highest mandatory data rate which is set at 24 Mbit/s, since we are considering that all stations are ERP (Extended Rate Physical) capable with data transmission rate of 54 Mbit/s.

Fig. 2 depicts the value of $n_{\text{max}}$ for the three VoIP codecs considered. These values were obtained by applying Equation (2) and they were also confirmed by simulation. It is obvious that applying the NoACK policy increases the maximum number of frames that can be incorporated into the CFB for all VoIP codecs. Another thing that can be noted is that selecting the CTS-to-self as the NoACK protection mechanism $n_{\text{max}}$ is increased by one frame, compared to the RTS/CTS case.

In Fig. 3 the $T_{\text{burst}}$ is illustrated for an ascending number of frames in the CFB. A finding firstly reported in [5] is visible in the graph. An efficiency threshold exists under which the NoACK policy with RTS/CTS as protection
mechanism is outperformed by the standard acknowledgment policy. This threshold is settled on two frames, meaning that in order to obtain an efficiency increase by the application of NoACK it is required that the frames included in the CFB is more than two. This threshold exists only for the RTS/CTS protection mechanism and is vanished when the CTS-to-self is applied. It must be noted that the number of frames included in the CFB is well beyond the five frames limit exhibited in the graph (see Fig. 2), however this was limited to five frames in order to highlight clearly the efficiency threshold.

Figure 4 shows the achieved throughput by the AC_VO under the standard DATA/ACK mechanism and the two variations of the NoACK protocol for the G.711, G.728 and G.729 VoIP codecs. Firstly, the expected finding is that codecs with higher frame sizes produce higher throughput values. However, in all three cases the NoACK policy with the CTS-to-self mechanism as protection outperforms the other two options. CTS-to-self is able to eliminate the efficiency threshold and if the CFB incorporates more than a single frame an efficiency increase can be observed.

We must note here, that the throughput values illustrated in Fig. 4 are obtained for a single active QSTA which is not a realistic scenario but gives us an insight on the performance potential of each mechanism.

5 Proposed Scheme

We are interested in VoIP WLANs that operate in infrastructure mode. These networks include a QAP and a number of QSTAs which are associated with
that QAP, QSTAs nodes typically incorporate a single outgoing voice stream. Since the packet inter-arrival time of VoIP codecs range from 5 ms (G.728 and G.726 codecs) to 30 ms (G.723.1 codec) a QSTA underutilizes its assigned TXOP for AC_VO. This means that the CFB of a QSTA with a single voice station will typically contain a single voice frame. On the contrary, the QAP accumulates all voice traffic destined to the wireless hop and thus it may include multiple voice frames in the CFB of its AC_VO. Hence, the NoACK policy can be considered as an alternative acknowledgment policy at the QAP.

Based on the findings presented in the previous Section, we recommend the usage of NoACK with CTS-to-self protection only at the QAP and only if the RTS/CTS mechanism is enabled at the QSTAs of the BSS. With this setup a possible collision will occur among the RTS frame of a QSTA and the CTS frame originating from the QAP. This will lead the QSTA to enter its backoff procedure and the QAP to continue with its frame burst. Furthermore, the period that the network will be jammed by the occurring collision will be significantly less if the colliding frames are control frames (such as the RTS and CTS frames) than the case that involves a data frame in the collision event. The above remarks are illustrated in Fig. 5.

In order to ensure the usage of RTS/CTS at the QSTAs when the QAP switches to the NoACK policy, the beacon frame can be used to indicate this transition to QSTAs. More specifically, the acknowledgment policy that the QSTAs must use can be indicated into the EDCA Parameter Set of the beacon frame.
CTS-to-self as a protection mechanism for the NoACK protocol

6 Evaluation of the Proposed Scheme

In order to evaluate the performance of the proposed scheme we implemented it in a simulation environment using the OPNET Modeler tool. A series of simulation scenarios with an infrastructure BSS and a variable number of active VoIP nodes were conducted. Each QSTA produced 160 Bytes voice frames at regular intervals of 20 ms leading to a 64 kbps one-way load (G.711 codec). Since VoIP traffic is typically symmetrical, the load produced by the QAP towards the wireless hop was increased appropriately as the number of QSTAs increased. This accounted for the accumulating downlink voice traffic. More specifically, the load at the QAP was set to:

\[ QAP\ load = Number\ of\ QSTAs \times 64\text{kbps} \]  

The NoACK mechanism was enabled only at the AC_VO of the QAP with a different protection mechanism (RTS/CTS and CTS-to-self) in each simulation run. The PHY and MAC parameters used during the simulations were the ones presented in Table 1.

We have considered two simulation cases:

- **NoACK with RTS/CTS Protection:** in this case the NoACK policy is enabled only at the QAP and is protected by the RTS/CTS frame exchange. All QSTAs use the standard acknowledgment policy during their frame transmissions.
- **Proposed Scheme**: as before, the NoACK policy is enabled only at the QAP but this time with CTS-to-self as protection. All QSTAs use the RTS/CTS frame exchange prior to their data transmissions. This was indicated by a modified beacon frame transmitted by the QAP in regular intervals.

For the above cases we are interested on the average system throughput and the average delay achieved in the WLAN. All the results depicted in the graphs were estimated with a 95% confidence interval. For the average throughput (Fig. 6) the error is in the scale of 1 Kbps on average, while for the delay (Fig. 7) it is in the scale of 0.5 msec on average. With such narrow error margins the error bars in the graph are not visible, thus they are omitted. Fig. 6 compares the throughput obtained by the NoACK protocol protected by the RTS/CTS mechanism and the proposed scheme both enabled at the QAP. The graph also includes the offered load produced in the WLAN as a reference. As the number of QSTAs increases, so is the load produced by the QAP, as Eq. (5) implies. This load increase combined with the high number of QSTAs will also lead to an increased number of frame collisions affecting the throughput obtained. As it is shown in the graph, this is indeed true for the case where RTS/CTS is selected as the protection mechanism. RTS frame collisions become high leading to increased backoff values and ultimately to a dropping tendency of the QAP throughput. However, when CTS-to-self protects the frame sequences, the throughput obtained for AC_VO follows closely the offered load, even for high numbers of QSTAs. Collisions still occur among CTS frames from the QAP and RTS frames from the QSTAs. Nevertheless, these collisions lead QSTAs to backoff stage while QAP continues with its voice frame sequence.

The increased collision count also has a negative influence on the delay experienced by voice packets. As Fig.7 illustrates, enabling RTS/CTS as protection of the NoACK protocol, the average packet delay values become prohibitively high for large numbers of QSTAs. ITU-T G.114 [7] recommends a maximum of a 150 ms one-way latency. Since this includes the entire voice path, part of which may be on the public Internet, the BSS should incur transit latencies of well below the 150 ms threshold. However, the RTS/CTS case exceeds this threshold considerably earlier than the CTS-to-self case. The latter manages to keep low and acceptable packet delays even when the number of QSTAs reaches the maximum value set in our simulations (60 nodes).

### 7 Security-related Issues

Enabling the proposed scheme in a VoIP infrastructure WLAN may expose the network to misbehavior-related attacks. More specifically, the RTS frames
Figure 6: Average overall throughput observed in the WLAN when NoACK is enabled at the QAP with RTS/CTS as protection and the proposed scheme for G.711 VoIP codec (95% confidence interval).

Figure 7: Average voice frame end-to-end delay observed in the WLAN when NoACK is enabled at the QAP with RTS/CTS as protection and the proposed scheme for G.711 VoIP codec (95% confidence interval).
transmitted by the QSTAs, which are administrated by users, can be modified accordingly to indicate a fraudulent high period of channel reservation. This will set the NAV at all stations (including the QAP) to this high value. The result will be that only the misbehaving node will be able to acquire channel ownership during that period. This forms a type of a Denial of Service (DoS) attack.

As also noted by [2], a way to alleviate this malicious act may be to introduce a threshold to the allowable value inserted in the Duration/ID field included in the MAC header of the frame. However, this will require additional modifications to the MAC-layer functionality. In any case, this type of attacks will be a topic of investigation in our future work.

8 Conclusion

In this paper the CTS-to-self frame exchange as a protection mechanism for the NoACK policy was investigated and compared with RTS/CTS for WLANs supporting VoIP traffic. The results indicate a noticeable efficiency increase in terms of both the achievable throughput and the delay which render CTS-to-self a more attractive solution. Based on the findings obtained we proposed a scheme that favors the usage of the NoACK protocol as the acknowledgment policy in VoIP WLANs operating in infrastructure mode. The simulation results indicate that the proposed scheme outperforms the current version of the NoACK protocol which is based on the RTS/CTS as a protection mechanism.

Acknowledgements. The authors wish to acknowledge financial support provided by the Special Account for Research Funds of the Technological Education Institute of Central Macedonia, Greece, under grant SAT/IE/04102017-152/20.

References


Received: April 16, 2018; Published: May 8, 2018