Scientific
Research
Publishing

Coupled IEEE 802.11ac and TCP Goodput Improvement Using Aggregation and Reverse Direction

Oran Sharon[1], Yaron Alpert[2]

[1]Department of Computer Science, Netanya Academic College, Netanya, Israel
[2]Intel, Ra'anana, Israel
Email: oran@math.haifa.ac.il, Yaron.Alpert@intel.com

Abstract

This paper suggests a new model for the transmission of Transmission Control Protocol (TCP) traffic over IEEE 802.11 using the new features of IEEE 802.11ac. The paper examines the first step in this direction and as such we first consider a single TCP connection, which is typical in a home environment. We show that when the IEEE 802.11ac MAC is aware of QoS TCP traffic, using Reverse Direction improves the TCP Goodput in tens of percentages compared to the traditional contention based channel access. In an error-free channel this improvement is 20% while in an error-prone channel the improvement reaches 60% also using blind retransmission of frames. In our operation modes we also assume the use in Two-Level aggregation scheme, the Automatic Repeat-Request (ARQ) protocol of the IEEE 802.11ac MAC layer, the data rates and the four Access Categories defined in this standard.

Keywords

802.11ac, TCP, Aggregation, Reverse Direction, Goodput

1. Introduction

1.1. Background

The latest IEEE 802.11-REVmc Standard (WiFi), created and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802.11) [1] that embedded and updated the IEEE 802.11ac amendment, is currently the most effective solution within the range of Wireless Local Area Networks (LAN). Since its first release in 1997, the

standard provides the basis for Wireless network products using the WiFi brand, and has since been improved upon in many ways. One of the main goals of these improvements is to optimize the Throughput of the MAC layer, and to improve its Quality-of-Service (QoS) capabilities.

To fulfill the promise of increasing IEEE 802.11 performance and QoS capabilities, and to effectively support more client devices on a network, the IEEE 802.11 working group introduced the fifth generation in IEEE 802.11 networking standards; the IEEE 802.11ac amendment, also known as Very High Throughput (VHT) [1] [2]. IEEE 802.11ac is intended to support fast, high quality data streaming and nearly instantaneous data syncing and backup to notebooks, tablets and mobile phones. The IEEE 802.11ac final version, 11ac-2013, released in 2013 [2], leverages new technologies to provide improvements over previous generation, *i.e.* IEEE 802.11-2012 [3]. Both versions are now included in IEEE 802.11-REVmc [1] which will be published as IEEE 802.11-2016.

The IEEE 802.11ac amendment [2] improves the achieved Throughput coverage and QoS capabilities, compared to previous generations, by introducing improvements and new features in the PHY and MAC layers. In the PHY layer, IEEE 802.11ac (VHT) continues the long-existing trend towards higher Modulation and Coding rates (256 QAM 5/6 modulation), working in wider bandwidth channels (up to 160 MHz) and using 8 spatial streams that enable higher spectral efficiency.

In the MAC layer IEEE 802.11ac includes many of the improvements first introduced with IEEE 802.11e and IEEE 802.11n [3], also known as High Throughput (HT). Integrated with the following two key performance features are the ability to aggregate packets in order to reduce transmission overheads in the PHY and MAC layers, and to use Reverse Direction (RD) which enables stations to exchange frames without the need to contend for the channel. We now describe these features.

Frame aggregation is a feature of the IEEE 802.11n and IEEE 802.11ac that increases Throughput by sending two or more consecutive data frames in a single transmission, followed by a single acknowledgment frame, denoted *Block Ack* (BAck). Aggregation schemes benefit from amortizing the control overhead over multiple packets. The achievable benefit from data aggregation is often interesting, especially in light of several factors that can impact its performance, e.g., link rates, collisions, error-recovery schemes, inter-frame spacing options, QoS guarantee, etc. IEEE 802.11n introduces, as a pivotal part of its MAC enhancements, three kinds of frame aggregation mechanisms: The Aggregate MAC Service Data Unit (A-MSDU) aggregation, the Aggregate MAC Protocol Data Unit (A-MPDU) aggregation and the Two-Level aggregation that combines both A-MSDU and A-MPDU. The last two schemes group several Mac Protocol Data Units (MPDU) frames into one large frame. IEEE 802.11ac also uses these three aggregation schemes, but enables larger frame sizes.

The basic idea behind the Reverse Direction (RD) feature is a time interval denoted Transmission Opportunity (TXOP). A station gains a TXOP by gaining access to the wireless channel and in a TXOP the station can transmit several PHY Protocol Data Units (PPDU) without interruption. This station is denoted the *TXOP holder*. The TXOP holder can also allocate some of the TXOP time interval to one or more receivers in order to allow data transmission in the reverse link. This is termed as *Reverse Direction* (*RD*). For scenarios with bidirectional traffic, such as Transmission Control Protocol (TCP) Data segments/TCP Acks, RD is very attractive because it reduces contention in the wireless channel (no collision).

The IEEE 802.11ac standard also defines an Automatic Repeat-Request (ARQ) protocol that enables a transmitter to retransmit lost MPDUs and guarantee in-order reception of MPDUs at the receiver. This protocol is also used to improve quality of the wireless channel.

Another feature in IEEE 802.11ac related to QoS capabilities is the use in Access Categories (AC). There are 4 ACs: Best Effort (BE), BackGround (BK), Video (VI) and Voice (VO). The difference between the 4 ACs is in the parameters that control access to the channel, namely the *Arbitrary Inter-Frame Space* (*AIFS*) length and the values of CW_{min} and CW_{max}. These vary in the various ACs and are intended to provide priority to traffic streams with QoS requirements such as Video and Voice.

1.2. Research Question

In this paper we investigate a model to transmit TCP traffic in an infrastructure IEEE 802.11 that optimizes the combined performance of the IEEE 802.11 MAC layer and the L4 TCP protocol using the new features that were developed in the latest generation of the IEEE 802.11, *i.e.* IEEE 802.11ac. These features enable to use IEEE 802.11 in a completely different way than before, as we now specify.

The issue of TCP performance over IEEE 802.11 networks has been investigated in many papers in the past,

e.g. [4]-[17]. In this past research, however, it is assumed that stations compete to get access to the channel using the contention based CSMA/CA access method. Collisions are possible between stations that are involved in different TCP connections and between the AP and stations with which it has TCP connections due to the exchange of TCP Data/Ack segments. Both the AP and the station(s) try to get access to the channel simultaneously and this results in collisions.

As far as we know, there was no development of models to transmit TCP traffic over IEEE 802.11 using new features of the standard. In the model we suggest in this paper the AP controls the TCP transmissions in the cell by configuring the stations to use large BackOff intervals such that effectively they never gain access to the channel and the AP enables the stations to transmit only through time periods delivered by the AP, the TXOP holder, *i.e.* the Reverse Direction (RD) capability. The AP communicates with one station during a TXOP and in this paper we evaluate the performance of such communication between the AP and the station. Establishing policies for the communication between the AP and several stations with which it maintains TCP connections is the issue for further research.

Therefore, as a first step we assume that the AP communicates with a single station and they have a single TCP connection between them. The AP is the TCP transmitter, transmitting TCP Data segments, and the station is the TCP receiver transmitting TCP Acks. Our performance criteria is the *Goodput*, defined as the number of MSDUs' bits (TCP Data segments) that are successfully transmitted and acknowledged by TCP Acks, in the wireless channel, on average, in a second. Such a scenario is possible, for instance in a home environment where a Network-Attached Storage (NAS) device [18] is attached to the AP, and a PC downloads data files from the NAS device. It uses the aggregation and RD capabilities of the IEEE 802.11ac MAC layer.

We use the four features of the MAC layer of the IEEE 802.11ac mentioned above, namely aggregation, Reverse Direction (RD), the ARQ protocol and the four Access Categories. Concerning aggregation, we assume the Two-Level aggregation scheme. This scheme enables transmission of several TCP Data segments and several TCP Acks in a single transmission over the wireless medium. Up to 64 MPDUs can be transmitted in a single transmission and every MPDU can contain several MSDUs. We measure the influence of aggregation on the Goodput.

Notice that in a TCP connection over IEEE 802.11ac both sides of the connection compete for the wireless channel-one for transmitting TCP Data segments and the other for transmitting TCP Acks. This competition can result in collisions and reduced Goodput. We examine two operation modes for the transmission of TCP traffic over the wireless medium. In one operation mode, using RD, the TCP transmitter allocates a TXOP when it acquires the wireless medium, and enables the TCP receiver to transmit TCP Acks during the TXOP without collisions. In the second operation mode, for comparison purposes, which we denote by *No-RD*, the traditional CSMA/CA random access MAC is used. The TCP transmitter and the TCP receiver contend for the wireless medium in every transmission attempt. The operation mode using RD is more complicated than the contention based one, and we want to check if, and to what degree, using RD improves the Goodput of the *No-RD* operation mode.

In addition to all the above we assume the ARQ protocol of the IEEE 802.11ac standard at the MAC layer. This protocol guarantees an in-order delivery of MPDUs between communicating entities. However, due to its Transmission Window, the ARQ protocol can sometimes limit the number of MPDUs transmitted in each transmission, *i.e.* this protocol can limit the amount of aggregation.

Finally, we check the influence of the values of the access parameters in the four ACs on the Goodput, namely the Arbitrary Inter Frame Space (AIFS), Contention Window min., CW_{min}, and Contention Window max. CW_{max}.

We assume that the AP and the station are the end points of the TCP connection. Following e.g. [5] [8] [9] [17] it is quite common to consider short Round Trip Times (RTT) in this kind of high speed networks such that no retransmission timeouts occur. Notice also that due to the MAC ARQ protocol, the L4 TCP protocol always receives TCP Data segments in order. Therefore, the TCP congestion window increases up to the TCP receiver advertised window. We assume that the TCP receiver window is large enough such that the TCP Transmitter Transmission window can always provide as many MSDUs to transmit as the MAC layer enables. We assume the above following the observation that aggregation is useful in scenario where the offered load on the channel is high. We therefore do not consider the TCP Transmission Window and our goal is to find the maximum possible Goodput that the wireless channel enables to a single TCP connection, where the TCP itself does not impose any limitations on the offered load, *i.e.* on the rate that MSDUs are given for transmission to the MAC

layer of the IEEE 802.11ac.

Following the above we also do not consider a particular flavor of TCP, e.g. TCP NewReno, Westwood, Cubic [19]-[21] if to mention only a few. All the TCP flavors differ in the way they handle the TCP congestion window but in this paper, as mentioned, we assume that the TCP Transmission Window is limited only by the TCP receiver advertised window.

Regarding the wireless channel quality we first assume an error-free channel, *i.e.* the Bit Error Rate (BER) equals 0. Then we assume another three BERs: $10^{-5}, 10^{-6}$ and 10^{-7}. The scenario of a single TCP connection with various BER values is possible for instance in the mentioned home environment where a Network-Attached Storage (NAS) device [18] is attached to the AP, and a PC, which is a client in the IEEE 802.11 system, is located close to the NAS and downloads data files from the NAS device. The various BERs are a function of the channel conditions between the client (e.g. PC) and the AP. If they are stable and have a low path loss channel between them. the BER is very low. However, if the PC is located in the basement for instance, the BER can be larger.

An additional feature that we use was introduced in [22]. In [22] a repetition scheme is introduced, in which several MPDUs in a single transmission are transmitted several times. This feature improves the achieved Goodput in large BERs, as will later become clear.

1.3. Our Results

We show that for an error-free channel, *i.e.* BER = 0, using RD improves the Goodput over not using RD by 20%. Moreover, using TXOPs of about 20 μs are sufficient to achieve that improvement, and this outcome has an impact on the delay at the TCP protocol from the time the TCP transmitter transmits TCP Data segments until it receives the corresponding TCP Acks.

For error-prone channels we show that using RD improves the Goodput in almost 50% and when also using the Repetition scheme of [22] the improvement can even reach 60%. TXOPs of about 4 μs are sufficient to achieve these Goodput improvements.

1.4. Previous Work

From the point of view of Transport protocols, the performance of the IEEE 802.11 protocol has been investigated in two models: UDP-like traffic and TCP traffic, *i.e.* when there is bi-directional traffic that can result in collisions. By UDP-like traffic we mean that the Data receiver does not transmit an Ack at the Transport layer, nor, in terms of IEEE 802.11, does it generate an MSDU for transmission. In TCP traffic, the receiver of TCP Data segments generates an MSDU which contains a TCP Ack, and depends on the channel for its transmission.

Regarding UDP-like traffic, the performance of IEEE 802.11 (taking into account the aggregation schemes) has been investigated in dozens of papers over the years. For example, in [23]-[35] the Throughput and Delay performance of the A-MSDU, A-MPDU and Two-Level aggregation schemes are investigated. Several papers assume an error-free channel with no collisions, several papers assume an error-prone channel and others also assume collisions. In [36]-[40] the performance of 802.11ac is investigated. Papers [37] [40] consider the performance of the aggregation schemes in 802.11ac and compare the performance of 802.11ac to that of 802.11n.

Another set of papers [41]-[46] deals with QoS together with the aggregation schemes. In particular, in [46] the use of the ARQ protocol of the IEEE 802.11 standard [1], together with the aggregation schemes, is investigated in relation to QoS guarantee.

Concerning TCP traffic, we can specify a first set of papers that deal with TCP's Throughput, Delay and Fairness performance over legacy IEEE 802.11/a/b/g networks. There are dozens of such papers, such as [4]-[7] [9]-[11] to mention only a few. None of the papers from this set consider Access Categories or aggregation schemes that were introduced in later versions of the standard, *i.e.* IEEE 802.11e and IEEE 802.11n respectively.

As the IEEE 802.11e was introduced, many papers appeared concerning this standard and the performance of TCP. In IEEE 802.11e the Access Categories are defined, enabling change to the fix values of the DIFS (now called AIFS) and $CW_{\min}$ of the previous versions of the standard. Also introduced is the TXOP time interval that enables the AP/stations to transmit several frames in a single transmission opportunity. Such frames are

acknowledged in the MAC layer, all together, by a new defined frame; the *Block-Ack* frame. Papers regarding TCP investigate the use of the above changes in improving TCP performance [12] [13]. None of the papers concerning IEEE 802.11e and TCP deal with ACs and aggregation schemes, as does our paper, since aggregation schemes were only introduced in a later version of the standard, namely IEEE 802.11n.

In relation to IEEE 802.11n/ac where aggregation is introduced, we are aware of only three research papers that handle the Throughput performance of TCP in the various aggregation schemes [14] [15] [25]. In [25] the authors also assume the model of the AP and a single station that maintain a TCP connection. The paper considers the A-MSDU and A-MPDU aggregation schemes only, and does not consider the Two-Level aggregation scheme, the RD and the various ACs. In the analysis the authors assume a TCP Transmission window of one TCP Data segment. On the other hand, in this paper we also handle the Two-Level aggregation scheme, the various ACs, the RD and a TCP Transmission window larger than one Data segment, which complicates the analysis.

In [14] it is argued that aggregation increases the discrepancy among upload TCP connections. The model is an AP with several stations that initiate TCP upload connections. The A-MPDU aggregation is considered and there is no a reference to Two-Level aggregation, to RD and to the standard ACs. The authors suggest an algorithm to reduce the discrepancy among TCP connections. Our paper deals with another model: we explore the influence of aggregation on the Goodput of a *single* TCP connection, e.g. in a home environment, consider Two-Level aggregation, RD and check the performance of the 4 ACs defined in IEEE 802.11 .

In [15] the performance of a single TCP connection is evaluated using all three aggregation schemes and four standard ACs. However, only an error-free channel is considered and there is no reference to RD, *i.e.* there can be collisions between the two parties of the TCP connection. The current paper is a next step to the research in [15] in the sense that it also considers an error-prone channel and explores the elimination of collisions by RD.

Regarding RD, there are several papers such as [47]-[50] that deal with RD's Goodput performance, also in relation to TCP. However, these papers do not consider aggregation, ACs and the IEEE 802.11ac ARQ protocol all together.

Finally, none of the papers mentioned in this literature survey consider the Repetition scheme of [22] and its influence on the Goodput performance.

The rest of the paper is organized as follows: In Section 2 we describe in detail the features of the IEEE 802.11ac that we use in this paper. In Section 3 we describe the model we suggest for TCP transmission over IEEE 802.11 using RD. In Sections 4 and 5 we compute the Goodput performance of the error-free and error-prone channels respectively. Section 6 concludes the paper and in the Appendix we present a Markov chain model for the scenario in which there is no use in RD and both the AP and the station contend for the channel in every transmission attempt.

2. Network Model

2.1. Aggregation Schemes

Three aggregation schemes are defined in IEEE 802.11n/ac: Aggregate MAC Service Data Unit (A-MSDU), Aggregate MAC Protocol Data Unit (A-MPDU) and Two-Level aggregation, which combines the former two.

In A-MPDU aggregation several MPDUs are combined together into a single PHY Service Data Unit (PSDU) denoted A-MPDU frame, and are transmitted in one PHY Protocol Data Unit (PPDU), thus saving PHY overhead. The Two-Level aggregation scheme is shown in **Figure 1**. In this aggregation scheme several MPDUs are again inserted for transmission into one A-MPDU frame. However, an MPDU can contain several MSDUs. Every MSDU is preceded by a SubFrame Header of 14 bytes and every MSDU, with its SubFrame Header, is rounded by a PAD to a size that is an integral multiple of 4 bytes. Every MPDU is preceded by a MAC Delimiter of 4 octet and is rounded by a PAD with its delimiter, to a length that is an integral multiple of 4 octets. Such MPDUs are denoted A-MSDU frames. The Two-Level aggregation scheme achieves a better ratio than the other aggregation schemes between the amount of Data octets transmitted to the PHY and MAC layers' overhead.

In 802.11ac the maximum A-MPDU's size is 1,048,575 octets and the MPDUs' maximum length is 11,454 octets. The maximum number of MPDUs in an A-MPDU frame is 64.

Assume an A-MPDU frame that contains K MPDUs and Y MSDUs. Let L be the length of an MSDU in bytes. Recall that according to the IEEE 802.11ac standard [1] [2], every MSDU within an MPDU frame is preceded
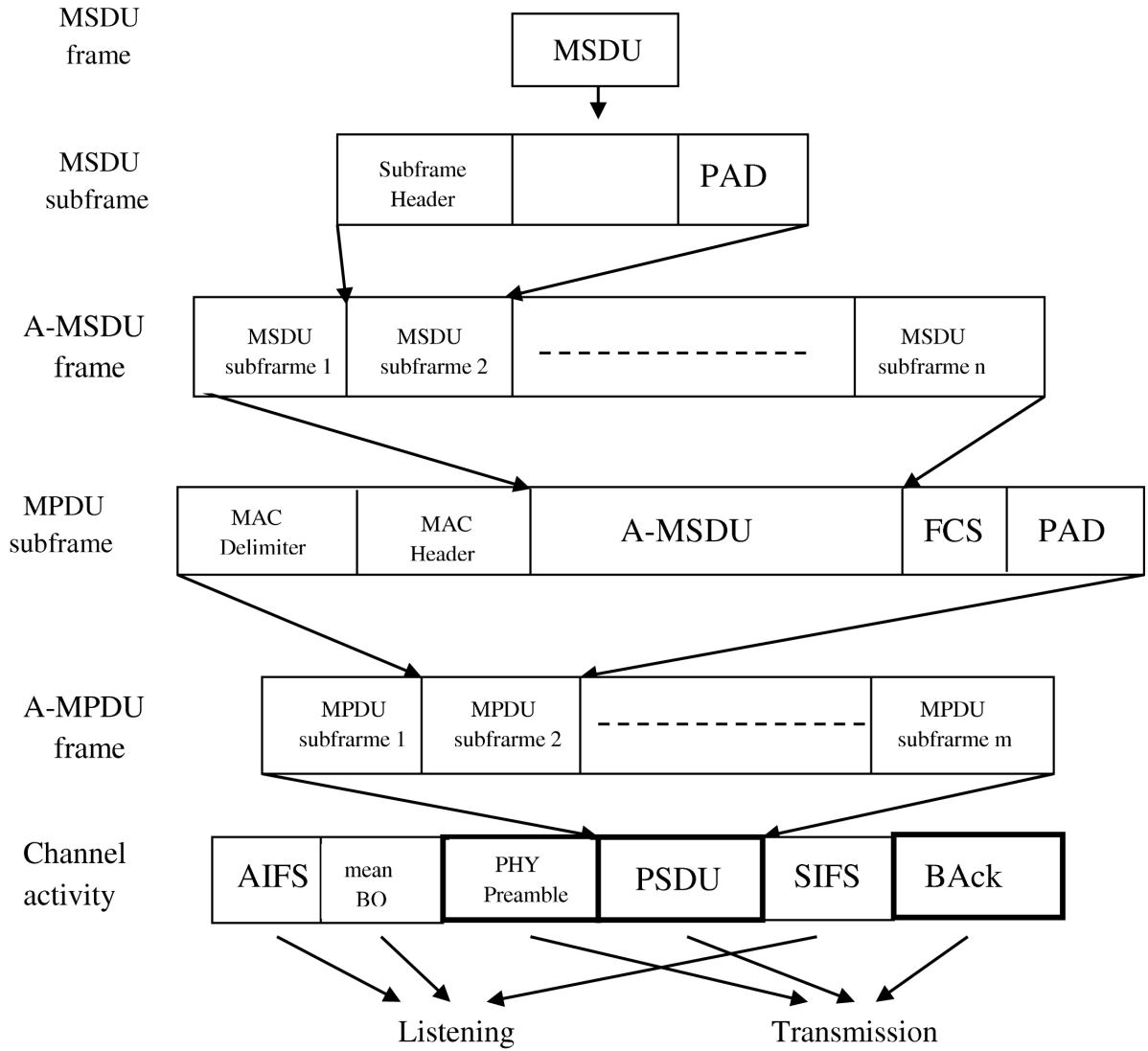
Figure 1. The Two-Level aggregation process.

by a SubFrame Header of 14 bytes, and with this field it is rounded to an integral multiple of 4 bytes. The received size is L' such that $L' = 4 \cdot \left\lceil \dfrac{L+14}{4} \right\rceil$. Recall also that every MPDU within a PSDU is preceded by a MAC Delimiter. To compose an MPDU one also adds the MAC Header and Frame Control Sequence (FCS) fields to the MSDUs of the MPDU. Thus, the length Len of the PSDU in bytes is $Len = (K \cdot H) + (Y \cdot L')$ where $H = 4 \cdot \left\lceil \dfrac{MacDelimiter + MacHeader + FCS}{4} \right\rceil$. MacDelimiter, MacHeader and FCS denote the sizes in bytes of the MAC delimiter, MAC Header and FCS fields respectively.

The receiver of an A-MPDU frame acknowledges its reception by a Block Ack (BAck) control frame. In BAck the receiver separately acknowledges the reception of every MPDU in the received A-MPDU frame.

Let $O_1 = AIFS + Preamble + SIFS + BAck$ and let $BackOff$ be the BackOff interval that a station uses in a given transmission. The transmission time without collisions of the above A-MPDU is [1]:

$$O_1 + BackOff + TSym \cdot \left\lceil \frac{8 \cdot Len + 22}{BitsPerSymbol \cdot R} \right\rceil. \tag{1}$$

The additional 22 bits are due to the SERVICE (16 bits) and TAIL (6 bits) fields that are added to every transmission by the PHY layer Conv. Protocol [1].

In Equation (1) we assume the OFDM PHY layer. *Tsym* is the duration of one Transmission Symbol in OFDM, and it is 4 μs. *BitsPerSymbol* equals 4 in OFDM and *R* is the PHY rate in Mbps. Any transmission in OFDM must be of an integral number of Symbols.

2.2. The Error Model

We assume that the process of frame loss in a wireless fading channel can be modeled with a good approximation by a low order Markovian chain, such as the two state Gilbert model [51] [52].

In this model the state diagram is composed of two states, "Good" and "Bad", meaning successful or unsuccessful reception of every bit arriving at the receiver, respectively. *Bit-Error-Rate* (BER) is the probability of moving from the Good state to the Bad state. $(1-BER)$ is the probability of staying at the Good state. According to the above model, the success probability of a frame of length *B* bits is $(1-BER)^B$ and the failure probability *p* is given by Equation (2):

$$p = 1 - (1-BER)^B \tag{2}$$

By the above model one can see that as the frame length *B* increases, so does the failure probability. Thus, in every aggregation scheme, increasing the aggregation amount increases the frame's size as well as the transmission delay of the frame. The failure probability can sometimes also increase.

We would like to mention that there are other models to represent the quality of the indoor wireless channel, e.g. the one in [53]. This model shows burstiness in the channel quality. In this paper however, we assume that the communicating stations use Link Adaptation by which they keep the effective SNR stable and in such a scenario the BER is stable.

2.3. IEEE 802.11ac ARQ Protocol

We give only a brief description of the IEEE 802.11ac ARQ protocol. A more detailed description can be found in [46] and in sections 9.21.7.3 - 9.21.7.9 in [1].

Consider the transmission of a series of MPDUs from one entity to another in IEEE 802.11ac. MPDUs are numbered, and the recipient signals the transmitter which MPDUs arrived successfully and which in error. Failed MPDUs are retransmitted by the transmitter. The number of retransmissions of an MPDU is limited.

The transmitter maintains a Transmission Window (TW) over the sequence numbers of the MPDUs. We denote this transmission window by MAC TW, to distinguish it from transmission windows of higher levels' protocols, such as the one of TCP. Only MPDUs within the MAC TW are allowed for transmission to the recipient. The maximum size of the TW is 64 consecutive sequence numbers because the recipient can acknowledge at most 64 MPDUs in one BAck control frame [1].

Let X be the smallest sequence number in the MAC TW and X + 63 be the largest. As long there is no acknowledgment from the recipient that MPDU X arrived successfully, the MAC TW does not change. When an acknowledgment for MPDU X arrives, the MAC TW moves one position (number) along the sequence numbers' space: X is taken out and X + 64 is inserted into the MAC TW.

Let *K* be the maximum number of MPDUs that can be transmitted in one PPDU in Two-Level aggregation. Assume that MPDU X has been transmitted several times with no success. In this case the MAC TW is unchanged and it is possible that only *M* MPDUs within the MAC TW, $M < K$ are unacknowledged by the recipient. In such a case only *M* MPDUs are transmitted and the MAC TW limits the number of transmitted MPDUs. As *K* is larger the probability for such a scenario is larger.

2.4. Timing

We assume the following values for the time intervals used in IEEE 802.11ac and we assume that the reader is familiar with the basic access scheme of IEEE 802.11ac networks. For the OFDM PHY layer SlotTime = 9 μs and SIFS = 16 μs. The BAck and Ack frames are 32 and 14 bytes long respectively. Their transmission times, denoted *BAckTime* and *AckTime* respectively are 32 μs and 28 μs respectively, using the Basic PHY Rate of 24 Mbps. These times include the PHY preamble preceding the transmissions of these frames. If the PHY rate *R*

used for data frame transmissions is lower than 24 Mbps, then *R* is also used for the BAck and Ack transmissions. However, in this paper we assume a PHY rate of 1299.9 Mbps assuming working point MCS9 with 3 spatial streams and an 80 MHz channel. With 3 spatial streams the PHY Preamble is 48 μs [1].

2.4.1. Successful Transmissions

In **Figure 2** we show the activity on the channel where a successful transmission occurs, *i.e.* without collisions. In this case, after a station senses an idle channel for a duration equal to its AIFS and BackOff intervals, it transmits the data frame. After a SIFS and a PHY Preamble the receiver acknowledges reception. In the case of Two-Level aggregation the BAck frame is used.

2.4.2. Collision Events

In **Figure 3** we show the activity on the channel in the event of collisions. We show two stations, A and B. After the channel is clear, and assuming that the Network Allocation Vector (NAV) at both stations is equal to 0, both stations wait the AIFS interval. If their BackOff intervals are equal, both stations begin together to transmit their data frames when the BackOff intervals terminate. If the data frame of B is shorter than that of A, then when B terminates its transmission it detects a carrier on the channel; that of A's transmission. Thus, it recognizes that they have collided. When A terminates its transmission, it waits the SIFS interval, recognizes that it has not received an acknowledgment and so detects the collision. Both A and B now wait the interval Extended Inter Frame Space (EIFS) after the transmission of A terminates. The channel then becomes clear and the BAckOff intervals at the stations start again. EIFS is the interval that stations wait in IEEE 802.11ac after a collision is detected [1].
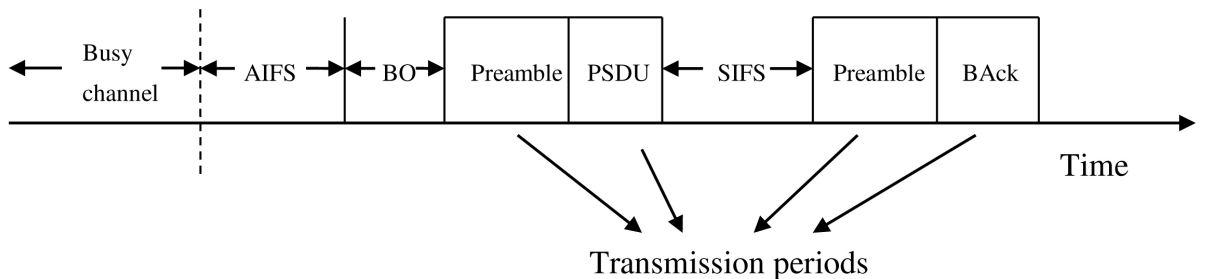
2.5. Access Categories

As mentioned, we consider the four ACs defined in the IEEE 802.11ac standard, *i.e.* BE, BK, VO and VI. The ACs defer in the values of the parameters that control the access to the channel, namely AIFS, CW_{min} and CW_{max}. For every AC the value of AIFS, denoted AIFS [AC], equals to $SIFS + AIFSN[AC] \cdot SlotTime$. The various Access Category Numbers (AIFSN [AC]), CW_{min} and CW_{max} appear in **Table 1**. The values are taken from the WiFi Alliance (WFA) publications [54]. The EIFS used in every AC, denoted $EIFS[AC]$, equals to $SIFS + AckTime + AIFS[AC]$. For the computation of EIFS, it is assumed that the Ack frame is transmitted in the smallest basic PHY rate of 6 Mbps, *i.e.* the $AckTime$ for the computation of the EIFS is 44 μs. The value of AIFS [AC] and EIFS for every AC also appear in **Table 1**.
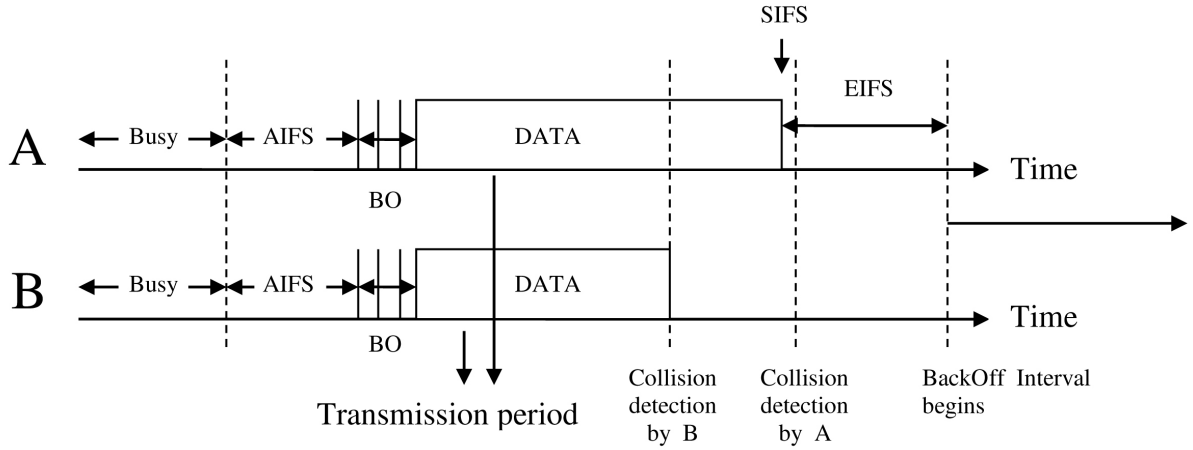
3. TCP Traffic Model over IEEE 802.11

In this section we describe our model for the transmission of TCP traffic over IEEE 802.11. In **Figure 4** we show the Traffic flow considered.

We assume an application with a massive Data stream, for example a Video stream that uses TCP, and its Data bytes are mapped into equal length TCP segments. A TCP segment is mapped into an IP Datagram. IP Datagrames are given to the MAC layer of the IEEE 802.11ac as MAC Service Data Units (MSDU) and these are packed into MPDUs (A-MSDUs). MPDUs are kept in a Transmission Queue and are transmitted using Two-Level aggregation. Recall that we assume a saturation scenario in which the TCP always has an unlimited number of Data segments to transmit. Also notice the MAC TW mentioned in Section 2.3.

Figure 2. The activity on the channel in the case of a successful transmission.

Figure 3. The activity on the channel in the case of a collision.

3.1. Operation Modes for TCP Usage of the Channel

We consider 2 operation modes for the transmission of TCP Data/Ack segments over the channel.

3.1.1. Operation Mode 1-No-RD, Competition

Both the TCP transmitter and the TCP receiver contend for the channel in every transmission attempt, *i.e.* when the TCP receiver has TCP Acks to transmit, it contends for the channel with the TCP transmitter in every transmission. Both stations use the Two-Level aggregation.

3.1.2. Operation Mode 2-Reverse Direction

Reverse Direction is a mechanism in which the owner of a Transmission Opportunity (TXOP) can enable its receiver to transmit back during the TXOP, so that the receiver does not need to contend for the channel. This is particularly efficient for a bi-directional traffic such as TCP Data segments and TCP Acks.

We examine an operation mode in which the TCP transmitter (AP) transmits A-MPDU frames containing MPDUs of TCP Data segments to the TCP receiver (station), and enables the TCP receiver to answer with an A-MPDU frame containing MPDUs frames of TCP Acks. Both stations use the Two-Level aggregation.

We assume the following scenario to use RD, as is illustrated in **Figure 5**:

After waiting AIFS and BackOff the TCP transmitter (AP) transmits n A-MPDU frames in a row in the TXOP. In **Figure 5** we assume $n = 2$. The TCP receiver (station) responds to every transmission by a BAck frame. In its last A-MPDU frame the TCP transmitter sets the RDG bit [1], enabling the TCP receiver to respond with an A-MPDU frame. The TCP transmitter then responds with a BAck frame and terminates the TXOP with the CF-End frame [1].

We assume that there are no collisions on the channel after the end of a TXOP because the TCP receiver is configured in a way that prevents collisions. For example, the TCP receiver is configured to choose its BackOff interval from a very large contention interval, other than the default ones in **Table 1**. Thus, the TCP transmitter always wins the channel without collisions. The transmissions on the channel are composed of TXOPs that repeat themselves one after the other. We denote by $RD(n)$ the case where the TCP transmitter transmits n A-MPDU frames in the TXOP.

4. Error-Free Channel Results

In this section we assume an error-free channel, *i.e.* BER = 0, and in this case the operation mode using Reverse Direction (RD) is as follows: Every transmission of the TCP transmitter contains K_D MPDUs, $1 \leq K_D \leq 64$. Assuming TCP Data segments of $L_{DATA} = 1480$ bytes, the resulting IP Datagrams are of 1500 bytes (20 bytes of IP Header are added) and together with the SubHeader field and rounding to an integral multiple of 4 bytes, every MSDU is of $L'_{DATA} = 1516$ bytes. Due to the limit of 11,454 bytes on the MPDU size, 7 such MSDUs are possible in one MPDU. The total number of MSDUs transmitted by the TCP transmitter in one TXOP is
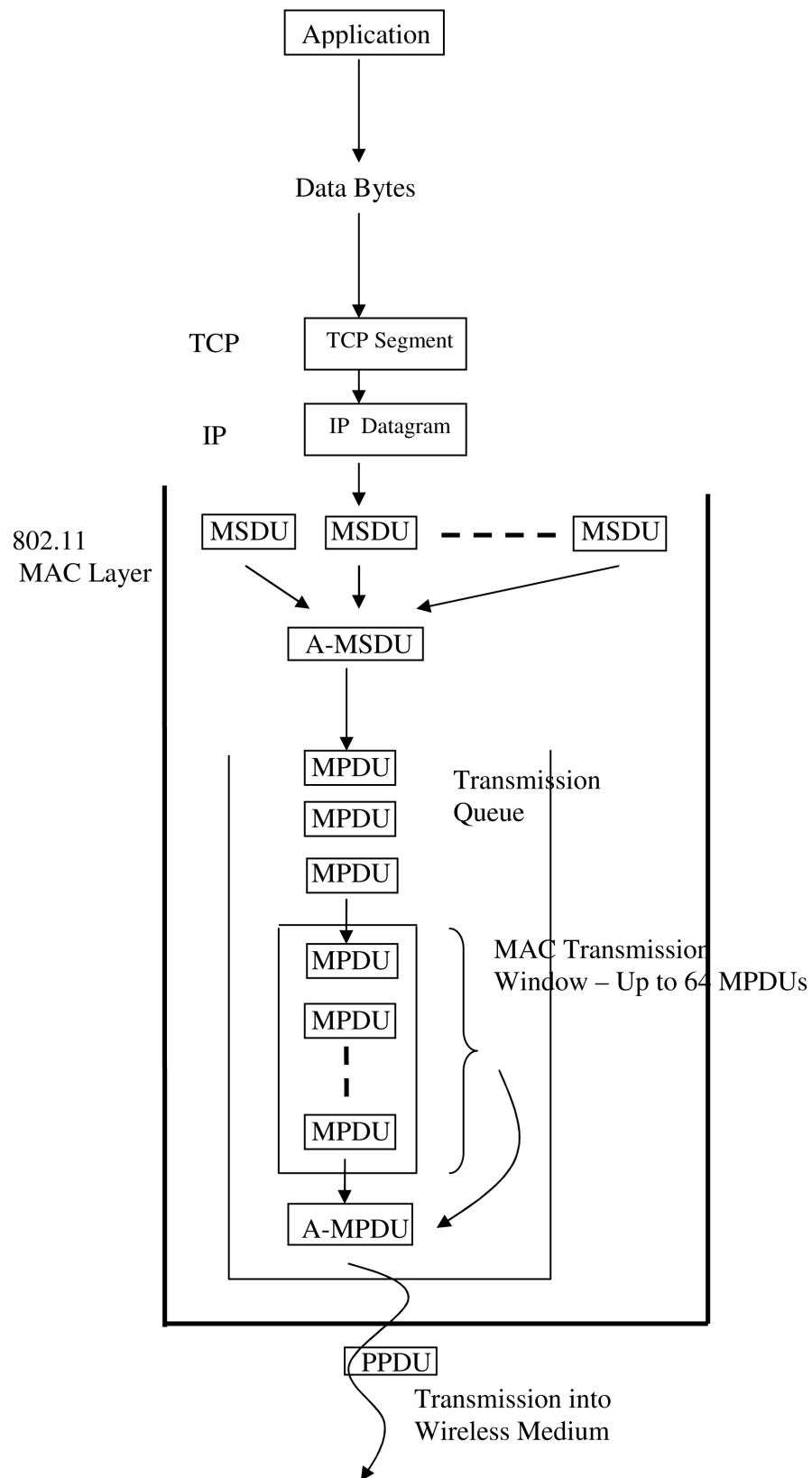
Application

Data Bytes

TCP — TCP Segment

IP — IP Datagram

802.11
MAC Layer

MSDU MSDU — — — — MSDU

A-MSDU

MPDU

MPDU   Transmission
Queue

MPDU

MPDU   MAC Transmission
Window – Up to 64 MPDUs

MPDU

MPDU

A-MPDU

PPDU

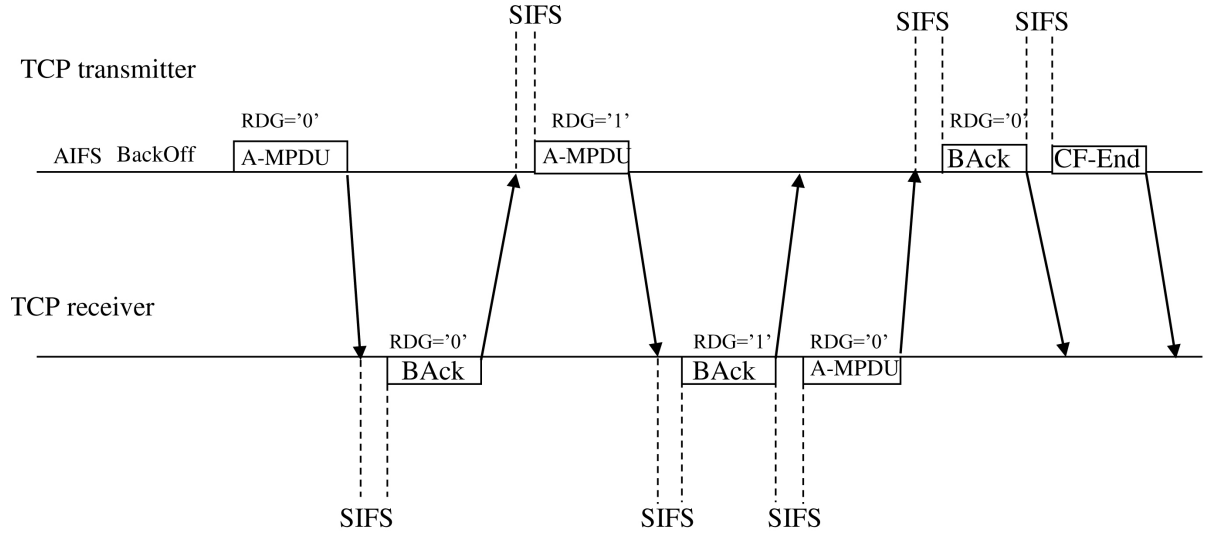Transmission into
Wireless Medium

Figure 4. The traffic model.

Figure 5. The operation mode that uses reverse direction.

Table 1. The values of CW_{min}, CW_{max}, AIFS number, AIFS and EIFS in the four access categories of IEEE 802.11ac for a station (Access point).

	BK	BE	VI	VO
CW_{min}	16 (16)	16 (16)	8 (8)	4 (4)
CW_{max}	1024 (1024)	1024 (64)	16 (16)	8 (8)
AIFSN	7 (7)	3 (3)	2 (1)	2 (1)
AIFS[AC](μs)	79 (79)	43 (43)	34 (25)	34 (25)
EIFS[AC](μs)	139 (139)	103 (103)	94 (83)	94 (83)

therefore $n \cdot K_D \cdot 7$.

The TCP receiver transmits TCP Acks. Every TCP Ack is of $L_{Ack} = 48$ bytes (20 bytes of TCP Header + 20 bytes of IP header + 8 bytes of LLC SNAP). Adding 14 bytes of the SubHeader field and rounding to an integral multiple of 4 bytes, every MSDU of the TCP receiver is $L'_{Ack} = 64$ bytes, and every MPDU, again due to the size limit of 11,454 bytes, can contain 178 MSDUs. Every transmission of the TCP receiver can contain up to 64 MPDUs.

The receiver can transmit up to 64×178 TCP Acks (MSDUs) in a single transmission. Therefore, the number n of transmissions of the TCP transmitter in a TXOP should be limited by the following inequality: $n \cdot K_D \cdot 7 \leq 64 \times 178$. Using larger n's will not increase the Goodput.

Let $AIFS, BO, Preamble$ and $SIFS$ denote the length, in μs, of the AIFS, BackOff, PHY Preamble and SIFS time intervals, and $BAck, CF\text{-}End$ denote, in μs, the transmission times of the BAck and CF-End control frames respectively. Let $H = MacDelimiter + MacHeader + FCS$ be the total length of the MAC Delimiter, MAC Header and FCS fields of an MPDU in bytes respectively. We assume that the MAC Header is of 28 bytes and the FCS is 4 bytes. Therefore, $H = 36$ bytes.

Since there are no collisions when using RD, holds $BO = \dfrac{(CW_{min} - 1)}{2} \cdot SlotTime$, where we refer to the CW_{min} of the AP. See **Table 1**. We now define C to be $C = AIFS + BO + SIFS + CF\text{-}End + Preamble$. The last $Preamble$ in C is the one preceding the transmission of the station.

Let $T(AP)$ and $T(STA)$ be the transmission times of the AP and the station's A-MPDU frames respectively. $T(AP)$ is given by the following (the details of how Equations (3)-(5) are derived can be found in [40]):

$$T(AP) = 4 \cdot \left\lceil \frac{K_D \cdot (L'_{DATA} \cdot 7 + H) \cdot 8 + 22}{4 \cdot R} \right\rceil \tag{3}$$

and $T(STA)$ is as follows:

$$T(STA) = 4 \cdot \left\lceil \frac{(n \cdot K_D \cdot 7 \cdot L'_{Ack} + K_A \cdot H) \cdot 8 + 22}{4 \cdot R} \right\rceil \tag{4}$$

where K_A is the number of MPDUs in the station's A-MPDU frame and $K_A = \left\lceil \frac{n \cdot K_D \cdot 7}{178} \right\rceil$.

The length $cycle$ of a TXOP is therefore given by

$$cycle = C + n\left(Preamble + T(AP) + SIFS + BAck + SIFS\right) + T(STA) + SIFS + BAck \tag{5}$$
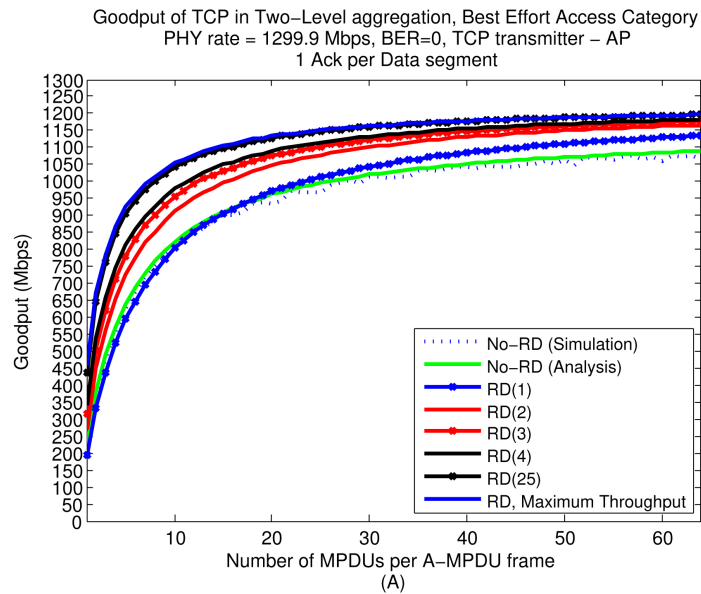
and the Goodput of the system is

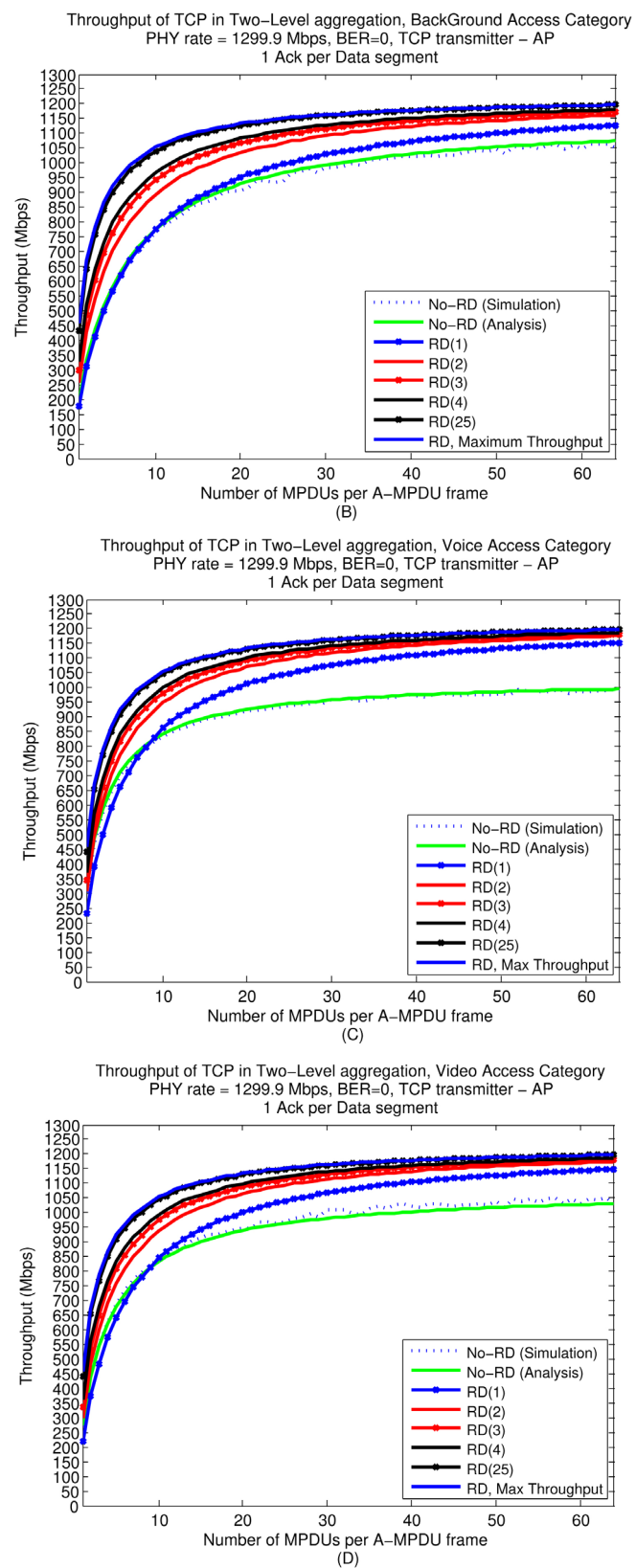$$Goodput = \frac{n \cdot K_D \cdot 7 \cdot (L_{DATA} \cdot 8)}{cycle} \tag{6}$$

Neglecting the rounding of $T(AP), T(STA)$ and K_A the Goodput can be written as :

$Goodput$

$$= \frac{n \cdot K_D \cdot 7 \cdot (L_{DATA} \cdot 8)}{C + SIFS + BAck + n \cdot (Preamble + 2 \cdot SIFS + BAck) + \frac{n \cdot K_D (L'_{DATA} \cdot 7 + H) \cdot 8 + 22}{R} + \frac{(n \cdot K_D \cdot 7 \cdot L'_{Ack} + K_A \cdot H) \cdot 8 + 22}{R}} \tag{7}$$

$$= \frac{L_{DATA} \cdot 8}{\frac{C + SIFS + BAck}{n \cdot K_D \cdot 7} + \frac{Preamble + 2 \cdot SIFS + BAck}{K_D \cdot 7} + \frac{8}{7R}(L'_{DATA} \cdot 7 + H) + \frac{22 \cdot 2}{n \cdot K_D \cdot 7 \cdot R} + \frac{8 \cdot L'_{Ack}}{R} + \frac{8 \cdot H}{178 \cdot R}}$$

One can see that as n increases and/or K_D increases, so does the Goodput. Notice that since BER = 0, the MAC ARQ protocol does not impose any limitation on the number K_D of MPDUs that are transmitted by the TCP transmitter in every transmission, as long as $K_D \le 64$. Also, it is most efficient to contain 7 MSDUs in every MPDU because this choice best amortizes the PHY/MAC overheads over the MSDUs.

In **Figures 6(A)-(D)** we show the Goodput results for the BK, BE, VI and VO ACs respectively, as a function of K_D. These results are derived from Equation (7) and were validated by simulation. The simulation is carried out by a software that we wrote and it is verified by analysis using a Markov chain model (in the **Appendix**).

Goodput of TCP in Two–Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter – AP
1 Ack per Data segment

(A)

Throughput of TCP in Two−Level aggregation, BackGround Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per Data segment

Throughput of TCP in Two−Level aggregation, Voice Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per Data segment

Throughput of TCP in Two−Level aggregation, Video Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per Data segment

Figure 6. The Goodput of the various ACs with and without RD, 1 TCP Ack per 1 TCP Data segment, BER = 0.

In every graph there are curves for $n = 1, 2, 3, 4, 25$. There is also a curve for the Goodput in the operation mode where RD is not used, *i.e.* the AP and the station use the "regular" IEEE 802.11ac MAC and compete for the channel in every transmission attempt. We denote this scenario by *No-RD*. This curve was obtained by simulation and the station always tries to transmit as many MPDUs as it can (up to 64), that are in its Transmission Queue at the time it acquires the right to transmit. The results for the *No-RD* scenario are also validated by an analysis based on a Markov chain. In the Appendix we present the Markov chain and its design.

We see in all the graphs that as the number of transmissions increases and/or as K_D increases, so does the Goodput. We also include a curve showing the maximum possible Goodput using RD. This curve is obtained as follows: For every K_D we first find the maximum number of possible transmissions, n_{max}, such that $n_{max} = \left\lfloor \dfrac{178 \times 64}{7 \cdot K_D} \right\rfloor$. Recall that 64×178 is the maximum number of TCP Acks that the receiver can transmit in a TXOP. Then, we compute the received Goodput for n_{max} using Equation (7). For example, for $K_D = 64$ holds that $n_{max} = 25$ and for $K_D = 1$ holds $n_{max} = \left\lfloor \dfrac{167 \times 64}{7 \times 1} \right\rfloor = 1627$.

Notice that in the VI and VO ACs and for K_Ds larger than 15, the difference in performance between *No-RD* and using RD is the largest among all the ACs. This happens because in these ACs CW_{min} and CW_{max} are the smallest among the ACs and so the probability for collisions is the largest. In large K_Ds collisions waste relatively long intervals of time and so the decrease in the Goodput is significant. As CW_{min} and CW_{max} decrease, the difference between using RD and *No-RD* increases. Notice that in VO $CW_{min} = 4$ and so the collision probability is 25%. In VI $CW_{min} = 8$ and the collision probability is 12.5%. For the BK and BE $CW_{min} = 16$ and the collision probability is only 6.25%.

For smaller K_Ds, *i.e.* $1 \leq K_D \leq 15$, notice that *No-RD* sometimes outperforms $RD(1)$. In BK and BE the collision probability is small and the AP and the station transmit almost alternately. Therefore, *No-RD* and $RD(1)$ have almost the same performance, except that in $RD(1)$ there is an extra overhead of *CF-End* and *SIFS* at the end of every TXOP.

As the value of AIFS is larger, this overhead is less significant. In BE the AIFS is 43 μs compared to 79 μs in BK and therefore the $CF\text{-}End + SIFS = 26 + 16 = 48\,\mu s$ is more significant in BE and *No-RD* slightly out-performs $RD(1)$, while in BK they perform equally.
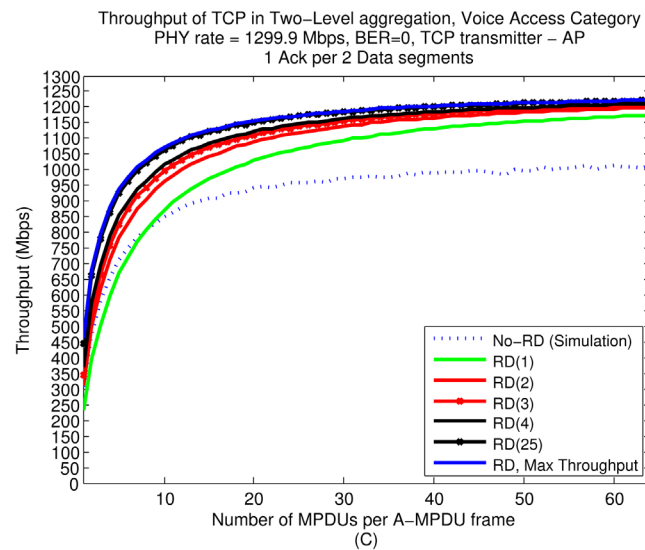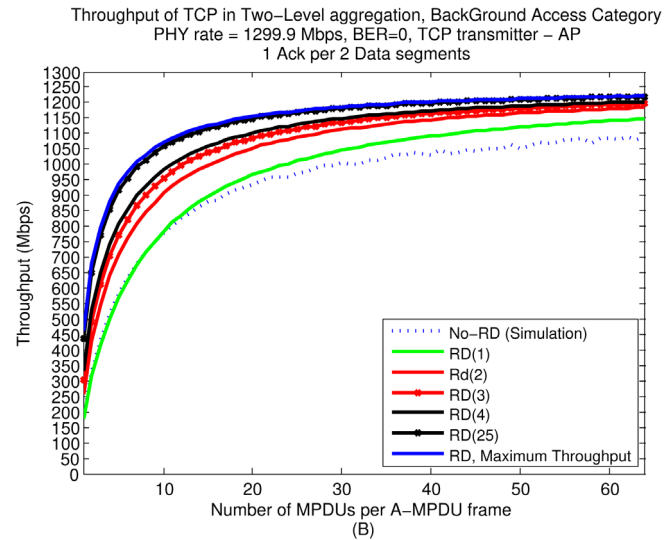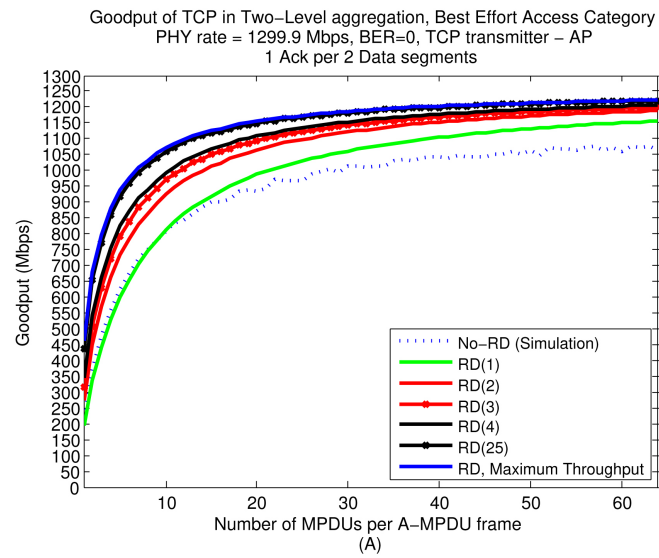
In VI and VO the AIFS is smaller than in BK and BE and so the $CF\text{-}End + SIFS$ overhead is more significant. Moreover, the AP in these ACs has a higher probability of accessing the channel than the station because its AIFS is shorter by one *Slot-Time*. This enables the AP in *No-RD* to transmit several times in a row before the station replies. This also enables a better Goodput in *No-RD* than in $RD(1)$ where the AP and the station transmit alternately. On the other hand, the collision probability is larger in VI and VO. However, the AP transmits many times without competition in *No-RD* when the TCP receiver has no TCP Acks to transmit. The overall outcome is a slightly larger Goodput in *No-RD*, compared to $RD(1)$, than in BK and BE.
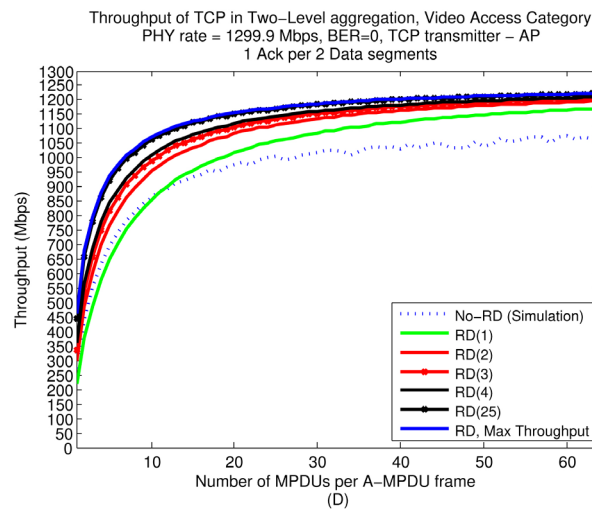
In **Figure 7** we show the same results as in **Figure 6** but now every TCP Ack acknowledges two TCP Data segments, a feature known as *TCP Delayed Acks*. For clarity, for the *No-RD* scheme we only show the simulated results. The analytical results are similar, as can be seen in **Figure 6**. Normally, the TCP receiver does not send an Ack the instant it receives data. Instead, it delays the Ack, hoping to have data going in the same direction as the Ack, so the Ack can be sent along with the data. This delay is usually in the order of 200 μs. However, if meanwhile another data segment arrives, the TCP receiver immediately generates an Ack to send.

Using TCP Delayed Acks enables the TCP transmitter to transmit more TCP Data segments in one TXOP: the limiting condition is now $n \cdot K_D \cdot 7 \leq 2 \times 64 \times 178$. Comparing between the curves of the Maximum Goodputs in the cases of with and without TCP Delayed Acks reveals an improvement of only about 2% in the Goodput for large K_Ds in the case of TCP Delayed Acks. The reason for the small improvement can be understood from Equation (7): the main impact of TCP Delayed Acks is in enabling more transmissions of the TCP transmitter during a TXOP. However, increasing the number of transmissions n does not increase the Goodput significantly for "large" n's.
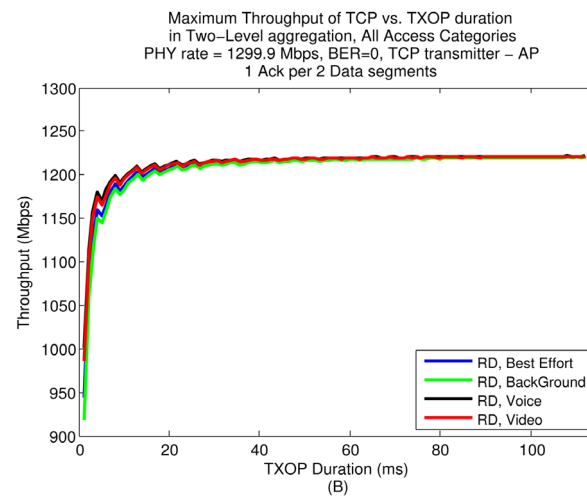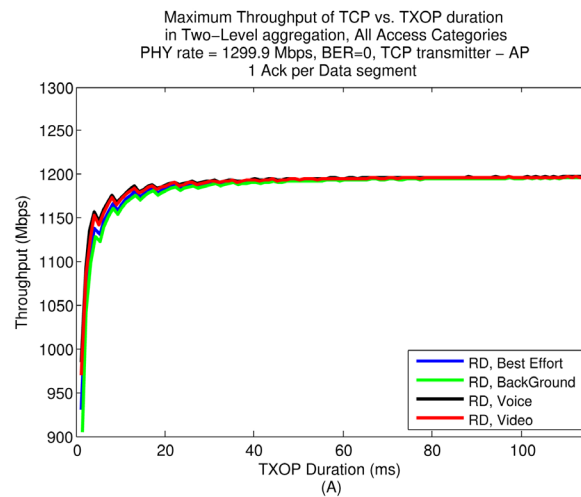
Finally, in **Figure 8** we show the Goodput of the various ACs as a function of the TXOP duration. In **Figure 8(A)** and **Figure 8(B)** we assume that TCP Delayed Acks are not in use and in use respectively. The curves were computed as follows: For every S TCP Data segments, $1 \leq S \leq 64 \times 178$ ($2 \times 64 \times 178$ for TCP Delayed Acks)

Goodput of TCP in Two−Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per 2 Data segments

(A)

Throughput of TCP in Two−Level aggregation, BackGround Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per 2 Data segments

(B)

Throughput of TCP in Two−Level aggregation, Voice Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per 2 Data segments

(C)

Throughput of TCP in Two−Level aggregation, Video Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per 2 Data segments

Figure 7. The Goodput of the various ACs with and without RD, 1 TCP Ack per 2 TCP Data segment (TCP Delayed Acks), BER = 0.

Maximum Throughput of TCP vs. TXOP duration
in Two−Level aggregation, All Access Categories
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per Data segment

Maximum Throughput of TCP vs. TXOP duration
in Two−Level aggregation, All Access Categories
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter − AP
1 Ack per 2 Data segments

Figure 8. The maximum Goodput vs. the duration of the TXOPs in the various ACs, with and without TCP Delayed Acks, BER = 0.

we explored the most efficient way (Maximum Goodput) to transmit these segments, and the achieved Goodput. The most efficient way is to transmit as many MPDUs as possible in a single transmission, up to 64, while every MPDU contains 7 MSDUs except the last one, when S is not divided by 7. We then checked the cycle length, and how the curves relate between cycles' lengths and Goodputs. For example, if we find that transmitting S TCP Data segments with the largest Goodput G takes Cms, it is easy to verify that G is the largest Goodput possible in Cms.
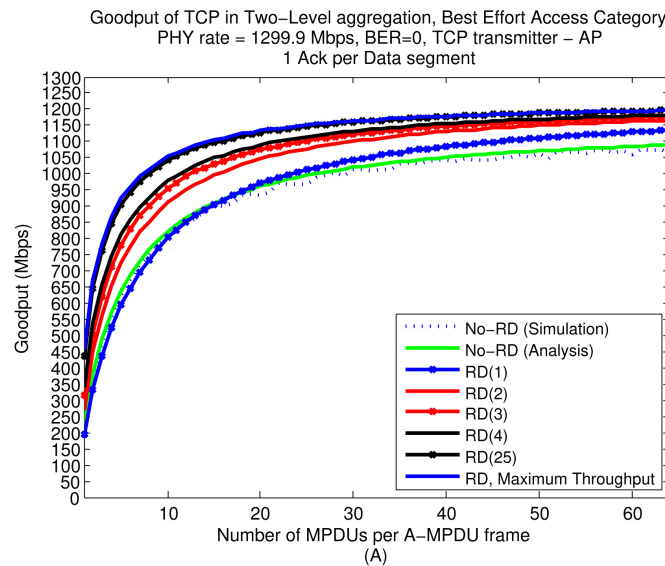
We see that all the ACs achieve the same Goodput for "long" TXOP. This happens because the cycles in the various ACs differ only in the AIFS and BackOff time intervals which become negligible in long cycles. In shorter cycles the VI and VO ACs achieve the same best performance because their AIFS are the shortest, 25 μs for the AP. BE outperforms BK because its AIFS is 43 μs (AP) compared to 79 μs in BK. See **Table 1**.
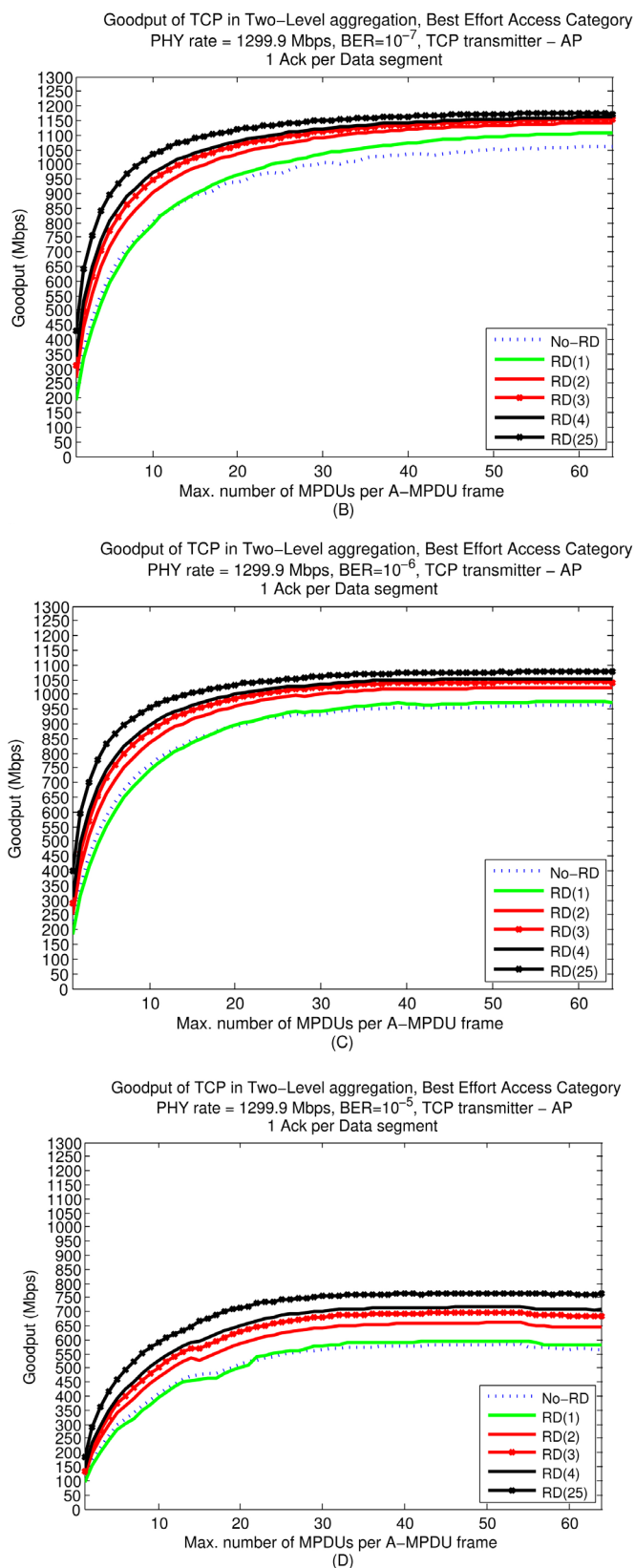
There are two important outcomes from **Figure 8**. First, using a TXOP of 20 - 30 ms is sufficient to achieve almost the largest Goodput possible. This is important since it enables short time-outs in the TCP protocol and so the TCP transmitter can receive TCP Acks sooner, while still using the wireless channel efficiently. Second, in a scenario where there are several TCP connections between the AP and several stations, it is sufficient for the AP to use TXOPs of 20 - 30 ms in order to use the channel efficiently. This has an impact on the fairness among the stations and is the basis for further research.

5. Error-Prone Channel Results

In this Section we assume the BERs of $10^{-5}, 10^{-6}$ and 10^{-7}. We concentrate only on the BE AC. The results for the other ACs are similar, with the same differences compared to BE as described in Section 4. In **Figures 9(A)-(D)** we show the Goodput vs. the maximum number K_D of MPDUs per transmission of the TCP transmitter in the BE AC for $BER = 0, 10^{-7}, 10^{-6}, 10^{-5}$ respectively. First notice that the title of the X-axis in **Figure 9(A)** is different than those of parts (B), (C) and (D). This is because the positive BER can cause the MAC TW to limit the number of transmitted MPDUs in a single transmission to be smaller than K_D, and so K_D is only the maximum allowed MPDUs in a single transmission.

In general using RD results in a larger Goodput. Notice however that as the BER increases, the advantage of $RD(1)$ over *No-RD* decreases. As the BER increases, the number of MPDUs that the TCP transmitter is able to transmit in every transmission decreases. The MAC TW is not always able to slide so that it will contain K_D MPDUs, *i.e.* the maximum allowed number of MPDUs. This results in two outcomes: First, as the BER increases a smaller number of MPDUs are transmitted in *No-RD* and $RD(1)$, and a smaller number of MPDUs arrive successfully at the TCP receiver in both schemes. However, shorter A-MPDU frames have an advantage in *No-RD* because the penalty of collisions is smaller. These two outcomes cause $RD(1)$ and *No-RD* to coincide as the BER increases.

Goodput of TCP in Two−Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter – AP
1 Ack per Data segment

(A)

Goodput of TCP in Two−Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10⁻⁷, TCP transmitter − AP
1 Ack per Data segment

(B)

Goodput of TCP in Two−Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10⁻⁶, TCP transmitter − AP
1 Ack per Data segment

(C)

Goodput of TCP in Two−Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10⁻⁵, TCP transmitter − AP
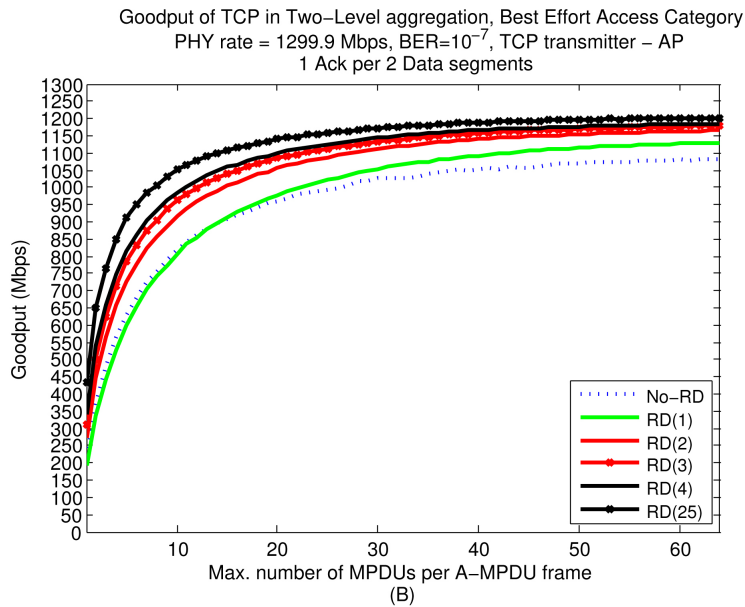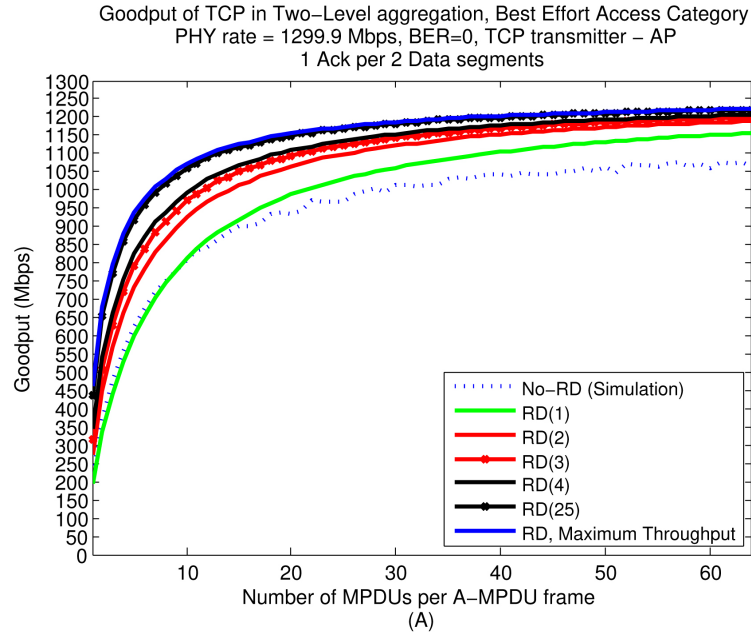1 Ack per Data segment

(D)

Figure 9. The Goodput of the Best Effort AC with and without RD for various BERs, 1 TCP Ack per 1 TCP Data segment.
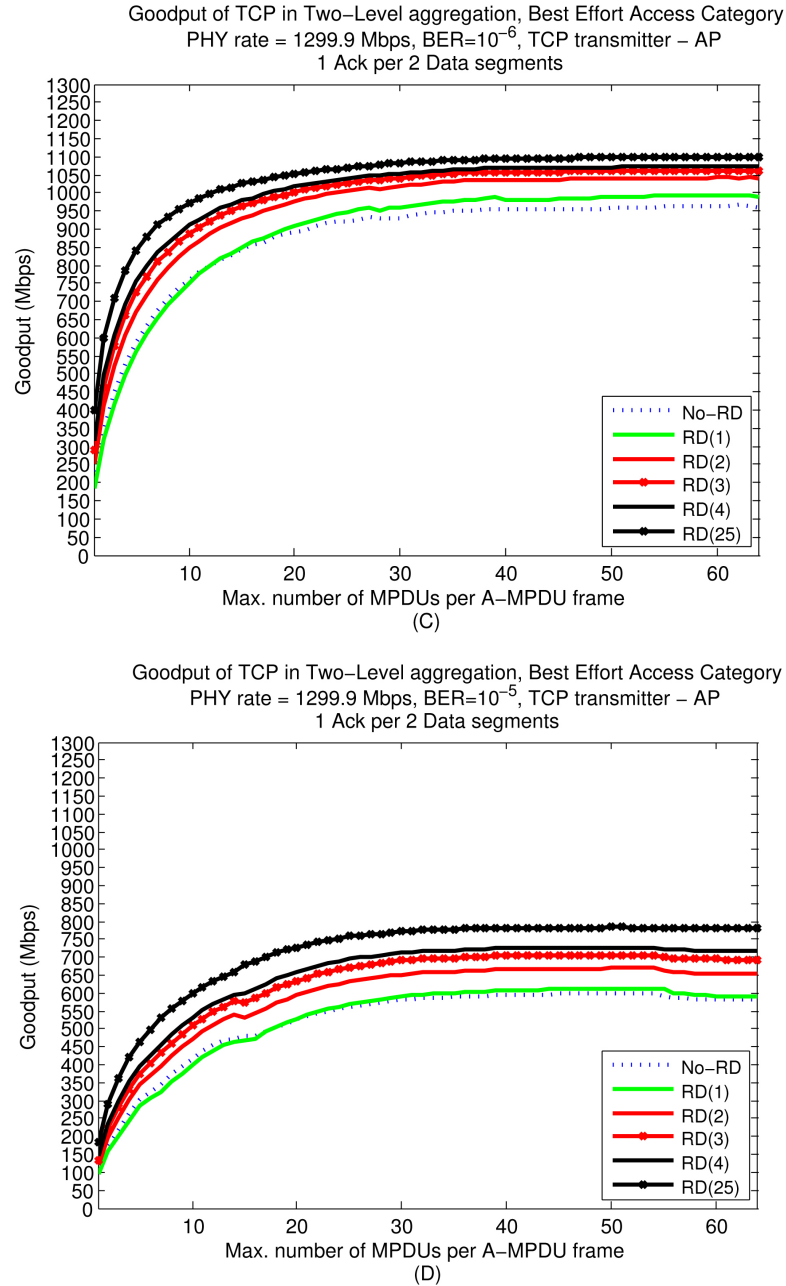
In $RD(2)$ there is a second transmission in every TXOP which increases the probability that MPDUs arrive successfully. Thus the MAC TW slides faster, enabling more successful transmissions of MPDUs. This causes a significant improvement in the Goodput of $RD(2)$ compared to *No-RD*.

Notice that **Figure 9(A)** is for BER = 0 and it is the same as **Figure 6(A)**. In **Figure 9(A)** we can provide a curve showing the maximum possible Goodput. However, for BER > 0, in order to find such a curve one needs to know, given K_D, the actual average number of transmitted MPDUs in every transmission of the TCP transmitter. This number might be smaller than K_D, especially for large K_Ds, because it is possible that the MAC TW does not contain K_d MPDUs. Such a computation is difficult [25] [46] and it is out of the scope of this paper. This is also the reason why we cannot provide analytical results for the *No-RD* scheme as for the case BER = 0. Notice again that for small K_Ds *No-RD* slightly outperforms $RD(1)$ for the same reasons given for this phenomena in Section 4.

In **Figure 10** we show the same results as in **Figure 9**, but now there is a use in the TCP Delayed Acks. Using
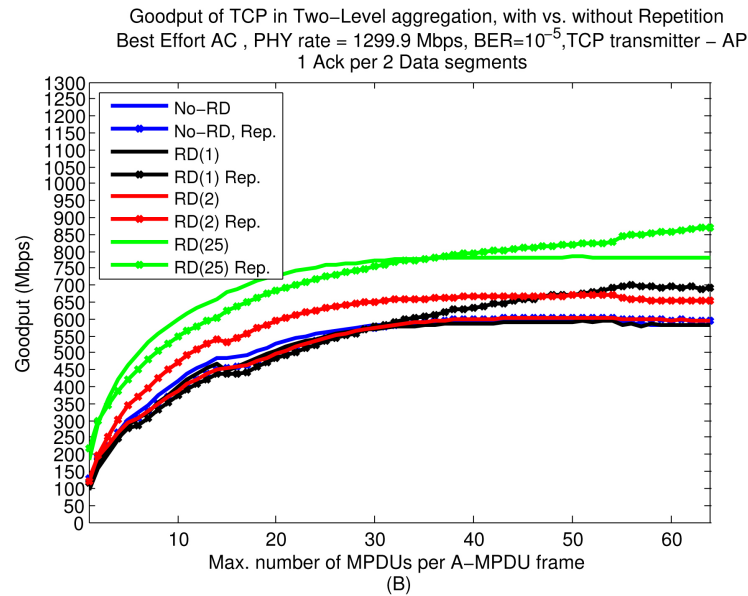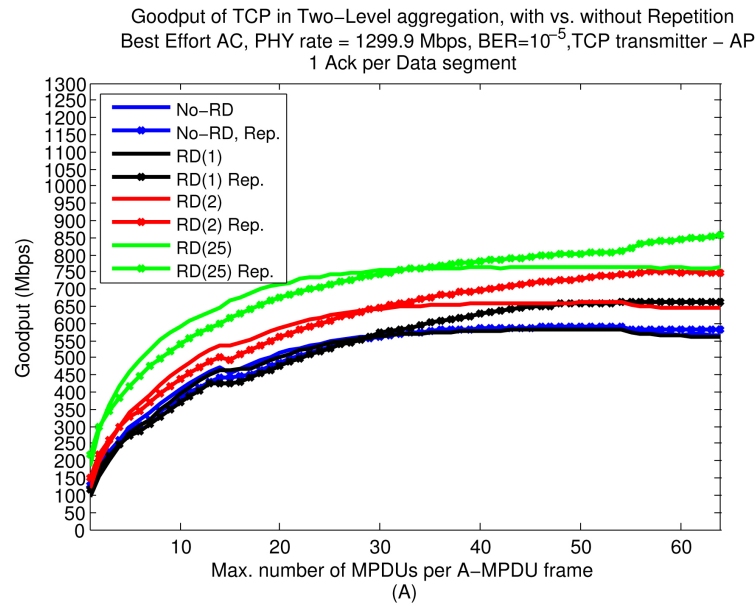

Goodput of TCP in Two–Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=0, TCP transmitter – AP
1 Ack per 2 Data segments

(A)

Goodput of TCP in Two–Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10^{-7}, TCP transmitter – AP
1 Ack per 2 Data segments

(B)

Goodput of TCP in Two–Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10^{-6}, TCP transmitter – AP
1 Ack per 2 Data segments

(C)

Goodput of TCP in Two–Level aggregation, Best Effort Access Category
PHY rate = 1299.9 Mbps, BER=10^{-5}, TCP transmitter – AP
1 Ack per 2 Data segments

(D)

Figure 10. The Goodput of the Best Effort AC with and without RD for various BERs, 1 TCP Ack per 2 TCP Data segments (TCP Delayed Acks).
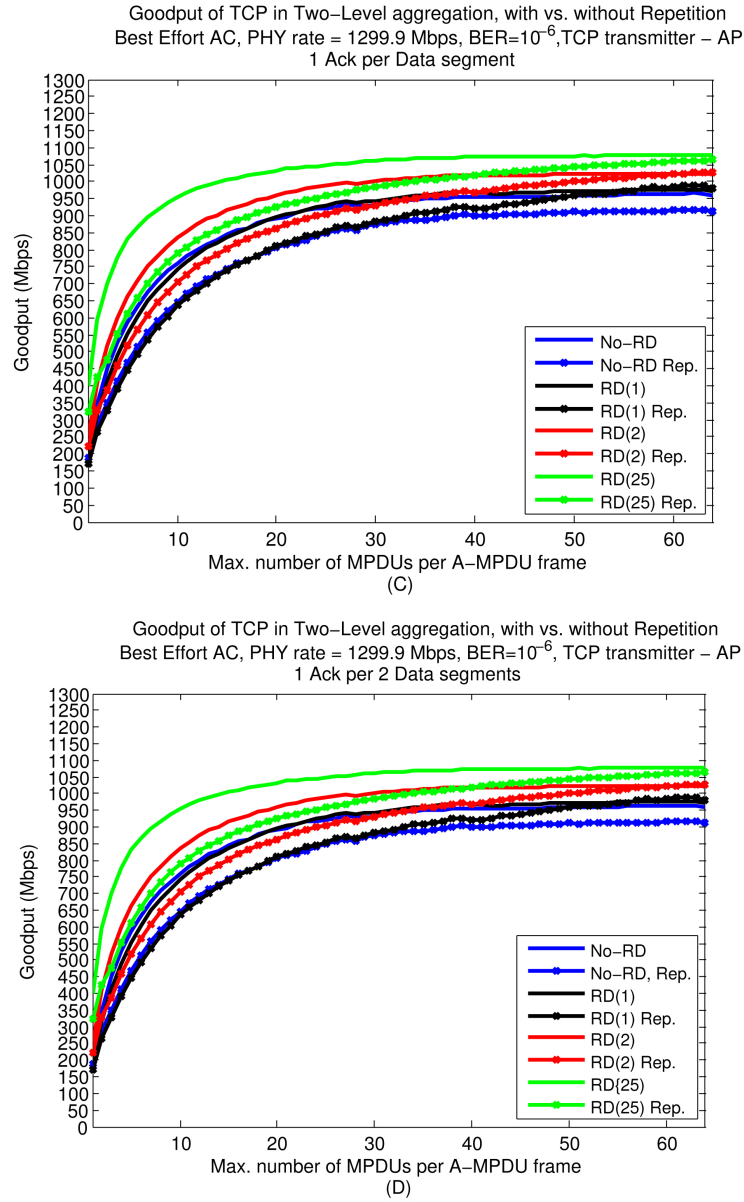
TCP Delayed Acks does not improve the performance of the *No-RD* scheme because in the case of collisions, the time wasted is the time of transmitting the TCP Data segments. The shorter time of transmitting the TCP Acks has no influence in this case. On the other hand, in the schemes that use RD the reduced time of transmitting TCP Acks has an influence because the TXOP length is shorter. Therefore, one can see that the difference between the performance of the RD schemes to that of *No-RD* is larger than in the case of not using TCP Delayed Acks.

In **Figure 11** we show the use in the scheme of [22] where each of the first 3 MPDUs in every A-MPDU frame of the TCP transmitter is transmitted twice, *i.e.* MPDU repetition. Only the first 3 MPDUs are transmitted twice because it is the most efficient scheme (max. Goodput), as was shown in [22]. This scheme, which we

denote $Repetition(Rep.)$, has two effects. First, it increases the arrival success probabilities of the first 3 MPDUs in the A-MPDU frames. As an outcome, it enables the MAC TW to slide faster and to contain more MPDUs ready for transmission, compared to the case of not using Rep. On the negative side the transmission time of the A-MPDU frames increases by transmitting the first 3 MPDUs twice. Therefore, for BER = 0 it is clear that the performance of $Rep.$ is worse than not using it. As the BER increases, the advantage of $Rep.$ increases. We also found that for BER = 10^{-7} it is inefficient to use $Rep.$ However, for BER = 10^{-5}, 10^{-6} $Rep.$ improves the achieved Goodput as we show in **Figure 11**.

In order to demonstrate the improvement consider **Figure 11(A)** for BER = 10^{-5} without TCP Delayed Acks. One can see that all the schemes, namely No-RD, $RD(1)$, $RD(2)$ and $RD(25)$, benefit from using $Rep.$ in the case of large K_Ds, while for small K_Ds it is not efficient. Notice that in the case of small K_Ds the probability that the MAC TW will contain K_D MPDUs ready for transmission is much larger than in the case of larger K_Ds. Therefore using $Rep.$ in the former case only increases the transmission time of the TCP transmitter A-MPDU frames with no benefit.
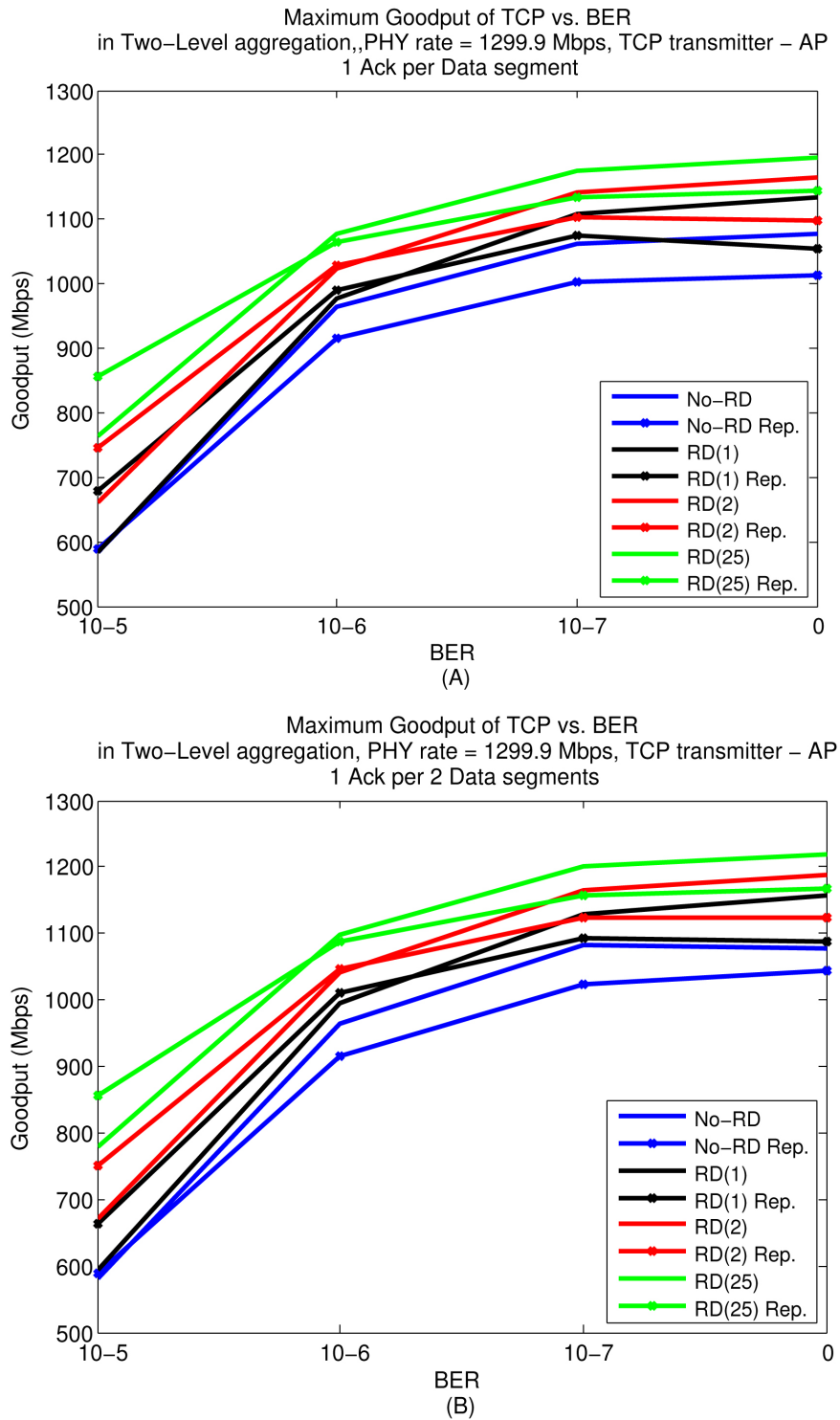

Goodput of TCP in Two−Level aggregation, with vs. without Repetition
Best Effort AC, PHY rate = 1299.9 Mbps, BER=10^{-5},TCP transmitter − AP
1 Ack per Data segment

(A)

Goodput of TCP in Two−Level aggregation, with vs. without Repetition
Best Effort AC , PHY rate = 1299.9 Mbps, BER=10^{-5},TCP transmitter − AP
1 Ack per 2 Data segments

(B)

Figure 11. Comparing transmissions with and without *Rep.* in the Best Effort AC.

In **Figure 12** we show the maximum received Goodputs vs. the BER for the *No-RD*, *RD*(1), *RD*(2) and *RD*(25) schemes. In **Figure 12(A)** and **Figure 12(B)** we consider the cases without and with TCP Delayed Acks respectively. We see that for every BER, using RD is more efficient than not using RD. For BER = 10^{-5} and in several cases when BER = 10^{-6} using *Rep.* even improves the Goodput further. For example, in BER = 10^{-5} the Goodput of *RD*(25) is 780 Mbps, compared to 600 Mbps in the *No-RD* case. With using *Rep.* the Goodput of *RD*(25) is 860 Mbps, over 40% improvement compared to *No-RD*. For BER = 10^{-7} and BER = 0 using *Rep.* decreases the performance for the reasons mentioned previously.
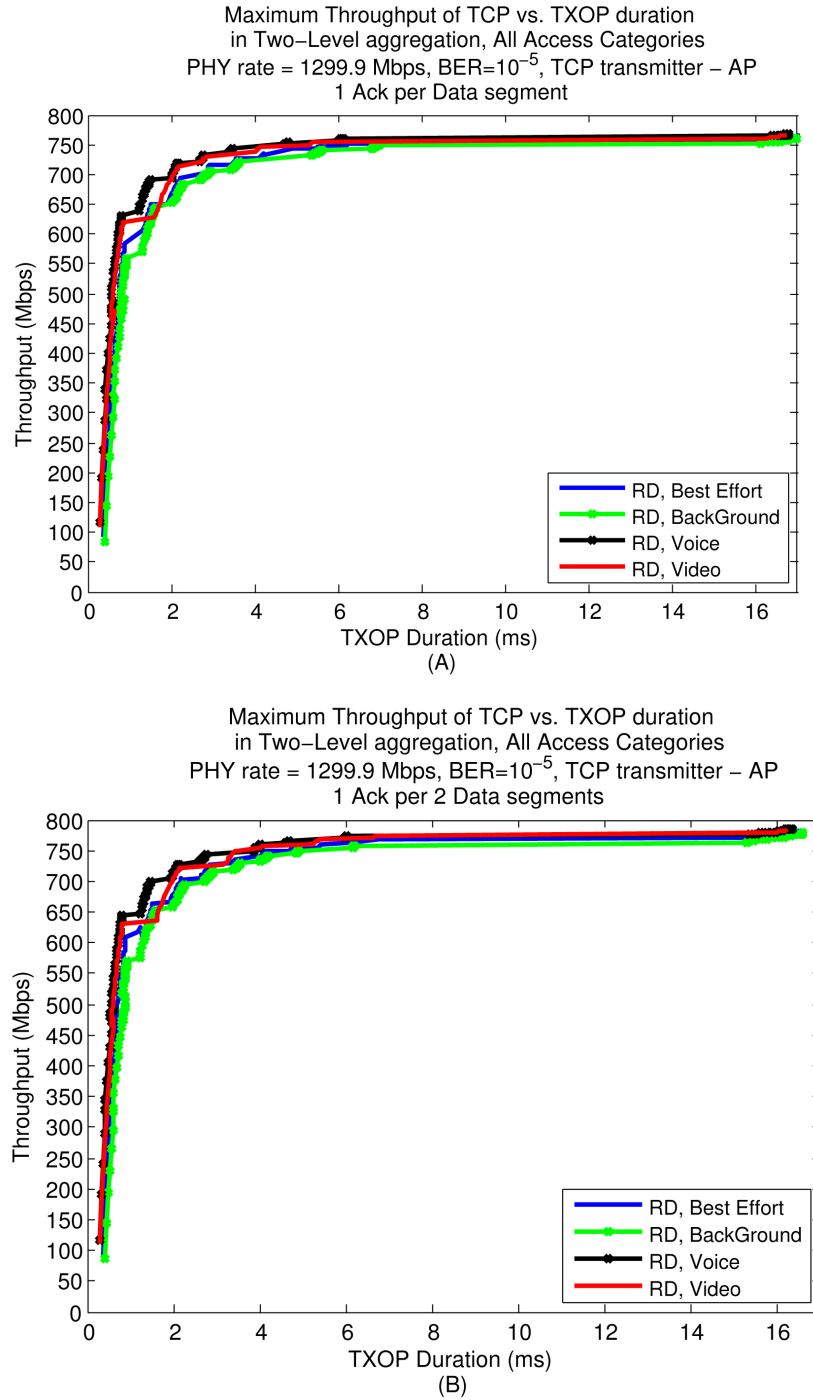
Finally, in **Figure 13** we show the maximum received Goodput as a function of the TXOP for BER = 10^{-5}. Recall that for BER > 0 it is difficult to find the number of actually transmitted MPDUs in every transmission of the TCP transmitter. Therefore, we cannot use the same technique to compute the maximum Goodput as in Section 4, BER = 0.

Instead, we computed the average TXOP duration for $RD(1), RD(2), \cdots, RD(25)$. For every received Goodput G we looked for the $RD(n), 1 \leq n \leq 25$ that achieves G with the shortest TXOP duration.

Figure 12. The maximum Goodput in the various transmission schemes vs. BER, for the Best Effort AC.

The outcomes and conclusions are similar in trend to those in **Figure 8** except that the achieved Goodputs are much lower because of the positive BER. On the other hand the delays, *i.e.* the length of the TXOPs, are shorter. In BER = 0 there is no benefit to using TXOPs of more than 20 μs while for BER = 10^{-5} there is no benefit to

Maximum Throughput of TCP vs. TXOP duration
in Two–Level aggregation, All Access Categories
PHY rate = 1299.9 Mbps, BER=10^{-5}, TCP transmitter – AP
1 Ack per Data segment

(A)

Maximum Throughput of TCP vs. TXOP duration
in Two–Level aggregation, All Access Categories
PHY rate = 1299.9 Mbps, BER=10^{-5}, TCP transmitter – AP
1 Ack per 2 Data segments

(B)

Figure 13. The maximum Goodput vs. the duration of the TXOPs in the various ACs, with and without TCP Delayed Acks, BER = 0.

using TXOPs of more than 4 μs.

6. Summary

This paper shows an example of the benefit achieved when different layers in the protocol stack co-operate. In particular, we show the improvement in the TCP Goodput that is achieved when the MAC layer of the IEEE

802.11ac standard is aware of TCP traffic. Using Reverse Direction, the contention between the TCP transmitter and receiver is eliminated, and no time is wasted due to collisions.

Using also the Two-Level aggregation scheme, in an error-free channel the TCP Goodput is improved by 20% compared to contension based channel access. In an error-prone channel the TCP Goodput is improved by 60% also blindly using retransmission of frames in A-MPDU frames.

This paper assumes only one TCP connection in the system, which is a scenario that is possible in small systems such as in the Home environment. A next research step is to investigate the performance of Reverse Direction and aggregation when the AP maintains several TCP connections at the same time.

References

[1] IEEE P802.11-REVmc TM/D4.3. IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, New York.

[2] IEEE Std. 802.11ac TM-2013. IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Require-ments. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specific requirements. Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. IEEE, New York.

[3] IEEE Std. 802.11 TM-2012. Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE, New York.

[4] Pilosof, S., Ramjee, R., Raz, D., Shavitt, Y. and Sinha, P. (2003) Understanding TCP Fairness over Wireless LAN. *Proceedings of the* 22*nd Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, 30 March-3 April 2003, 863-872. http://dx.doi.org/10.1109/infcom.2003.1208924

[5] Miorandi, D., Kherani, A.A. and Altman, E. (2006) A Queueing Model for HTTP Traffic over IEEE802.11 WLANs. *Computer Networks*, **50**, 63-79. http://dx.doi.org/10.1016/j.comnet.2005.04.004

[6] Yu, J. and Choi, S. (2007) Modeling and Analysis of TCP Dynamics over IEEE 802.11 WLAN. 4*th Annual Conference on Wireless on Demand Network Systems and Services*, Oberguyrgl, 24-26 January 2007, 154-161. http://dx.doi.org/10.1109/wons.2007.340482

[7] Keceli, E., Inan, I. and Ayanoglu, E. (2008) Fair and Efficient TCP Access in IEEE 802.11 WLANs. *IEEE Wireless Communications and Networking Conference*, Las Vegas, 31 March-3 April 2008, 1745-1750. http://dx.doi.org/10.1109/wcnc.2008.311

[8] Bruno, R., Conti, M. and Gregori, E. (2005) Throughput Analysis of UDP and TCP Flows in IEEE 802.11b WLANs: A Simple Model and Its Validation. *Workshop on Techniques*, *Methodologies and Tools for Performance Evaluation of Complex Systems*, 19 September 2005, 54-63. http://dx.doi.org/10.1109/FIRB-PERF.2005.20

[9] Bruno, R., Conti, M. and Gregori, E. (2008) Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows. *IEEE Transactions on Mobile Computing*, **7**, 171-186. http://dx.doi.org/10.1109/TMC.2007.70718

[10] Bruno, R., Conti, M. and Gregori, E. (2005) Modeling TCP Throughput over Wireless LANs. *Proceedings of* 17th *IMACS World Congress Scientific Computation*, *Applied Mathematics and Simulation*, Paris, 11-15 July 2005, 626-637.

[11] Statkus, A. and Paulikas, S. (2013) Improving TCP Performance in IEEE 802.11 Networks. *Elektronica IR Elektrotechnika*, **19**, 99-102. http://dx.doi.org/10.5755/j01.eee.19.5.2571

[12] Ng, A.C.H., Malone, D. and Leith, D.J. (2005) Experimental Evaluation of TCP Performance and Fairness in an 802.11e Test-Bed. *Proceedings of ACM SIGCOMM Workshop on Experimental Approaches to Wireless Networks Design and Analysis*, Philadelphia, 22-26 August 2005, 17-22. http://dx.doi.org/10.1145/1080148.1080152

[13] Leith, D.J., Clifford, P., Malone, D. and Ng, A.C.H. (2005) TCP Fairness in 802.11e WLANs. *IEEE Communications Letters*, **9**, 964-966. http://dx.doi.org/10.1109/LCOMM.2005.11004

[14] Kim, M., Park, E.C. and Kim, W. (2013) TCP Acknowledgement Compression for Fairness among Uplink TCP Flows in IEEE 802.11n WLANs. *Journal of Institute of Control*, *Robotics and Systems*, **19**, 653-660. http://dx.doi.org/10.5302/J.ICROS.2013.13.1926

[15] Sharon, O. and Alpert, Y. (2016) Coupled IEEE 802.11ac and TCP Performance Evaluation in Various Aggregation Schemes and Access Categories. *Computer Networks*, **100**, 141-156. http://dx.doi.org/10.1016/j.comnet.2016.02.023

[16] Yu, J., Choi, S. and Qiao, D. (2002) Analytical Study of TCP Performance over IEEE 802.11e WLANs. *Mobile Networks and Applications*, **14**, 470-485. http://dx.doi.org/10.1007/s11036-008-0111-z

[17] Kumar, A., Altman, E., Miorandi, D. and Goyal, M. (2007) New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs. *IEEE/ACM Transactions on Networking*, **15**, 588-601. http://dx.doi.org/10.1109/TNET.2007.893091

[18] Lynn, S. (2014) The   Best NAS   for Most Home Users. http://thewirecutter.com/reviews/best-network-attached-storage/

[19] Ha, S., Rhee, I. and Xu, L. (2008) CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating Systems Review*, **42**, 64-74. http://dx.doi.org/10.1145/1400097.1400105

[20] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M.Y. and Wang, R. (2001) TCP Westwood: Band-Width Estimation for Enhanced Transport over Wireless Links. *Proceedings of the* 7*th Annual International Conference on Mobile Computing and Networking*, Rome, 16-21 July 2001, 287-297. http://dx.doi.org/10.1145/381677.381704

[21] Henderson, T., Floyd, S., Gurtov, A. and Nishida, Y. (2002) The NewReno Modification to TCP's Recovery Algorithm. Request for Comments (RFC) 6582. https://www.rfc-editor.org/rfc/rfc6582.txt

[22] Sharon, O, and Alpert, Y. (2016) A New Aggregation Based Scheduling Method for Rapidly Changing IEEE 802.11ac Wireless Channels.

[23] Sidelnikov, A. and Choi, J.S. (2006) Fragmentation/Aggregation Scheme for Throughput Enhancement of IEEE 802.11n WLAN. *Proceedings of IEEE Asia Pacific Wireless Communications Symposium*, Dacjon, 24-25 August 2006.

[24] Lin, Y. and Wong, V.W.S. (2006) Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs. *IEEE Global Telecommunications Conference*, San Francisco, 27 November-1 December 2006, 1-6. http://dx.doi.org/10.1109/glocom.2006.925

[25] Ginzbyrg, B. and Kesselman, A. (2007) Performance Analysis of A-MPDU and A-MSDU Aggregation in IEEE 802.11n. *IEEE Sarnoff Symposium*, Nassau Inn, Princeton, 30 April-2 May 2007, 1-5. http://dx.doi.org/10.1109/sarnof.2007.4567389

[26] Skotdoulis, D., Ni, Q., Chen, H.H., Stephens, A.P., Kiu, C. and Jamalipour, A. (2008) IEEE 802.11n MAC Frame Aggregation Mechanisms for Next-Generation High-Throughput WLANS. *IEEE Wireless Communications*, **15**, 40-47. http://dx.doi.org/10.1109/MWC.2008.4454703

[27] Kim, B.S., Huong, H.Y. and Sung, D.K. (2008) Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n. *Wireless Communications and Networking Conference*, Las Vegas, 31 March-3 April 2008, 1740-1744. http://dx.doi.org/10.1109/wcnc.2008.310

[28] Chan, K. (2009) Evaluation and Enhancements in 802.11n WLANs: Error Sensitive Adaptive Frame Aggregation. Master Thesis, San Jose State University, San Jose. http://scholarworks.sjsu.edu/etd_projects/79/

[29] Wang, C.Y. and Wei, H.Y. (2009) IEEE 802.11n MAC Enhancement and Performance Evaluation. *Mobile Networks and Applications*, **14**, 760-771. http://dx.doi.org/10.1007/s11036-008-0129-2

[30] Selvan, T. and Srikanth, S. (2010) A Frame Aggregation Scheduler for IEEE 802.11n. *National Conference on Communication*, Chennai, 29-31 January 2010, 1-5.

[31] Zielinski, B. (2011) Efficiency Analysis of IEEE 802.11 Protocol with Block Acknowledge and Frame Aggregation. *Bulletin of the Polish Academy of Sciences Technical Sciences*, **59**, 235-243. http://dx.doi.org/10.2478/v10175-011-0029-7

[32] Daldoul, Y., Ahmed, T. and Meddour, D.E. (2011) IEEE 802.11n Aggregation Performance Study for the Multicast. *IFIP Wireless Days*, Niagara Falls, 10-12 October 2011, 1-6. http://dx.doi.org/10.1109/wd.2011.6098211

[33] Saif, A., Othman, M., Subramaniam, S. and Hamid, N.A. (2012) An Enhanced A-MSDU Frame Aggregation Scheme for 802.11n Wireless networks. *Wireless Personal Communications*, **6**, 683-706. http://dx.doi.org/10.1007/s11277-011-0358-8

[34] Kolap, J., Krishnan, S. and Shaha, N. (2012) Frame Aggregation Mechnism for High-Throughput 802.11n WLANs. *International Journal of Wireless & Mobile Networks*, **4**, 141-153. http://dx.doi.org/10.5121/ijwmn.2012.4309

[35] Kim, Y., Monroy, E., Lee, O., Park, K.J. and Choi, S. (2012) Adaptive Two-Level Frame Aggregation in IEEE 802.11n WLAN. 18*th Asia-Pacific Conference on Communication*, Jeju Island, 15-17 October 2012, 658-663. http://dx.doi.org/10.1109/apcc.2012.6388276

[36] Park, N. (2011) IEEE 802.11ac: Dynamic Bandwidth Channel Access. *IEEE International Conference on Communications*, Kyoto, 5-9 June 2011, 1-5. http://dx.doi.org/10.1109/icc.2011.5963089

[37] Ong, E.H., Kneck, J., Alanen, O., Chang, Z., Huovinen, T. and Nihtila, T. (2011) IEEE 802.11ac: Enhancements for Very High Throughput WLANs. *IEEE* 22*nd International Symposium on Personal*, *Indoor and Mobile Radio Communications*, Toronto, 11-14 September 2011, 849-853.

[38] Chang, Z., Alanen, O., Huovinen, T., Nihtila, T., Ong, E.H., Kneckt, J. and Ristaniemi, T. (2012) Performance Analy-

sis of IEEE 802.11ac DCF with Hidden Nodes. *IEEE 75th Vehicular Technology Conference*, Yokohama, 6-9 May 2012, 1-5. http://dx.doi.org/10.1109/vetecs.2012.6240054

[39] Bellalta, B., Barcelo, J., Staehle, D., Vinel, A. and Oliver, M. (2012) On the Performance of Packet Aggregation in IEEE 802.11ac MU-MIMO WLANs. *IEEE Communication Letters*, **16**, 1588-1591. http://dx.doi.org/10.1109/LCOMM.2012.081612.120744

[40] Sharon, O. and Alpert, Y. (2014) MAC Level Throughput Comparison: 802.11ac vs. 802.11n. *Physical Communication*, **12**, 33-49. http://dx.doi.org/10.1016/j.phycom.2014.01.007

[41] Changwen, L. and Stephens, A.P. (2006) Delayed Channel Access for IEEE 802.11e Based WLAN. *IEEE International Conference on Communication*, Istanbul, June 2006, 4811-4817.

[42] Skordoulis, D., Ni, Q., Min, G. and Borg, K. (2008) Adaptive Delayed Channel Access for IEEE 802.11n WLANs. *IEEE International Conference on Circuits and Systems for Communications*, Shanghai, 26-28 May 2008, 167-171. http://dx.doi.org/10.1109/iccsc.2008.42

[43] Li, T., Ni, Q., Malone, D., Leith, D., Xiao, Y. and Turletti, T. (2009) Aggregation with Fragment Retransmission for Very High-Speed WLANs. *IEEE/ACM Transactions on Networking*, **17**, 591-604. http://dx.doi.org/10.1109/TNET.2009.2014654

[44] Skordoulis, D., Ni, Q. and Zarakovitis, C. (2009) A Selective Delayed Channel Access (SDCA) for the High-Throughput IEEE 802.11n. *IEEE Wireless Communication and Networking Conference*, Budapest, 5-8 April 2009, 1-8. http://dx.doi.org/10.1109/wcnc.2009.4917544

[45] Camps-Mur, D., Gomony, M.D., Perez-Costa, X. and Sllent-Ribes, S. (2012) Leveraging 802.11n Frame Aggregation to Enhance QoS and Power Consumption in Wi-Fi Networks. *Computer Networks*, **56**, 2896-2911. http://dx.doi.org/10.1016/j.comnet.2012.05.004

[46] Sharon, O. and Alpert, Y. (2015) The Combination of Aggregation, ARQ, QoS Guarantee and Mapping of Application Flows in Very High Throughput 802.11ac Networks. *Physical Communication*, **17**, 15-36. http://dx.doi.org/10.1016/j.phycom.2015.06.002

[47] Garcia, M.A., Santos, M.A. and Villalon, J. (2011) IEEE 802.11n MAC Mechanisms for High Throughput: A Performance Evaluation. *The 7th International Conference on Networking and Services*, Venice, 22-27 May 2011, 32-37.

[48] Milad, A.A., Noh, Z.A.B.M., Shibghatullah, A.S. and Algaet, M.A. (2013) Reverse Direction Transmission in Wireless Networks: Review. *Middle-East Journal of Scientific Research*, **18**, 767-778.

[49] Milad, A.A., Noh, Z.A.B.M., Shibghatullah, A.S., Algaet, M.A. and Mustaphs, A. (2014) Reverse Direction Transmission using Single Data Frame and Multi Data Frames to Improve the Performance of MAC Layer Based on IEEE 802.11N. *International Symposium on Research in Innovation and Sustainability*, Melaka, 15-16 October 2014, 1861-1864.

[50] Palacios, R., Franch, F., Gallego, F.V., Zarate, J.A. and, Granelli, F. (2015) Experimental Evaluation of Reverse Direction Transmissions in WLAN Using the WARP Platform. *IEEE Int. Conf. on Communication*, London, 8-12 June 2015, 6139-6145.

[51] Lemmon, J. (2002) Wireless Link Statistical Bit Error Rate Model. Technical Report 02-934, U.S. Dept. of Commerce, Boulder.

[52] Zorzi, M., Rao, R. and Milstein, L.B. (1995) On the Accuracy of a First Order Markov Model for Data Transmission on Fading Channels. 4*th IEEE International Conference on Universal Personal Communications*, Tokyo, 6-10 November 1995, 211-215. http://dx.doi.org/10.4108/icst.simutools.2013.251700

[53] Gomez, D., Aguero, R., Garcia-Arranz, M. and Munoz, L. (2013) Replication of the Bursty Behavior of Indoor WLAN Channels. *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, Cannes, 6-8 March 2013, 219-226.

[54] (2011) Wi-Fi CERTIFIED n Interoperability Test Plan Version v2.0.38. WiFi Alliance.

Appendix

In this Appendix we describe a Markov chain model for the *No-RD* scheme and for an error-free channel. The Markov chain is based on two assumptions: First, we assume that the case of 3 or more consecutive collisions on the channel is very rare. Notice that for the VO AC the probability for 2 consecutive collisions is $\left(\frac{1}{4}\right)^2\left(\frac{1}{8}\right)^2 \sim 10^{-3}$.

Therefore, we assume that only two sizes of contention intervals are used, $[0,\cdots,CW_{min}-1]$ and $[0,\cdots,2\cdot CW_{min}-1]$. Second, as already mentioned, we assume the saturated scenario where the TCP transmitter always has TCP Data segments to transmit and that the TCP transmission window does not limit the offered load. In particular, we assume that for every $K_D, 1 \le K_D \le 64$, the TCP transmitter can always transmit K_D MPDUs in a single transmission. Every MPDU contains 7 MSDUs of TCP Data segments. The TCP receiver transmits all the TCP Acks it has in one A-MPDU, up to 178×64 in one transmission.

We also assume that every TCP Ack acknowledges one TCP Ack. The extension to the case of Delayed Acks is immediate.

We first present a Markov chain for the BE and BK ACs, which are symmetrical in the sense that the AIFS of the AP and the station are equal, **Table 1**. We later show what changes are needed for the VI and VO ACs that are a-symmetrical.

The Markov chain follows after the channel access state. The set of its states, together with the transitions among the states, is shown in **Figure 14**. A state, except for the *Initial State*, represents 3 variables, and is

Group (A)

Initial State $\xrightarrow{1/C}$ $(0,C_{AP},*)$ $\quad 0 \le C_{AP} \le$ C-1

Group (B)

$(0,C_{AP},*)$ $\xrightarrow{1/(C^2)}$ $(1,C'_{AP},C'_{STA})$ $\quad 0 \le C'_{AP}, C'_{STA} \le$ C-1
$0 \le C_{AP} \le$ 2C-1

Group (C), $1 \le y \le$ (M-1)

(y,C_{AP},C_{STA}) $\quad \xrightarrow[1/C]{C_{AP}<C_{STA}}$ $(y+1,C'_{AP},C_{STA}-C_{AP})$ $\quad 0 \le C'_{AP} \le$ C-1
$0 \le C_{AP},C_{STA} \le$ 2C-1

$\xrightarrow[1]{C_{AP}>C_{STA}}$ $(0,C_{AP}-C_{STA},*)$

$\xrightarrow[1/(4C^2)]{C_{AP}=C_{STA}}$ (y,C'_{AP},C'_{STA}) $\quad 0 \le C'_{AP}, C'_{STA} \le$ 2C-1

Group (D)

(M,C_{AP},C_{STA}) $\quad \xrightarrow[1/C]{C_{AP}<C_{STA}}$ $(M,C'_{AP},C_{STA}-C_{AP})$ $\quad 0 \le C'_{AP} \le$ C-1
$0 \le C_{AP},C_{STA} \le$ 2C-1

$\xrightarrow[1]{C_{AP}>C_{STA}}$ $(0,C_{AP}-C_{STA},*)$

$\xrightarrow[1/(4C^2)]{C_{AP}=C_{STA}}$ (M,C'_{AP},C'_{STA}) $\quad 0 \le C'_{AP}, C'_{STA} \le$ 2C-1

Figure 14. Groups (A)-(D) of the Markov chain for the *No-RD* scheme and BE, BK ACs. *C* stands for CW_{min}. "*" means there is no BackOff number.

denoted (X, C_{AP}, C_{STA}). X denotes the number of $K_D \cdot 7$ TCP Acks that the TCP receiver accumulated to transmit. C_{AP} and C_{STA} are the values of the BackOff numbers of the AP and the station respectively, to be used in their next transmission attempt. These numbers are multiplied by *SlotTime* to get the BackOff intervals. When the station does not have TCP Acks to transmit, it does not have a BackOff number, denoted by "*" in **Figure 14**.

The Markov chain is based on 4 groups of states, denoted Groups (A)-(D). In the first group, Group (A), there is only one state, the *Initial State*. In this case the AP randomly chooses its first BackOff number, and move to the appropriate state in Group (B). Every transition probability is $\frac{1}{C}$ where C stands for CW_{min} and notice that the BackOff number is chosen at this stage from the interval $[0, \cdots, C-1]$.

In Group (B) there are states of the form $(0, C_{AP}, *), 0 \le C_{AP} \le 2C-1$. A state in this group denotes that the station does not have TCP Acks to transmit. The number of states in this group is $2C$. The transitions from every state in this group are according to the randomly chosen C_{AP} and C_{STA}, which are chosen from the interval $[0, \cdots, C-1]$.

In group (C) the station has $y \cdot K_D \cdot 7$ TCP Acks to transmit $1 \le y \le M$. We explain what is M later. Notice that there are three types of transitions from a state in this group-a transition when the AP transmits, when the station transmits and when there is a collision. The transitions and their corresponding probabilities are straight forward. Notice that after the station transmits it is left without TCP Acks, and the transition is to a state in Group (B). We later explain why we assume that the station is left without TCP Acks.

Notice that in principal the size of the Markov chain is unlimited. We therefore look for a finite size that will give analytical results within say 1% of those of the simulation. This seems to be a reasonable error range. We

Group (A)

Initial State $\xrightarrow[1/C]{}$ $(0,C_{AP},*)$ $\quad 0 \le C_{AP} \le$ C-1

Group (B)

$(0,C_{AP},*)$ $\xrightarrow[1/(C^2)]{}$ $(1,C'_{AP},C'_{STA})$ $\quad 0 \le C'_{AP}, C'_{STA} \le$ C-1
$0 \le C_{AP} \le 2C$-1

Group (C), $1 \le y \le$ (M-1)

(y,C_{AP},C_{STA}) $\xrightarrow[1/C]{C_{AP}<C_{STA}+1}$ $(y+1,C'_{AP},C_{STA}-C_{AP})$ $\quad 0 \le C'_{AP} \le$ C-1
$0 \le C_{AP},C_{STA} \le 2C$-1

$\xrightarrow[1]{C_{AP}>C_{STA}+1}$ $(0,C_{AP}-C_{STA},*)$

$\xrightarrow[1/(4C^2)]{C_{AP}=C_{STA}+1}$ (y,C'_{AP},C'_{STA}) $\quad 0 \le C'_{AP}, C'_{STA} \le 2C$-1

Group (D)

(M,C_{AP},C_{STA}) $\xrightarrow[1/C]{C_{AP}<C_{STA}+1}$ $(M,C'_{AP},C_{STA}-C_{AP})$ $\quad 0 \le C'_{AP} \le$ C-1
$0 \le C_{AP},C_{STA} \le 2C$-1

$\xrightarrow[1]{C_{AP}>C_{STA}+1}$ $(0,C_{AP}-C_{STA},*)$

$\xrightarrow[1/(4C^2)]{C_{AP}=C_{STA}+1}$ (M,C'_{AP},C'_{STA}) $\quad 0 \le C'_{AP}, C'_{STA} \le 2C$-1

Figure 15. Groups (A)-(D) of the Markov chain for the *No-RD* scheme and VI, VO ACs. C stands for CW_{min}. "*" means there is no BackOff number.

therefore assume that the station cannot accumulate more than $M \cdot K_D \cdot 7$ TCP Acks, and $M = 20$ gives the desired error range. Therefore, in every state in Group (C) the station can transmit all the TCP Acks it has in one transmission. (The station can accumulate up to $25 \times 64 \times 7$ TCP Acks and transmit them in one A-MPDU.)

Group (D) of states is similar to Group (B), except that the station already has $M \cdot K_D \cdot 7$ TCP Acks to transmit, and every another A-MPDU that the AP transmits is dropped by the station.

We attach a *Time* metric to every state. The *Time* metric denotes the time elapsed on the channel in this state. The *Time* metric of the *Initial State* is 0. A state in which the AP transmits has a *Time* metric equals to $AIFS + BO + Preamble + T(DATA) + SIFS + BAck$. $T(DATA)$ is the transmission time of an A-MPDU frame containing K_D MPDUs of 7 TCP Data segments each. A state in which the station transmits has a *Time* metric equals to $AIFS + BO + Preamble + T(ACK) + SIFS + BAck$. $T(ACK)$ is the transmission time of an A-MPDU frame containing all the TCP Acks that the station has. For a state in which there is a collision the *Time* metric equals to $AIFS + BO + Preamble + T(COL) + SIFS + Ack$. $T(COL)$ is the maximum between $T(DATA)$ and $T(ACK)$ of the frames involved in the collision. We denote by Ts the Time metric of state *S*.

We also attach a *Goodput* metric to every state. Recall that we consider a transmission of a TCP Data segment to be a successful one only when a TCP Ack segment is received for this segment. Thus, the *Goodput* metrics of the *Initial State*, every state in which the AP transmits and every state that denotes a collision are all 0. For any other state the *Goodput* metric is the amount of bits of $X \cdot K_D \cdot 7$ TCP Data segments, where *X* is the number of $K_D \cdot 7$ TCP Acks that are transmitted in the state, divided by the *Time* metric of the state. We denote by Gs the Goodput metric of state *S*.

The Goodput *G* of the system is $G = \dfrac{\sum_{s \in states} \pi s \cdot Ts \cdot Gs}{\sum_{s \in states} \pi s \cdot Ts}$ where $\pi s, Ts$ and Gs are the stationary probability, *Time* metric and *Goodput* metric respectively of every state *S* in the Markov chain.

Concerning the VI and VO ACs, the AIFS of the AP is shorter than that of the station by one SlotTime, **Table 1**. Therefore, a collision occurs when $C_{AP} = C_{STA} + 1$, the AP transmits when $C_{AP} < C_{STA} + 1$ and the station transmits when $C_{AP} > C_{STA} + 1$. The modified Markov chain for these ACs is shown in **Figure 15**.