IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 2, 2017, accepted January 20, 2017, date of publication February 7, 2017, date of current version March 15, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2665554

CSMA/CQ: A Novel SDN-Based Design to Enable Concurrent Execution of Channel Contention and Data Transmission in IEEE 802.11 Networks

QINGLIN ZHAO[1], FANGXIN XU[1], JIE YANG[1], AND YUJUN ZHANG[2]

[1]Faculty of Information Technology, Macau University of Science and Technology, Macau 853, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Q. Zhao (zqlict@hotmail.com)

This work was supported in part by the Macao FDCT-MOST under Grant 001/2015/AMJ, in part by the Macao FDCT under Grant 104/2014/A3 and Grant 005/2016/A1, in part by the China MOST-FDCT under Grant 2016YFE0121500, in part by the National Natural Science Foundation of China under Grant 61672500, and in part by the Instrument Development Project of CAS in China under Grant YZ201426.

ABSTRACT Conventional 802.11 carrier sense multiple access/collision avoidance (CSMA/CA) networks perform channel contention and data transmission serially over a whole channel. This leads to low throughput efficiency. In this paper, we propose a novel design called CSMA/CQ (contention queuing) to address this problem. This design is developed from the decoupling idea of SDN (Software-defined networking). In CSMA/CQ, each node concurrently executes channel contention and data transmission over two separate subchannels, where a CQ is introduced to coordinate the concurrency. This design enables CSMA/CQ to inherit the merit of the conventional distributed random channel access and carry out a centralized control on data transmission, while achieving far higher throughput than conventional 802.11 networks. We then develop a theoretical model to optimize bandwidth allocation for the channel contention and the data transmission, and prove the existence and uniqueness of the optimal solution. Extensive simulations verify the efficiency of CSMA/CQ and the accuracy of our theoretical model. *To the best of our knowledge, CSMA/CQ is the first protocol to make channel contention and data transmission be executed independently and concurrently for wireless LANs (WLANs).* This paper provides novel ideas in designing software-defined wireless networks.

INDEX TERMS OFDM, software-defined wireless networks, function separation, 802.11.

I. INTRODUCTION

A. MOTIVATIONS

With the rapid development of modern communication technologies such as orthogonal frequency division multiplexing (OFDM) [1], the physical-layer (PHY) data rate of IEEE 802.11 networks has increased significantly. However, the throughput efficiency (i.e., the ratio between the throughput and the PHY data rate) may degrade rapidly. For example, it has been pointed out in [2] that if the packet size is 1500 bytes, the efficiency quickly decreases from 60% at 54 Mb/s (802.11a/g) [3], [4] to less than 10% at 1 Gb/s (802.11ac/ad) [5]. On the other hand, with a growing deployment of various multimedia applications with diverse quality of service (QoS) requirements, IEEE 802.11 networks, which were originally designed to provide best-effort services, are very hard to meet these QoS requirements.

One fundamental reason for this throughput inefficiency is that the contention process and the data transmission process are tightly coupled. That is, all nodes must first contend for the whole channel and then the winner performs an immediate transmission over the channel. This tight coupling relationship dictates a serial execution of the two processes. As a consequence, the whole system is in an either contention or transmission state. In the contention state, all nodes perform backoff and the whole channel remains idle, naturally leading to a low channel utilization. It has been shown in [6] that because of backoff, more than 30% of throughput is reduced. What is more, the lack of QoS support is also deeply related to the tight coupling of the channel contention and the data transmission. Only in the contention process is a few parameters are allowed to be statically configured for providing service differentiation. This very limited control is the root of failing to meet these diverse QoS requirements. Therefore it is desirable to decouple the aforementioned two processes and enforce more controls in order to minimize the contention overheads and provide guaranteed QoS services.

2169-3536 © 2017 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Current IEEE 802.11 networks face the dilemmas very similar to that traditional IP networks do. In traditional IP networks, the control plane (that decides how to handle network traffic) and the data plane (that forwards traffic according to the decisions made by the control plane) are bundled inside the networking devices (such as routers and switches), reducing the flexibility of network configuration and reconfiguration to respond to faults and load changes, and hindering innovation and evolution of the networking infrastructure [7]. Software-defined networking (SDN) [8], [9] is an emerging paradigm that gives hope to change this state of affairs. Its key idea is to separate the control plane from the underlying routers and switches (that integrates with the data plane), where this separation can be realized via a well-defined programming interface between the routers/switches and the SDN controller. This separation breaks the network control problem into tractable pieces, greatly simplifying network management and facilitating network evolution.

This SDN paradigm greatly inspires us to apply the function separation idea to redesign IEEE 802.11 PHY and MAC protocols, for resolving the existing dilemmas of 802.11 networks.

B. OUR CONTRIBUTIONS

In this paper, we propose a novel design called CSMA/CQ for an infrastructure-based WLAN consisting of one access point (AP) and multiple nodes. In our design, the decoupling idea of SDN inspires us to separate the two processes of 802.11: channel contention and data transmission, while its centralized control idea motivates us to introduce a contention queue to coordinate the two processes. CSMA/CQ consists of the following PHY and MAC protocols.

- In physical layer, we adopt the OFDM technology and partition the whole channel into a contention subchannel and a transmission subchannel. Borrowing the 4G LTE method, our design enables each node to respectively contend for channel and transmit data over the two subchannels independently.
- In MAC layer, we define three processes: a contention process over the contention subchannel, a transmission process over the transmission subchannel, and a queuing process that coordinates the concurrent execution of the first two processes. In the contention process, like 802.11, all nodes execute the channel contention in a distributed manner. However, unlike 802.11 where the winner will always transmit immediately, the winner's ID is added into the queue of each node for immediate and subsequent transmission. In the transmission process, the data transmission is executed following the order in the queue.

It is true that a sender should first contend for channel and then transmit packets. However, 802.11 or all related protocols requires a node to transmit its data immediately after it wins the channel, namely, the winner in the last contention process is sure to be the transmitter of the next transmission process. Our CSMA/CQ removes the tight time-dependence between the contention and the transmission. To the best of our knowledge, *CSMA/CQ is the first protocol* to enables the contention and transmission processes to be executed independently and concurrently for WLAN. By *independence of the contention and transmission processes*, we mean

- Over the contention channel, each node only needs to concentrate on the contention of transmission opportunity, independent of the status of the data channel. For each successful contention, the winner's ID is enqueued into the CQ buffer. When to schedule the winner completely depends on the status of the data channel.
- Over the data channel, the node in the CQ buffer transmits its packets one by one following its order in the buffer, completely independent of the status of the contention channel.

We point out that without the CQ buffer (and thereby no room to store the contention results), it is impossible that the contention and transmission processes are executed independently. This independence also makes our channel allocation optimization tractable. Essentially, the queue provides an apparatus, by which a centralized entity can enforce diverse and flexible controls over the whole network. For example, the AP in CSMA/CQ can adjust the queue order utilizing all kind of exiting methods (such as priority queue) of wired networks, and then broadcast it, thereby more easily providing prioritized transmission than the conventional AP (that only depends on a static parameter configuration), as well as QoS services.

Next, we propose a novel theoretical framework to optimize the bandwidth allocation for the channel contention and the data transmission. In this framework, we model the whole network as a queuing system. By maximizing the system throughput, we explicitly express the optimal bandwidth allocation in terms of the node number and the protocol parameters, and prove the existence and uniqueness of the optimal solution.

Finally, we verify the efficiency of CSMA/CQ and the accuracy of our theoretical model with extensive simulations. For example, adopting the same time-domain contention mechanism like 802.11, CSMA/CQ has a potential of improving the throughput efficiency by 33% over 802.11.

C. RELATED WORKS

While SDN is being adopted as a novel paradigm for wired networking, in the recent two years, SDN architecture starts being introduced into wireless networking [10]–[17]. For example, in [12], an OpenFlow-based framework is applied to enterprise WLANs for realizing downlink packet scheduling among multiple APs, where OpenFlow [8] is the first standard communications interface implementing the SDN architecture. The framework is very similar to that in wired networks; the difference is that the controller not only controls traffic forwarding in the data plane (as in wired network), but also mitigates signal interference among APs. In [13], a cross-layer architecture called OpenRF is proposed for WLANs, where the controller mainly manages multiple-input multiple-out (MIMO) interference across APs. Like the OpenFlow interface, the OpenRF interface operates over a table of (FlowID, Actions) tuples. However, different from an action of OpenFlow that identifies which port to transmit the flow on, an action of OpenRF specifies the relative power used to transmit the flow on each of the AP's antennas. All these works apply the SDN architecture (that concerns the separation of the data and control planes) to coordinate the transmission among multiple WLANs. Other studies on SDN in WLANs include the access technologies and handover schemes [14], [15], the dense deployment and centralized management schemes [16], and the load balance between APs [17]. However, none of them considers the serial execution problem of contention and data transmission in WLANs. Borrowing the function separation idea of the SDN architecture, this paper proposes CSMA/CQ to decouple channel contention and data Transmission for solving the existing problems in a WLAN.

CSMA/CQ aims at supporting concurrent execution of channel contention and data transmission, utilizing an OFDMA mechanism (where a part of OFDM subcarriers are used for the channel contention and the other part are used for the data transmission). In contrast, most existing OFDM-subcarriers-based schemes [2], [18]–[20] still keep the channel contention and the data transmission being executed serially, while both the channel contention and the data transmission occupy the whole channel. They differ mainly at the channel contention process. Sen *et al.* [18], [20] proposed a frequency-domain contention scheme called T2F and Back2F. In this scheme, each node randomly chooses a subcarrier to signal and the node choosing the minimum subcarrier number is the winner; upon the completion of contention, the winner immediately transmits data over all subcarriers. Different from T2F and Back2F, Fang *et al.* [2] proposed a fine-grained channel access scheme called FICA. In this scheme, each node contends for a certain number of subcarriers according to its traffic demand, and then transmits data over these subcarriers upon success. In [19], the author proposed a bitwise arbitration mechanism. In this scheme, each node adopts a time- and frequency-domain mechanism to resolve collision; upon success, the winner transmits data over the whole subcarriers.

CSMA/CQ partitions the whole channel into two subchannels: one for contention and another for data transmission. Therefore it is also a multichannel MAC protocol. Most existing multichannel protocols were designed for multi-hop wireless networks [21]–[24]. Ref. [25], [26] well classified and compared these multichannel protocols. Among these multichannel protocols such as [21] and [22], the channels are often classified into a common control channel (over which each single-hop network adopts the RTS/CTS mechanism to contend for the transmission opportunity and transmit its chosen data channel number, thereby avoiding the hidden/exposed terminal problems during the data transmission) and a number of data channels (where each single-hop network can choose one available data channel for data transmission after it wins the transmission opportunity). However, for each single-hop network, the contention and data transmission processes are still executed serially, namely, the sender must first win the transmission opportunity over the common control channel and then transmit data over a data channel. As a result, the data channel remains idle when the control channel is busy, leading to low channel utilization. In contrast, CSMA/CQ, which is designed for a one-hop WLAN, aims at making the control and data channel work concurrently, therefore improving the channel utilization.

Among these multichannel protocols, the study in [23] is the most relevant to ours. Kyasanur *et al.* [23] proposed a control channel-based MAC protocol (CCM) that is applicable for the infrastructure-based WLAN (we considered in this paper). CCM adopts a low-frequency band (i.e., below 900MHz) with narrow bandwidth (1 or 2 MHz) for contention, and utilizes a high-frequency band (i.e., 2.4 GHz) with wide bandwidth (i.e., 22 MHz) like 802.11 for data transmission. Because the contention channel is separated far away from the data channel, they have little interference. As a result, while the i-th packet is transmitted over the data channel, nodes can contend for the transmission opportunity for the i+1 packet; note that it is not allowed to contend for the transmission of the i+2 packet because no buffer is employed to store the contention result, implying a strong dependence between the contention and the data transmission. To avoid that the contention channel is the bottleneck of system performance due to its narrow bandwidth, CCM introduces a data aggregation technology (i.e., a node needs to aggregate a certain number of packets before it begins contention) and an advance reservation technology (which allows a node to transmit multiple packets for one successful contention). However, the data aggregation technology might lead to a high delay while the advance reservation technology must be carefully designed, both technologies complicating the protocol. In contrast, CSMA/CQ has distinct differences from the CCM as follows.

- First, we employ one full radio frequency (RF) module for contention and data transmission (instead of two full RF modules as in CCM). This is explained in Section III-A.2.
- Second, we allow the contention and data channel to work in a contiguous frequency band (instead of two non-contiguous frequency band), and we use the OFDMA mechanism (which is used in 4G LTE) to eliminate the channel interference. It is worth pointing out that currently, the sub-1-GHz license-exempt frequency spectrum has been used in IEEE 802.11ah [37] for Internet of Things, etc. It is hard to find a separate license-exempt frequency spectrum for contention only.
- Third, we allow flexible bandwidth allocation (instead of fixed narrow bandwidth) for contention; we optimize the bandwidth allocation and prove its existence and uniqueness in Section IV, and verify its accuracy in Section V. Note that the throughput efficiency of our

design can be 9% higher than that of CCM, as shown in Figure 7.

- Fourth, the most importance feature is that we apply the SDN idea to enable an independent and concurrent execution of the contention and transmission processes (stated in Section I-B). In contrast, CCM retains dependence between contention and data transmission in some degree (namely, while the i-th packet is transmitted over the data channel, nodes are allowed to contend for the transmission opportunity of the i+1 packet only). Moreover, in CSMA/CQ, the AP can modify the CQ buffer (say, reschedule the transmission order of the nodes) and achieve a software-defined transmission.

In addition, our previous work [36] mainly focused on the MAC design (i.e., Section III-B) of CSMA/CQ. This paper makes a considerable extension in the following aspects: (i) clarify the motivation and contributions (from the viewpoint of SDN), as well as carry out a thorough survey on related works in Section I, (ii) detail the PHY design in Section III-A and handle the hidden/exposed terminal problems in Section III-D, (iii) develop a novel theoretical framework to optimize subcarrier allocation in Section IV, and (iv) run more extensive simulations to verify the effectiveness of our design and the accuracy of our theoretical model.

The rest of the paper is organized as follows: Section II outlines CSMA/CQ. Section III details CSMA/CQ. Section IV optimizes the bandwidth allocation. Section V evaluates via simulation the performance of CSMA/CQ. Finally, Section VI concludes the paper.

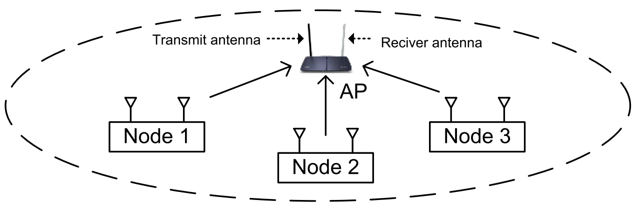


FIGURE 1. An infrastructure-based WLAN.

II. CSMA/CQ OVERVIEW

CSMA/CQ is designed for an infrastructure-based WLAN, as shown in Figure 1. It supports the concurrent execution of contention and data transmission over two separate channels (in contrast, DCF supports the serial execution of contention and data transmission). To better explain our design idea, we first outline 802.11 DCF and then CSMA/CQ.

A. 802.11 DCF

We now describe the serial process of 802.11 DCF with RTS/CTS mode, and specify its drawbacks.

With the help of Figure 2(a), we outline the serial process of DCF. In DCF, after sensing channel idle for DIFS, each node starts to contend for the channel using the 802.11 DCF binary exponential backoff (BEB) algorithm, and then sends a RTS frame to reserve the channel. Upon receiving a CTS feedback, the RTS-sender transmits a DATA frame to AP. Finally, AP sends an ACK back to the node. For example, node 2 executes the whole serial process following the pattern: DIFS/backoff/RTS/SIFS/CTS/SIFS/DATA/SIFS/AC K.

However, this serial process has the following drawbacks. First, the backoff process, generally taking a quite long time, often forces the whole channel to be idle. This seriously lowers the channel utilization. Second, DCF allocates the entire channel to a node as a single resource. This allocation strategy becomes too coarse-grained when the PHY data rate increases [1].

B. CSMA/CQ

To overcome the two drawbacks, we decompose the serial process into three concurrent processes (namely, contention process, transmission process, and queuing process) with different resources. For contention process and transmission process, we split the whole channel into two subchannels: one for contention and the other for data transmission. For queuing process, we design a contention queuing buffer (CQ buffer) to let it coordinate the above two processes. With the help of Figure 2 (b), we now describe these three processes and specify how they work cooperatively.

1) CONTENTION PROCESS

In this process, nodes use the 802.11 time-domain contention mechanism with RTS/CTS to contend for channel. When a node finishes a RTS/CTS hand shake, all nodes store the contention result into their CQ buffers. As shown in Figure 2 (b), nodes 2, 1, 3 sequentially win the channel, and all nodes add these node IDs into theirs buffers one by one. To improve the contention efficiency, the improvement of the time-domain contention mechanism (such as the idle sense scheme [27]) or the frequency-domain contention mechanism (such as such as T2F [18] and Back2F [20]) can be directly adopted in the contention process. However, we here focus on the conventional 802.11 channel contention mechanism for presenting our CSMA/CQ design.

2) TRANSMISSION PROCESS

In this process, as long as the CQ buffer is not empty, nodes first listen to the channel, and then sequentially transmit their frames, according to the order of node IDs stored in the CQ buffer. For each transmission completes, all nodes delete the corresponding node ID from their buffers. As illustrated in Figure 2 (b), nodes 2, 1, 3 sequentially complete their transmissions, and all nodes remove these node IDs from theirs buffers one by one.

3) QUEUING PROCESS

This queuing process connects the contention process and the transmission process. Each node dynamically manipulates its own CQ buffer, according to the contention and transmission processes. For example, as shown in Figure 2 (b), nodes 2 and 1 first win the channel in order, and then

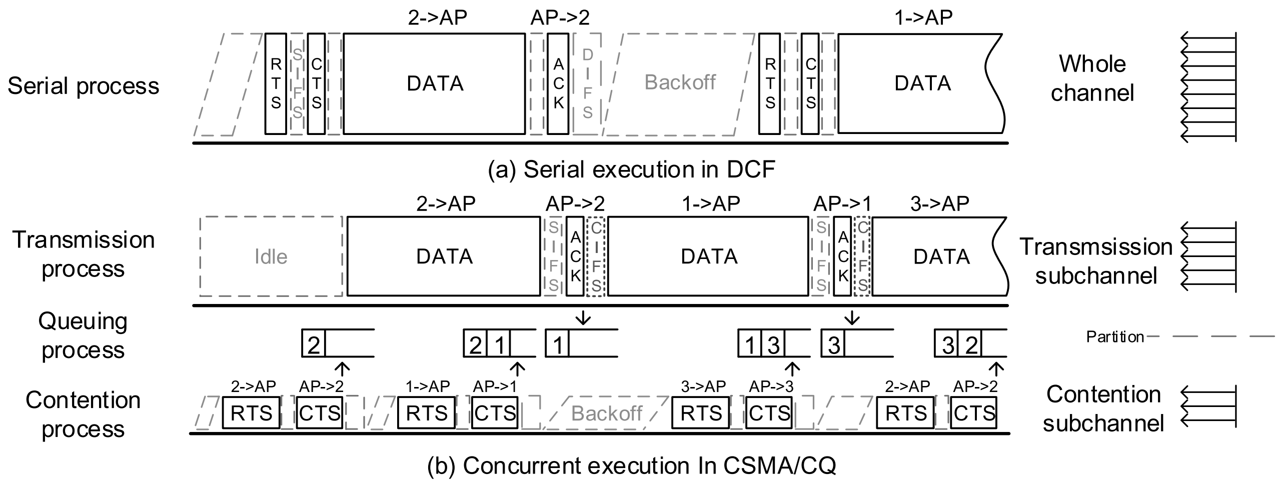


FIGURE 2. Channel contention and data transmission: (a) serial execution in the whole channel for DCF, (b) concurrent execution in separate subchannels for CSMA/CQ.

node 2 completes its transmission. As a result, all nodes first add the IDs of nodes 2 and 1 into their respective CQ buffers sequentially, and then remove the ID of node 2 from their buffer. In the same manner, the queuing process repeats.

In short, the three processes are concurrently executed, without wasting the whole channel for idle during contention. This greatly improves the channel utilization.

III. CSMA/CQ DESIGN DETAILS

In this section, we detail the CSMA/CQ design in an infrastructure-based WLAN (as shown in Figure 1), where there are one AP and a number of nodes.

A. PHY DESIGN

In CSMA/CQ, each node and the AP have a full RF module, which consists of one transmit (TX) RF module equipped with a TX antenna, and one receive (RX) RF module equipped with a RX antenna, as shown Figure 1. Note that the TX and RX antennas should be separated by about 2 feet to reduce self-interference [19]. For each node, the RF module is shared by the contention and transmission subchannels.

We assume that the physical layer adopts the OFDM modulation scheme [1], which has been widely used in modern wireless communications such as IEEE 802.11a/g/n [3], [4], WiMAX [28], and 4G LTE [29]. OFDM divides a given channel into a number of subcarriers including data subcarriers for data transmission, pilot subcarriers for estimation and synchronization purposes, and null subcarriers for guard bands. For example, in 802.11a/g, OFDM divides a 20MHz channel into 48 data subcarriers, 4 pilot subcarriers, and 12 null subcarriers. In OFDM, to send a packet, the OFDM transmitter first modulates the packet into each data subcarrier, then performs an inverse discrete Fourier transform (IDFT) for converting these frequency-domain data blocks into a series of time-domain data blocks, and finally transmits them. At the receiver end, the OFDM receiver can recover the packet by performing a DFT operation on these time-domain data blocks. We call each time-domain data block an OFDM symbol. Note that each OFDM symbol is a superposition of all subcarriers. When OFDM is used in multi-access environments like WiMax and LTE, a multiaccess technology called OFDM access (OFDMA) has been proposed to ensure that OFDM symbols from multiple nodes align. This requires that all mobile stations should maintain tight timing synchronization with a central controller.

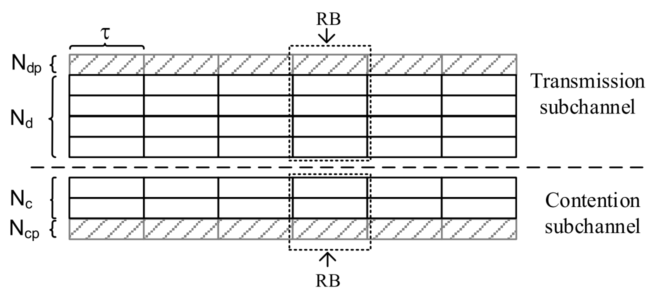


FIGURE 3. Resource blocks (RBs) in CSMA/CQ.

Assume that the whole channel has N data subcarriers, N_p pilot subcarriers, and some null subcharriers. We can partition the whole channel into two subchannels (as illustrated in Figure **3**): the contention subchannel that consists of N_c data subcarriers and N_{cp} pilot subcarriers, and the transmission subchannel that consists of N_d data subcarriers and N_{dp} pilot subcarriers, where $N = N_c + N_d$, and $N_p = N_{cp} + N_{dp}$. How to spread the pilot subcarriers is implementation-dependent. For example, like 802.11a, the pilot subcarriers can be uniformly spread throughout the transmission bandwidth.

1) TIME-SYNCHRONIZATION AND INDEPENDENT EXECUTION OF TWO SUBCHANNELS

In CSMA/CQ, time is divided into a series of slots. The channel contention or the data transmission starts at the beginning of a slot. We have two design requirements below:

- Each antenna can decode the received data correctly and independently in every symbol time.
- Each antenna can transmit or receive data at the beginning of any slot, whenever it wants, without worrying about whether an OFDM symbol is aligned or not.

The first requirement implies that a strict time-synchronization of each OFDM symbol should be maintained. The second requirement implies that for each node, the random channel access and the data transmission are independent. This ensures that the contention and transmission processes can be executed concurrently.

To further specify our PHY design, let τ denote an OFDM symbol time. Like 4G LTE, define a resource block (RB) for each subchannel as follows (as shown in Figure **3**).

- A RB of the contention channel is a minimum time- and frequency- resource that occupies ($N_c + N_{cp}$) subcarriers and one OFDM symbol time.
- A RB of the transmission channel is a minimum time- and frequency-resource that occupies ($N_d + N_{dp}$) subcarriers and one OFDM symbol time.

Now, to fulfill the aforementioned two design requirements, we make two rules:

- In each OFDM symbol time, the two RBs of the two subchannels are time-aligned.
- Each time block (i.e., slot, DIFS, SIFS, CIFS, RTS, CTS, DATA, and ACK) is an integer multiple of the OFDM symbol time.

The first rule fulfills requirement 1. The reason is as follows. When the node only transmits data over the transmission subchannel (rather than transmits signals over the contention subchannel), it can set the RB of the contention subchannel to zero. When the node only transmits signals over the contention subchannel (rather than transmits data over the transmission subchannel), it can set the set the RB of the transmission subchannel to zero. According to this rule, in every symbol time, all OFDM symbols from any antenna are always aligned and therefore each antenna can always decode an OFDM symbol correctly (we assume idea channel conditions). The second rule fulfills requirement 2. The reason is as follows. According to this rule, a slot is an integer multiple of the OFDM symbol time. Then, as long as the channel access or the data transmission starts at the beginning of any slot, we can always align all OFDM symbols. Therefore, each antenna can work independently.

Finally, we point out that the first rule is easily achieved, as long as each node maintains tight timing synchronization with the AP as required in WiMax and LTE. So is the second rule, because it only involves some parameter settings. In addition, we can also adopt the state-of-the-art physical techniques (such as FBMC [30] and GFDM [31]) to improve the PHY design and increase the spectral efficiency.

2) TRANSMISSION AND RECEPTION OF TWO SUBCHANNNELS OVER A RF MODULE

Here, based on the concept of RB defined in Section III-A.1, we specify the transmission and reception operations of the two subchannels over a RF module. With the help of Figure 2(a), we explain the work procedure.

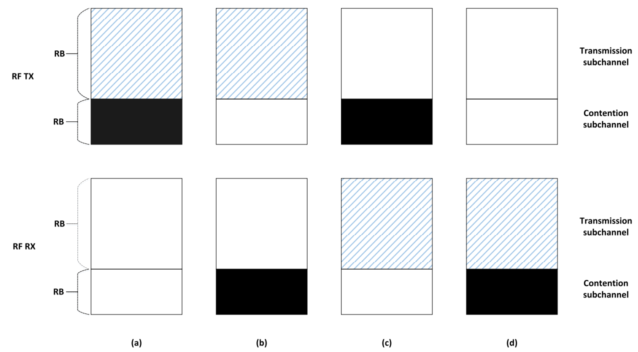


FIGURE 4. Contention and transmission subchannels share the same RF TX and RX: (a) both subchannels are in the TX sate; (b) the transmission subchannel is in the TX state, while the contention subchannel is in the RX state; (c) the transmitssion subchannel is in the RX state, while the contention subchannel is in the TX state; (d) both subchannels are in the RX sate.

In CSMA/CQ, a node only has one RF transmit (TX) antenna and one RF receive (RX) antenna. For both subchannels, the node employs the same RF TX for transmission, and employs the same RF RX for reception. Note that each subchannel is in either TX or RX states. We have four cases in total.

(a). Both subchannels are in the TX sate (Figure 4 (a)). In this case, the node sets the corresponding RBs, and then transmits them over the RF TX, while ignoring anything received from the RF RX.

(b). The transmission subchannel is in the TX state, while the contention subchannel is in the RX state (Figure 4 (b)). In this case, the node constructs an OFDM symbol (in which the RB of transmission subchannel is set to the useful information and the RB of contention subchannel is set to zero) and transmit it over the RF TX. At the same time, upon receiving an OFDM symbols over the RF RX, the node ignores the RB of the transmission subchannel and only handles the RB of the contention subchannel. Note that the RB of contention subchannel over the RF TX is set to zero. The transmission over the RF TX has no any impact on the reception over the RF RX.

(c). The transmitssion subchannel is in the RX state, while the contention subchannel is in the TX state (Figure 4 (c)). This case is similar to case (b).

(d). Both subchannels are in the RX sate (Figure 4 (d)). In this case, the node transmits nothing over the RF TX. At the same time, upon receiving an OFDM symbol over the RF RX, the node handles the corresponding RB respectively.

In short, employing one RF module only, CSMA/CQ can enable the transmission and the reception of the two subchannels to be executed in parallel.

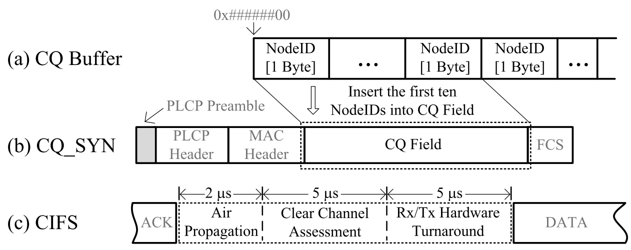


FIGURE 5. Three new structures in CSMA/CQ: (a) CQ Buffer (b) CQ_SYN (c) CIFS.

B. MAC DESIGN

1) THREE NEW STRUCTURES

CSMA/CQ introduces three novel structures (as illustrated in Figure 5): the CQ (contention queuing) buffer, the CQ_SYN frame, and the CIFS (contention queuing inter frame spacing) interval.

CQ buffer: the CQ buffer consists of at most 20 bytes. It is used to store the ID of the winner in contention process. When one contention process finishes, the winner's ID will be enqueued into the CQ buffer. When one transmission process finishes, the ID of the node transmitting the frame will be dequeued from the CQ buffer.

CQ_SYN: CQ_SYN is a frame consists of PLCP Preamble, PLCP header, MAC header, CQ field and FCS field. The CQ_SYN is used to synchronize the transmitting order, where the length of CQ filed is 10 bytes (ten is chosen by our experience), where each byte is used to store one node ID. In this paper, we assume that each node ID is mapped from its MAC address. In the transmission process, once detecting that the current sender's ID is not the head of the CQ buffer, AP constructs a CQ_SYN frame and inserts the first ten node IDs in its own CQ into the CQ field, and then broadcast it. Upon receiving the CQ_SYN, all nodes will update their CQ buffers.

CIFS: a CIFS time is set to 12μs, it consists of an Air Propagation, a Clear Channel Assessment and an Rx/Tx Hardware Turnaround time, where Air Propagation is set to 2μs and the other two are set to 5μs. The purpose of CIFS is to provide a protection interval. It marks the end of the current transmission, and the beginning of a new transmission.

2) MAC PROTOCOL

Consideration of downlink transmission. In CSMA/CQ, only the AP is allowed to control and modify the CQ buffer. This greatly facilitates the downlink transmission (from the AP to nodes). To make downlink transmission simple, we borrow two concepts well defined in conventional 802.11 networks:

- CTS-to-Self. It is a special control frame. Usually, a node should send a RTS/CTS pair to reserve channel. 802.11 allows a node to send CTS-to-Self directly without sending RTS before.
- AIFS (arbitration inter-frame spacing). It is a time spacing. AIFS = SIFS + k × slot, where k is a positive integer. Therefore, AIFS can be either larger or less than DIFS.

CSMA/CQ adopts an AIFS (less than DISF). We stipulate that only the AP can send CTS-to-Self after listening channel for an idle time of AIFS, and then all nodes and the AP will enqueue the AP's ID into the CQ buffer. In other words, the downlink traffic has the highest priority and can be transmitted without experiencing any collision. To avoid that the AP always monopolizes channel, it is easy to define rules to limits the use of CTS-to-Self.

Algorithm 1 Contention Process (Node → AP)

```
Perform backoff (like BEB)
// Case 1: Node counts down to zero
Send a RTS
Delay(SIFS) /*Wait for a CTS*/
// If Node itself wins.
if Receive a CTS feedback then
  /* Node joins buffer and waits for its transmission
  (instead of transmitting immediately as in DCF) */
  CQ_Enqueue(CTS.DstID)
else /* RTS collides */
  Increase contention window
// Case 2: Node is in backoff
if Receive a CTS then //other node wins
  CQ_Enqueue(CTS.DstID) // Add it into buffer.
```

Algorithm 2 Contention Process (AP → Node)

```
if Receive a RTS then
  * Add the sender into buffer for scheduling
  its transmission */
  CQ_Enqueue(RTS.SrcID)
  Feed back a CTS (like DCF)
end
```

Consideration of uplink transmission. In this paper, our MAC design mainly focuses on uplink traffic (i.e., all nodes send DATA to AP but AP just acts as a receiver). In CSMA/CQ, all nodes follows the BEB algorithm to perform the contention process (specified in Algorithm 1 and Algorithm 2) over the contention subchannel, where the winner of each contention will be added into the CQ buffer. At the same time, all nodes also perform the TDMA-like transmission process (specified in Algorithm 3 and Algorithm 4) according to the order of node IDs stored in the CQ buffer. We now detail the four algorithms.

Algorithm 1 specifies the contention process (Node→AP) where each node transmits RTS over the contention subchannel to reserve the transmission subchannel. In this process, each node performs backoff as BEB. Once the backoff counter reaches zero, the node immediately sends a RTS to AP and expects a CTS feedback. If there comes a CTS, both the node itself (i.e., the RTS sender in case 1) and other nodes (i.e., the "backing off" nodes in case 2) enqueue CTS.DstID (i.e., the node ID mapped from the Destination field in CTS or the corresponding RTS sender's ID) into their CQ buffers. If there is a collision, the collided nodes double the contention windows, and then perform backoff and repeat the above process.

Algorithm 2 specifies the contention process (AP→Node), where AP feeds back CTS over contention subchannel. When AP receives a RTS frame, it enqueues RTS.SrcID (i.e., the RTS sender's ID) into its CQ buffer (and sends back a CTS like DCF. Note that compared with DCF, CSMA/CQ does not use the Duration field in RTS and CTS frames, so more functions can be extended to use this Duration field.

Algorithm 3 Transmission process (Node → AP)

```
while (CQ_Head ~= Null)
 // Case 1: Node is at the CQ_Head
 if (CQ_Head == NodeID) then
  if channel remains idle for CIFS then
   Send a DATA frame
   Delay(SIFS) // Wait for an ACK
   if Receive an ACK then
    CQ_Dequeue() // Delete the ID of transmitted node
   else // Update CQ buffer by AP's feedback
    if Receive a CQ_SYN then
     CQ_Update(CQ_SYN)
     /* If Node is no longer at the CQ_Head, it drops
      the frame being transmitted; otherwise, retransmit
*/
     if (CQ_Head ~= NodeID) Drop the data frame
    else // CQ_SYN timeout
     Goto line 01
 end
 // Case 2: Node is not at the CQ_Head
 if (CQ_Head ~= NodeID) then // Other node is in
transmission
  if Receive an ACK then CQ_Dequeue()
 else // Synchronize its CQ buffer with AP
  if Receive a CQ_SYN then CQ_Update(CQ_SYN)
 end
```

Algorithm 3 specifies the transmission process (Node→AP), where each node transmits over the transmission subchannel. In this process, each node transmits following the order of NodeIDs stored in the CQ buffer. While the CQ buffer is not empty, node checks whether the head of the CQ buffer (CQ_Head) equals to NodeID. If yes, the node first sends a DATA frame upon the channel is idle for CIFS and next waits for AP's response; if no, the node just waits for AP's response passively. Both node itself (i.e., the node in case 1, who matches the CQ_Head) and the other nodes (i.e., the nodes in case 2, who does not match the CQ_Head) need to handle three situations: (a) If there comes an ACK, nodes dequeue the current CQ_Head as the transmission has been finished. (b) If there comes a CQ_SYN frame (a frame contains AP's own CQ buffer), it manifests that AP has found some issues (e.g., out of synchronous or reach the max retry limit) and needs to resynchronize all the CQ buffers, nodes then update their CQ buffers according to the CQ_SYN (namely, reset the CQ buffer to a new one obtained from CQ_SYN). In addition, the current transmitter also has to drop the frame being transmitted (note that it is the high layer who is responsible for retransmitting the dropped frame). (c) If CQ_SYN timeout occurs (CQ_SYN timeout is a time longer than ACK timeout, namely, it is the case where nodes receive neither ACK nor CQ_SYN), since CQ_Head remains unchanged, the sender retransmits the DATA frame.

Algorithm 4 Transmission Process (AP → Node)

```
while Receive a DATA frame
 if No errors in DATA then
  if DATA.SrcID == CQ_Head then
   Feed back an ACK; CQ_Dequeue()
  else // Synchronize all CQ buffer forcibly
   Feed back a CQ_SYN
 else //Check RetryLimit for dequeue
  if RetryLimit < threshold then RetryLimit++
  else CQ_Dequeue()
  //Synchronize all CQ buffers for each retransmission
  Feed back a CQ_SYN
```

Algorithm 4 specifies the transmission process (AP→Node), where AP responds towards present transmission over the transmission subchannel. Upon receiving a DATA frame, AP first uses the FCS field to check whether the frame is corrupted. If there are no errors, AP then turns to compare its CQ_HEAD with DATA.SrcID (i.e., the DATA sender's ID). If they match, AP feeds back an ACK and dequeues its CQ_HEAD to finish the transmission; otherwise it broadcasts a CQ_SYN frame to synchronize all the CQ buffers forcibly. If there is any error in DATA, AP first checks RetryLimit (namely, the number of times that the corresponding node has retransmitted for). If RetryLimit reaches the threshold (the threshold is chosen by experience), AP infers that the corresponding channel may not be satisfactory, thus dequeues CQ_HEAD (namely, drops the corresponding node to release channel for other better-condition nodes) and broadcasts a CQ_SYN. If the RetryLimit is lower than threshold, AP only broadcasts a CQ_SYN to let the node retransmit.

C. CQ-BASED COORDINATION/SYNCHRONIZATION MECHANISM

CSMA/CQ inherits the listen before transmit (LBT) mechanism of conventional 802.11 networks. On this base, we introduce a CQ buffer to coordinate the whole concurrent execution of contention and data transmission, as well as synchronize the data frame transmissions in the transmission subchannel.

In CSMA/CQ, each node and the AP have a CQ buffer. The CQ buffer stores the IDs of nodes who has already wined the channel and will transmit packets. The AP controls or maintains the CQ, ensuring that each CQ buffer shares the same info. In our design, each CQ buffer is updated, if and only if the AP transmits one of the four control frames below:

- *CTS:* When a node contends via the RTS/CTS mechanism, if the AP transmits a CTS, the winner's ID will be enqueued into each CQ buffer.
- *CTS-to-Self:* When the AP tries to transmit packets to nodes, it will send CTS-to-Self. Then the AP's ID will be enqueued into each CQ buffer.
- *ACK:* When a node transmits a packet, if the AP sends back an ACK, the transmitter's ID will be dequeued from each CQ buffer.
- *CQ_SYN:* Whenever the AP finds that the source ID of the transmitting packet is not the first one of the CQ buffer or finds the received packet erroneous, it will transmit CQ_SYN to force each node to update its CQ buffer [see Case 2 in Algorithm 3 and Algorithm 4].

The first two control frames are transmitted over the contention channel, while the latter two control frames are transmitted over the data channel.

In the above four control frames, the first three are used to coordinate the concurrent execution of the two subchannels (note that our PHY design can ensure the independent executions of the two subchannels by adopting the OFDMA method), while the fourth is used to synchronize the data frame transmissions in the transmission subchannel. In normal conditions, the first three frames can already ensure that each node shares the same CQ buffer info. However, in reality, some exceptions might still lead to inconsistent CQ buffer info among nodes. In this situation, AP can transmit the CQ_SYN to synchronize all CQ buffers.

D. HIDDEN/EXPOSED TERMINAL PROBLEMS

Here, we discuss the hidden/exposed terminal problems in a WLAN and among WLANs.

No hidden/exposed terminal problems in a WLAN. In the discussed WLAN, the AP is located in the center of the network. A hidden terminal problem happens, only if a node transmits a packet to the AP but another node does not hear this transmission. For the contention channel, each node adopts the RTS/CTS mechanism to contend; it has been well known that RTS/CTS can eliminate the hidden terminal problem. Second, for the data channel, each node adopts the LBT mechanism and transmits packet in a TDMA manner (namely, each time, only one node is allowed to transmit according to the order in the CQ buffer), thereby eliminating the hidden terminal problem as well. On the other hand, in an infrastructure-based WLAN, the AP is the unique receiver and therefore no exposed terminal problems happen.

Hidden/exposed terminal problems among WLANs. It is possible that the transmission in one WLAN will interfere with other transmissions in another WLAN. The CQ idea can be used to solve the hidden/exposed terminal problems. For example, we can let each AP be connected to a central controller via wired lines (of course, it is allowable to define some rules for letting one AP act as the controller). At the beginning, the controller can construct graphs on link-interference and carrier-sensing of these WLANs, which can be used to solve the hidden/exposed terminal problems by the method [32]. Then all AP can send their CQ buffers to the controller periodically. The controller then determines the transmission order of nodes by these graphs and sends each updated CQ buffer to the corresponding AP. In this way, we can solve the hidden/exposed terminal problems among WLANs.

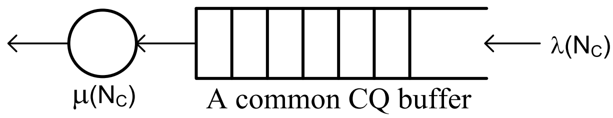


FIGURE 6. The queueing model for CSMA/CQ.

IV. OPTIMAL SUBCARRIER ALLOCATION

In CSMA/CQ, the most important design issue is how to optimize the data subcarrier numbers, $\mathrm{N_c}$ and $\mathrm{N_d}$, of the contention and transmission subchannels, which maximizes the system throughput. To this end, we first model the CSMA/CQ system as a queueing system, and then find the optimal solution. To facilitate the theoretical analysis, we assume that the CSMA/CQ system is in the saturation regime (where each node always has packets to transmit).

A. MODELING IDEAS

In CSMA/CQ, each node maintains a CQ buffer with the same information. Therefore, we can artificially assume that all nodes share a common CQ buffer. In the contention process, nodes contend for transmission opportunity over the contention subchannel; the IDs of winners will be enqueued into the common CQ buffer. At the same time, in the transmission process, nodes (whose IDs are in the common CQ buffer) send frames in order over the transmission subchannel; upon a successful transmission, the ID of the sender will be dequeued from the common CQ buffer. The subcarrier number N_c of the contention subchannel, which determines the subcarrier number $N_d = (N - N_c)$ of the transmission subchannel and determines the physical transmission rate, has an important impact on the enqueue and dequeue rates. Given N_c, let $\lambda(N_c)$ and $\mu(N_c)$ represent the enqueue and dequeue rates of the common CQ buffer, respectively. Then, we can model the CSMA/CQ system as a queueing system with an arrival rate $\lambda(\mathrm{N_c})$, a service rate $\mu(\mathrm{N_c})$, and an infinite buffer. IV-B illustrates the queueing model for CSMA/CQ.

B. ENQUEUE RATE

In the contention process of CSMA/CQ, nodes use the standard RTS/CTS mechanism in DCF to contend for transmission opportunity. The IDs of winners will be enqueued into the common CQ buffer. Then, the enqueue rate $\lambda(N_c)$ is equal to the number of winners in a unit time, which is determined by the DCF protocol and the subcarrier number N_c of the contention subchannel.

We now calculate the enqueue rate $\lambda(N_c)$ following the DCF performance analysis method [33], [34]. Below, we first calculate the saturated attempt rate, then define the generic slot, and finally express the enqueue rate.

1) SATURATED ATTEMPT RATE

Let β and α represent the saturated attempt rate and collision probability for each node, respectively. Since the contention process follows 802.11 DCF, we can calculate (β, α) as in 802.11. From [34], (β, α) is uniquely governed by the following fixed-point equations:

$$\begin{cases} \beta & = \dfrac{1+\alpha+\alpha^2+\cdots+\alpha^M}{b_0+\alpha b_1+\alpha^2 b_2+\cdots+\alpha^M b_M} \\ \alpha & = 1-(1-\beta)^{n-1} \end{cases} \tag{1}$$

where n is the number of contending nodes, M is the frame retry limit, the mean of the backoff count at stage k, $b_k = 2^k b_0$ for $0 \leq k \leq m-1$ and $b_k = 2^m b_0$ for $m \leq k \leq M$, and m determines the maximum backoff window size. In standard 802.11, $b_0 = 16$, $m = 5$ and $M = 7$.

2) GENERIC SLOT

Given N_c, let $\Omega(N_c)$ represent the mean time that elapses for one decrement of the backoff counter in DCF. Note that the backoff counter decreases by one for each idle time slot and is suspended when the channel is busy. $\Omega(N_c)$ is given by

$$\Omega(N_c) = (1-p_{tr})\,\sigma + p_s T_s(N_c) + (p_{tr}-p_s)\,T_c(N_c) \tag{2}$$

where $p_{tr} = 1-(1-\beta)^n$ is the probability of a busy slot; $p_s = n\beta(1-\beta)^{n-1}$ is the successful transmission probability among n contending nodes, and it is easy to prove $p_s < p_{tr}$ since the successful transmission is a special case of the transmission process; σ is the duration of one time slot, $T_s(N_c)$ and $T_c(N_c)$ are, respectively, the successful and unsuccessful RTS/CTS handshaking time for a given N_c. Since the RTS/CTS handshaking mechanism follows the transmission pattern: DIFS/RTS/SIFS/CTS, $T_s(N_c)$ is expressed as follows:

$$T_s(N_c) = T_{difs} + \frac{L_{rts}+L_{cts}}{R_b N_c} + T_{sifs},$$

where T_{difs} and T_{sifs} represent a DIFS time and a SIFS time respectively, L_{rts} and L_{cts} represent the lengths of the RTS and CTS frames respectively, R_b represents the physical transmission rate of each subcarrier, and N_c represents the subcarrier number of the contention subchannel. Further, we assume:

$$T_c(N_c) = T_s(N_c). \tag{3}$$

The assumption is equivalent to assuming that the CTS timeout after a collision matches the guard time observed by noncolliding nodes.

3) ENQUEUE RATE

We define the enqueue rate $\lambda(N_c)$ to be the mean number of successfully transmitted RTS frames (which is p_s, since the system has 1 successful transmission with probability p_s, and fails in transmission with $1-p_s$), in the time duration of $\Omega(N_c)$. In other words, we have

$$\lambda(N_c) = \frac{p_s}{\Omega(N_c)}. \tag{4}$$

C. DEQUEUE RATE

In the transmission process of CSMA/CQ, nodes perform the TDMA-like transmission following the order that the node IDs are stored in the common CQ buffer; upon a successful transmission, the ID of the sender will be dequeued from the common CQ buffer. In the transmission process, each frame transmission follows the pattern: CIFS/Frame/SIFS/ACK. Therefore, each frame transmission time is

$$\begin{aligned} & T_{cifs} + (T_h + T_p) + T_{sifs} + T_{ack} \\ & = T_{cifs} + T_{sifs} + T_h + T_p + T_{ack} \\ & = T_{cifs} + T_{sifs} + \frac{L_h + L_p + L_{ack}}{R_b N_d} \\ & = T_{cifs} + T_{sifs} + \frac{L_h + L_p + L_{ack}}{R_b (N - N_c)}, \end{aligned}$$

where $T_{cifs} \in (T_{sifs}, T_{difs})$ is the CIFS time; $T_h + T_p$ is the frame transmission time consisting of the frame header time T_h and the frame payload time T_p; T_{sifs} is the SIFS time; T_{ack} is the ACK time; L_h, L_p, and L_{ack} represent the lengths of the frame header, the frame payload and the ACK respectively; N_c, N_d, and N represent the subcarrier numbers of the contention subchannel and the transmission subchannel, and the total channel respectively; and $\mathrm{R_b}$ is the physical transmission rate of each subcarrier.

Then, the dequeueing rate $\mu(N_c)$ is equal to the reciprocal of one frame transmission time, namely,

$$\begin{aligned} \mu(N_c) & = \frac{1}{T_{cifs} + (T_h + T_p) + T_{sifs} + T_{ack}} \\ & = \frac{1}{T_{cifs} + T_{sifs} + \frac{L_h+L_p+L_{ack}}{R_b(N-N_c)}} \end{aligned} \tag{5}$$

D. OPTIMAL SUBCARRIER ALLOCATION

In this section, we first define system throughput and derive its formula, and next present the optimal subcarrier allocation that maximizes the system throughput.

Let $\Gamma(N_c)$ represent the system throughput (i.e., the number of successfully transmitted bits in a unit time) of CSMA/CQ. Then, if $\lambda(N_c) < \mu(N_c)$, the throughput is mainly constrained by $\lambda(N_c)$ (i.e., the total number of winners who are allowed to transmit data frame within a unit time). In other word, we have $\Gamma(N_c) = \lambda(N_c) L_p$; If $\lambda(N_c) > \mu(N_c)$, the throughput is mainly constrained by $\mu(N_c)$ (i.e., the capacity of transmission subchannel). Similarly, $\Gamma(N_c) = \mu(N_c) L_p$. In conclusion, we have

$$\Gamma(N_c) = min\,[\lambda(N_c), \mu(N_c)]\, L_p. \tag{6}$$

Because $\Gamma(N_c)$ is govern by the value of N_c, we desire to find the optimal N_c, $N_{c,opt}$, that maximizes $\Gamma(N_c)$, namely,

$$\begin{aligned} N_{c,opt} &= arg \max_{0\le N_c\le N} \Gamma(N_c) \\ &= arg \max_{0\le N_c\le N} min[\lambda(N_c), \mu(N_c)] L_p \\ &= arg \max_{0\le N_c\le N} min[\lambda(N_c), \mu(N_c)] \end{aligned} \tag{7}$$

Theorem 1 below proves that $N_{c,opt}$ uniquely exists and explicitly expresses $N_{c,opt}$ in terms of system parameters.

Theorem 1: For CSMA/CQ, the optimal subcarrier number $N_{c,opt}$ of the contention subchannel is given as follows.

$$N_{c,opt} = \frac{(aN-b-c)+\sqrt{(aN-b-c)^2+4abN}}{2a} \tag{8}$$

where

$$\begin{aligned} &0 < N_{c,opt} < N \\ &a = (1-p_{tr})\sigma + p_{tr}\left(T_{difs}+T_{sifs}\right) \\ &\quad -p_s\left(T_{cifs}+T_{sifs}\right) > 0 \\ &b = \frac{p_{tr}(L_{rts}+L_{cts})}{R_b} > 0 \\ &c = \frac{\left(L_h+L_p+L_{ack}\right)p_s}{R_b} > 0 \end{aligned}$$

Proof: Please refer to the Appendix.

Remarks: (a) Intuitively, to maximize $\Gamma(N_c)$, we require that that the enqueue rate should be equal to the dequeue rate, namely, $\lambda(N_c) = \mu(N_c)$. Note that $\lambda(N_c) \propto \emptyset N_c$ and $\mu(N_c) \propto \emptyset\frac{1}{N_c}$, namely, as N_c increases, $\lambda(N_c)$ increases but $\mu(N_c)$ decreases. We explain the reasons as follows.

- When $\lambda(N_c) > \mu(N_c)$, namely, the enqueue rate is larger than the dequeue rate, it means that a lot of nodes succeed in channel contention but cannot transmit their data. It hence implies that that when the total resources are fixed (i.e., N_c is fixed), more resources are allocated to the contention subchannel that to the transmission subchannel. In this situation, however, the system throughput is constrained by the capacity of the transmission subchannel. As a result, the resource overallocation to the contention subchannel is a waste.
- When $\lambda(N_c) < \mu(N_c)$, namely, the enqueue rate is less than the dequeue rate, it means that the number of the winners is small so that the transmission channel is often idle. It hence implies that that when the total resources are fixed (i.e., N_c is fixed), more resources are allocated to the transmssion subchannel that to the contention subchannel. In this situation, however, the system throughput is constrained by the capacity of the contention subchannel. As a result, the resource overallocation to the transmission subchannel is a waste.

In short, letting $\lambda(N_c) = \mu(N_c)$ can balance the capacities of the contention and transmission subchannels, maximally utilizing the system resources and producing the maximum system throughput.

(b) In practice, since the subcarrier number is a positive integer, $N_{c,opt}$ should be calculated as follows:

$$N_{c,opt} = \left\lfloor \frac{(aN-b-c)+\sqrt{(aN-b-c)^2+4abN}}{2a} \right\rfloor \in [0, N),$$

where $\lfloor\cdot\rfloor$ is the floor function. Note that $N_{c,opt} = 0$ means that in practice, the optimal solution does not exist.

TABLE 1. Default parameter settings in this paper.

n	20	DIFS	52 μs
N	48	R_b	1.125M bps
N_c	6	PHY Header	26 Byte
N_d	42	MAC Header	28 Byte
τ	4 μs	RTS	20 Byte
σ	20 μs	CTS	14 Byte
SIFS	12 μs	ACK	14 Byte
CIFS	12 μs	DATA	1000 Byte

V. PERFORMANCE EVALUATION

In this section, we evaluate the MAC-layer performance of CSMA/CQ via extensive simulations. The CSMA/CQ simulator is written using C++. The default parameter settings are shown in Table 1. In this table, the OFDM symbol time τ is set to 4 μs, which consists of the valid symbol time of 3.2 μs and a cyclic prefix time of 0.8 μs. Like 802.11g, we assume that the system has N=48 data subcarriers, which supports a data rate of 54 Mbps. Therefore, the per-subcarrier data rate is $R_b = 54/48 = 1.125$ Mbps. Each simulation value is an average over five simulation runs, where each run lasts for 100 seconds.

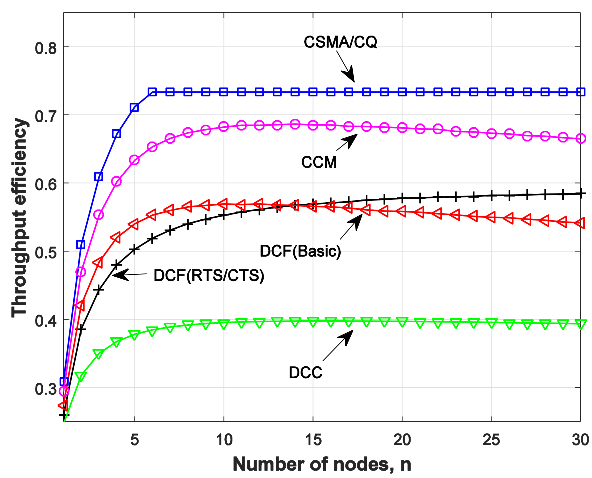


FIGURE 7. Comparison of throughput efficiency among CSMA/CQ, CCM [23], DCC [25], DCF with the basic access mode, and DCF with the RTS/CTS mode.

Figure 7 compares the throughput efficiency (i.e., the ratio between the network throughput and the PHY data rate) among CSMA/CQ, CCM [23], the dedicated control channel (DCC) [25] approach, DCF with the basic access mode, and DCF with the RTS/CTS mode, as the number of nodes, n, varies from 1 to 30. In this figure, we adopt the theoretical results (whose accuracy has been verified in [25]) for DCC, and the simulation results for other protocols. In CSMA/CQ, the PHY rate is set to 1.125∗6 Mbps for contention channel (consisting of 6 subcarriers) and to 1.125∗42 Mbps for data channel (consisting of 42 subcarriers). In CCM [23], the PHY rate is set to 54 Mbps for data channel and to 5.5 Mbps for contention channel. In DCC [25], the whole bandwidth is divided into 3 channels: one contention channel with the PHY rate = 1.125∗6 Mbps, and two data channels (each of which has a PHY rate = (54-1.125∗6)/2 Mbps. For DCF, the PHY rate of data and RTS/CTS/ACK is set to 54 Mbps. From this figure, we have the following observations.

- The throughput efficiency of CSMA/CQ first quickly increases to a maximum value and then keeps almost unchanged, as the number of nodes, n, increases. Moreover, the throughput efficiency always outperforms that of all other protocols. A rough calculation reveals that on average, CSMA/CQ has a potential of improving the throughput efficiency by more than 9% over CCM, by 82% over DCC, and by 30% over 802.11 DCF with the RTS/CTS mode and by 30% over 802.11 DCF with the basic mode. This is because our design can enable contention and data channels to be executed in parallel and our method can optimize the bandwidth allocation of each channel.
- The efficiency of CCM is 9% less than that of our design, but is higher than those of other protocols. One reason is: we assume that CCM has a higher total PHY rate (consisting of 54 Mbps for data channel and 5.5 Mbps for contention channel) than other protocols (each of which has a total PHY rate of 54 Mbps). It is worth pointing out that our design still has many advantages over CCM, as mentioned in Section I-C.
- The efficiency of DCF with the basic access mode first increases and then decreases as n increases. The reason is that more contending nodes will cause more collisions of the DATA frames (rather than the short RTS frame in CSMA/CQ and DCF with the RTS/CTS mode), thereby wasting more time.
- The efficiency of DCC is worst. The main reason is that its two data channels are too dependent on the contention channel.

Figure 8 plots the throughput efficiency, when # of nodes, n, varies from 1 to 30, and N_c:N_d =3:45,6:42,8:40. From this figure, when n is fixed (say, n=10), the efficiency first increases and then decreases as N_c increases (say, the efficiency is 0.48, 0.73, and 0.7 when N_c = 3, 6, 8). This indicates that for a given N, there exists an optimal subcarrier allocation. The reason is that for a given N, increasing N_c will decrease N_d; as a result, increasing N_c will increase the capacity of the contention channel but decrease the capacity of the transmission channel, and vice versa. However, the throughput is constrained by the minimum of the two capacities, as shown in (6). Then a smaller or lager N_c

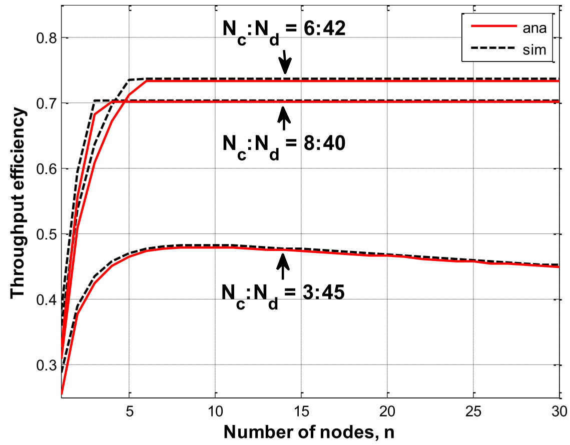


FIGURE 8. **Throughput efficiency as the ratio of N_c to N_d varies.**

will lead to a low throughput and hence these exists an optimal N_c that maximizes the system throughput.

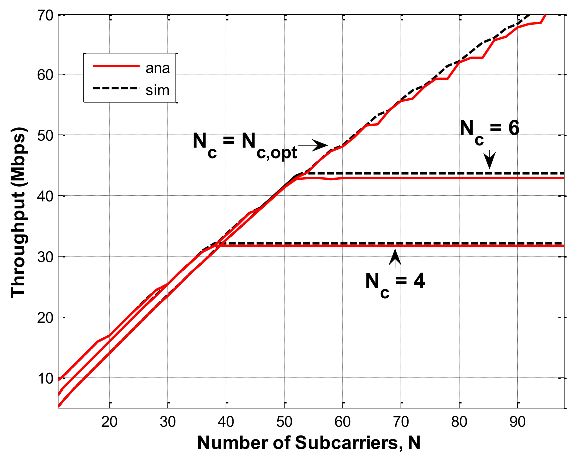


FIGURE 9. **Throughput vs. the total number of subcarriers N, when the node number n=20 and N_c = 4, 6 and $N_{c,opt}$.**

Figure 9 plots the throughput (in Mbps), when the # of total subcarriers, N, varies from 11 to 98, N_c = 4, 6 and $N_{c,opt}$, and n=20. The solid curves show the analysis results, whereas the dash curves show the simulation results. From this figure, we have the following observations.

- When N_c is fixed, the throughput first increases to a maximum value and then keeps almost unchanged, as N increases. For example, when N_c = 4, the throughput increases to the maximum value of 31 Mbps, and then keeps unchanged, as N incrases. The reason is that when N_c is fixed, increasing N will only increase N_d; this indicates that as N increases, the capacity of the contention subchannel keeps unchanged, but the capacity of the transmission subchannel will increase, as shown in IV-D. However, the system throughput is constrained by the minimum of the two capacities, as shown in (6). Therefore, below a critical value of N (say, N=36 for N_c =4), the system throughput is constrained by N_d

and keep increasing; above the critical value, the system throughput is constrained by N_c and keeps unchanged.

- When N_c is set to the optimal value $N_{c,opt}$, the system throughput keeps increasing as N increases. The reason is that as N increases, N_c and N_d increases simultaneously, implying that the two capacities of the contention and transmission subchannels increase simultaneously, and hence the system throughput keep increasing.

In addition, from this figure, our analysis results match well with the corresponding simulation results. A rough calculation reveals that the relative errors between the analysis and simulation results are, respectively, 0.80%, 1.11% and 0.79%, when $N_c = 4$, 6 and $N_{c,opt}$. This manifests that our theoretical model is very accurate.

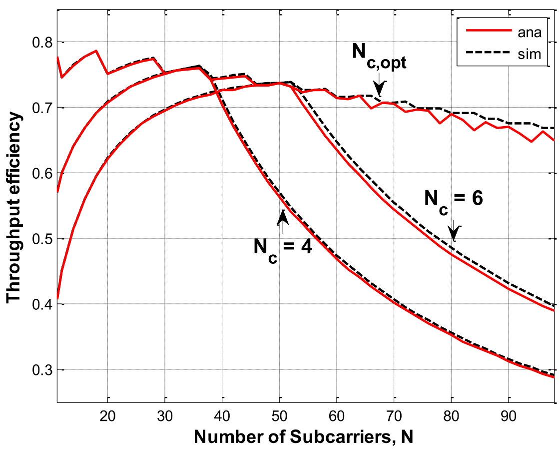


FIGURE 10. Throughput efficiency vs. the total number of subcarriers N, when the node number n=20 and $N_c = 4$, 6 and $N_{c,opt}$.

Figure 10 repeats Figure 9, except that the y-axis represents the throughput efficiency (i.e., the ratio between the throughput and the PHY data rate). From this figure, we have the following observations.

- When N_c is fixed, the throughput efficiency first increases to a maximum value and then decreases, as N increases. For example, when $N_c = 4$, the efficiency first increases to 0.76 and then decreases, as N increases. Note that the PHY data rate increase as N increases. We explain the reasons as follows. Below a critical value of N (say, N=36 when Nc=4), the system throughput is constrained by Nd and increases as N increases; at the same time the proportion of the system spending in packet transmission increases as the PHY data rate increases. Above the critical value, the system throughput is constrained by N_c and keeps unchanged; and therefore the throughput efficiency (i.e., the ratio between the throughput and the PHY data rate) decreases, as the PHY data rate increases.
- When N_c is set to the optimal value $N_{c,opt}$, the throughput efficiency has a decreasing trend, as N increases. The reasons are as follows. Note that the inherent overheads (such as DIFS and waiting time) keep unchanged, as N increases. In the optimal subcarrier allocation, as N increases (and hence the PHY data rate increases), the packet transmission time decreases but the overhead keeps unchanged; as a result, the throughput efficiency decreases. This indicates that when the PHY data rate increases, the throughput increases; however, the increment in throughput is not as much as the increment in the PHY date rate. In order to obtain the commensurate increment in throughput, we should reduce the inherent overhead (such as the waiting time overhead in the contention subchannel).

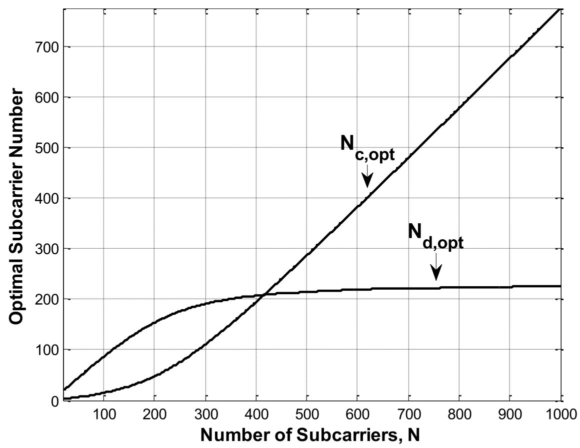


FIGURE 11. Optimal contention subcarrier number $N_{c,opt}$ and optimal transmission subcarrier number $N_{d,opt}$ vs. the total number of subcarriers N, when the node number n=20.

Figure 11 plots the optimal subcarrier number, $N_{c,opt}$ and $N_{d,opt}$, as the total subcarrier number N increases. From this figure, we can see that as N becomes larger and larger, $N_{d,opt}$, increases very slowly but $N_{c,opt}$, increases very fast. The main reason is that the time-domain contention efficiency will quickly converge to a limit (say, 0.371163 when the packet size = 1000 bytes from [35]), while the transmission efficiency still undergoes an approximately proportional increase from IV-D, as the PHY data rate increases. However, the system throughput is constrained by the minimum throughput efficiency of the two processes, as shown in (6). As a result, the system must allocate far more subcarrier to the contention subchannel than to the transmission subchannel, trading for increasing the system efficiency a little. This indicates that when the number of the total available subcarriers increases, the system efficiency limit is mainly constrained by the contention process. We can either improve the time-domain contention efficiency via some method (say, the idle sense scheme [27]) or replace the time-domain contention mechanism by the frequency-domain contention mechanism (such as such as T2F [18] and Back2F [20]), in order to improve the system throughput efficiency of CSMA/CQ significantly.

VI. CONCLUSION

Conventional 802.11 MAC protocols employ a simple distributed control to elegantly coordinate the channel access and the data transmission among multiple nodes. However, with the physical data rate growing and the various multimedia applications being deployed, this purely distributed control is hindering the improvement of the throughput efficiency and becomes very hard to meet the diverse QoS requirements of these applications. In this paper, borrowing the function separation and programmable idea of SDN, we propose the CSMA/CQ scheme to address these problems. Our scheme inherits the merit of the conventional distributed random channel access and enables nodes to perform data transmission in a TDMA manner, at the same time introduces a contention queue to coordinate the concurrent execution of the channel contention and the data transmission. The concurrent execution significantly improves the throughput efficiency of the conventional 802.11 networks, while the contention queue enables a centralized entity to enforce more diverse and flexible controls over the whole network. In addition, we also propose a theoretical model to optimize bandwidth allocation, and run extensive simulations to verify the efficiency of CSMA/CQ and the accuracy of our theoretical model. This study provides novel ideas in designing software-defined wireless networks.

APPENDIX

A. PROOF OF THEOREM 1

Note that as $N_c \in [0, N]$ increases, $\lambda(N_c)$ increases but $\mu(N_c)$ decreases. We prove Theorem 1 in two steps.

Step 1: $N_{c,opt} \in (0, N)$ is the unique intersection point between curves $\lambda(N_c)$ and $\mu(N_c)$.

From (2)-(3), IV-C, and IV-D, we have

$$\begin{aligned}\Omega(N_c) &= (1-p_{tr})\sigma + p_s T_s(N_c) + (p_{tr}-p_s) T_c(N_c)\\ &= (1-p_{tr})\sigma + p_{tr} T_s(N_c)\\ &= (1-p_{tr})\sigma + p_{tr}\left(T_{difs} + \frac{L_{rts}+L_{cts}}{R_b N_c} + T_{sifs}\right)\\ &= (1-p_{tr})\sigma + p_{tr}\left(T_{difs}+T_{sifs}\right)\\ &\quad + \frac{p_{tr}(L_{rts}+L_{cts})}{R_b}\frac{1}{N_c}\end{aligned}$$

$$\begin{aligned}\lambda(N_c) &= \frac{p_s}{\Omega(N_c)}\\ &= \frac{p_s}{\sigma + p_{tr}\left(T_{difs}+T_{sifs}\right) + \frac{p_{tr}(L_{rts}+L_{cts})}{R_b}\frac{1}{N_c}}\end{aligned}$$

$$\mu(N_c) = \frac{1}{\left(T_{cifs}+T_{sifs}\right) + \frac{L_h+L_p+L_{ack}}{R_b}\frac{1}{N-N_c}}$$

To find the intersection point of the two curves, let $\lambda(N_c) = \mu(N_c)$. After some algebraic manipulations, we have

$$-aN_c^2 + (aN-b-c)N_c + bN = 0,$$

Solving the above equation, we have

$$N_{c,opt} = \frac{(aN-b-c) \pm \sqrt{(aN-b-c)^2 + 4abN}}{2a}.$$

On one hand, we have

$$\begin{aligned}&\frac{(aN-b-c) - \sqrt{(aN-b-c)^2+4abN}}{2a}\\ &\quad < \frac{(aN-b-c) - |aN-b-c|}{2a} \le 0,\end{aligned}$$

On the other hand,

$$\begin{aligned}&\frac{(aN-b-c) + \sqrt{(aN-b-c)^2+4abN}}{2a}\\ &\quad > \frac{(aN-b-c) + |aN-b-c|}{2a} \ge 0.\end{aligned}$$

In addition, we note that

$$\begin{aligned}&4abN < 4aN(b+c)\\ &\quad\Rightarrow -2aN(b+c) + 4abN < 2aN(b+c)\\ &\quad\Rightarrow (aN-b-c)^2 + 4abN < (aN+b+c)^2\\ &\quad\Rightarrow \sqrt{(aN-b-c)^2+4abN} < aN+b+c\\ &\quad\Rightarrow (aN-b-c) + \sqrt{(aN-b-c)^2+4abN} < 2aN\\ &\quad\Rightarrow \frac{(aN-b-c) + \sqrt{(aN-b-c)^2+4abN}}{2a} < N,\end{aligned}$$

Since the subcarrier number is positive, $N_{c,opt}$ should be expressed by (8), namely,

$$N_{c,opt} = \frac{(aN-b-c) + \sqrt{(aN-b-c)^2+4abN}}{2a}.$$

In short, $N_{c,opt} \in (0, N)$ is the unique intersection point between curves $\lambda(N_c)$ and $\mu(N_c)$.

Step 2: $N_{c,opt}$ is the unique value that maximizes the system throughput of CSMA/CQ.

Note that $\lambda(N_c)$ is increasing monotonically and $\mu(N_c)$ is decreasing monotonically in $N_c \in [0, N]$, while $N_{c,opt} \in (0, N)$ is the unique intersection point between curves $\lambda(N_c)$ and $\mu(N_c)$ and therefore we have $\lambda\left(N_{c,opt}\right) = \mu\left(N_{c,opt}\right)$.

When $N_c \le N_{c,opt}$, we have $\lambda(N_c) \le \lambda\left(N_{c,opt}\right)$ and $\mu\left(N_{c,opt}\right) \le \mu(N_c)$, and hence

$$\begin{aligned}&\max_{0\le N_c\le N_{c,opt}} min\left[\lambda(N_c), \mu(N_c)\right]\\ &\quad = \max_{0\le N_c\le N_{c,opt}} \lambda(N_c)\\ &\quad = \lambda\left(N_{c,opt}\right)\\ &\quad = \mu\left(N_{c,opt}\right)\end{aligned}$$

When $N_c \ge N_{c,opt}$, we have $\lambda(N_c) \ge \lambda\left(N_{c,opt}\right)$ and $\mu\left(N_{c,opt}\right) \ge \mu(N_c)$, and hence

$$\begin{aligned}&\max_{N_{c,opt}\le N_c\le N} min\left[\lambda(N_c), \mu(N_c)\right]\\ &\quad = \max_{N_{c,opt}\le N_c\le N} \mu(N_c)\\ &\quad = \mu\left(N_{c,opt}\right)\end{aligned} \tag{9}$$

Then, from (7), $N_{c,opt}$ is the unique value that maximizes the system throughput of CSMA/CQ.

COMPETING INTERESTS

The authors declare that there is no conflict of interests regarding the publication of this paper.

REFERENCES

[1] J. G. Proakis and M. Salehi, *Digital Communications*. New York, NY, USA: McGraw-Hill, 2008.

[2] J. Fang *et al.*, "Fine-grained channel access in wireless LAN," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 772–787, Jun. 2013.

[3] *IEEE 802.11-WLAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 02.11-2007, Jun. 2007.

[4] *IEEE 802.11n-Enhancements for Higher Throughput, Amendment 4 to IEEE 802.11 Part 11: WLAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard IEEE 802.11n/D3.11, Dec. 2007.

[5] E. Perahia and M. X. Gong, "Gigabit wireless LANs: An overview of IEEE 802.11ac and 802.11ad," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 15, no. 3, pp. 23–33, Jul. 2011.

[6] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding congestion in IEEE 802.11b wireless networks," in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, Oct. 2005, p. 25.

[7] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[8] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[9] K. Greene. (2009). *MIT Tech Review 10 Breakthrough Technologies: Software-Defined Networking*. [Online]. Available: http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/

[10] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[11] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Athanasios, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 4–18, Feb. 2014.

[12] D. Zhao, M. Zhu, and M. Xu, "Leveraging SDN and OpenFlow to mitigate interference in enterprise WLAN," *J. Netw.*, vol. 9, no. 6, pp. 1526–1533, 2014.

[13] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's WLANs," in *Proc. ACM Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2013, pp. 387–398.

[14] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANs with Odin," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, New York, NY, USA, 2012, pp. 115–120.

[15] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: Towards software defined WLANs," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 4, pp. 42–45, 2013.

[16] H. Ali-Ahmad *et al.*, "Crowd: An SDN approach for densenets," in *Proc. Softw. Defined Netw. (EWSDN), 2nd Eur. Workshop* Piscataway, NJ, USA, Oct. 2013, pp. 25–31.

[17] A. Rangisetti *et al.*, "Load-aware hand-offs in software defined wireless LANs," in *Proc. IEEE 10th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Piscataway, NJ, USA, Oct. 2014, pp. 685–690.

[18] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Listen (on the frequency domain) before you talk," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, 2010, Art. no. 16.

[19] P. Huang, X. Yang, and L. Xiao, "WiFi-BA: Choosing arbitration over backoff in high speed multicarrier wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 771–779.

[20] S. Sen, R. R. Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *Proc. ACM Mobicom*, 2011, pp. 241–252.

[21] J. Deng and Z. J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Proc. IEEE ICUPC*, Oct. 1998, pp. 973–977.

[22] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-L. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proc. I-SPAN*, Dec. 2000, pp. 232–237.

[23] P. Kyasanur, J. Padhye, and V. Bahl, "On the efficacy of separating control and data into different frequency bands," in *Proc. BROADNETS*, 2005, pp. 646–655.

[24] K. H. Almotairi and X. S. Shen, "A distributed multi-channel MAC protocol for ad hoc wireless networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 1, pp. 1–13, Jan. 2015.

[25] J. Mo, H.-S. So, and J. Walrand, "Comparison of multichannel MAC protocols," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 50–65, Jan. 2008.

[26] S. Kim, D. Kim, and Y. Suh, "A survey and comparison of multichannel protocols for performance anomaly mitigation in IEEE 802.11wireless networks," *Int. J. Commun. Syst.*, vol. 26, no. 10, pp. 1288–1307, Oct. 2013.

[27] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse WLANs," in *Proc. ACM SIGCOMM*, 2005, pp. 121–132.

[28] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems*, IEEE Standard 802.16-2009, 2009.

[29] *Physical Layer Procedures (Release 11)*, document TS 36.213 V11.1.0 (2012-02), 3GPP, 2012. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.213/36213-b10.zip

[30] S. Premnath, D. Wasden, S. K. Kasera, N. Patwari, and B. Farhang-Boroujeny, "Beyond OFDM: Best-effort dynamic spectrum access using filterbank multicarrier," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 869–882, Jun. 2013.

[31] N. Michailow *et al.*, "Generalized frequency division multiplexing for 5th generation cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3045–3061, Sep. 2014.

[32] L. B. Jiang and S. C. Liew, "Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 34–49, Jan. 2008.

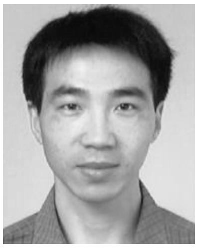
[33] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[34] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 588–601, Jun. 2007.

[35] Q. Zhao, D. H. K. Tsang, and T. Sakurai, "A novel CAC scheme for homogeneous 802.11 networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp. 1168–1174, Mar. 2010.

[36] F. Xu, Q. Zhao, Y. Zeng, J. Yang, H. Dai, and P. Dang, "A novel OFDM-based MAC protocol for wireless LANs," in *Proc. ChinaCom*, 2015, pp. 222–226.

[37] M. Park, "IEEE 802.11ah: Sub-1-GHz license-exempt operation for the Internet of Things," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 145–151, Sep. 2015.



QINGLIN ZHAO received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2005. From 2005 to 2009, he did research with The Chinese University of Hong Kong, Hong Kong, and then with The Hong Kong University of Science and Technology, Hong Kong. Since 2009, he has been with the Faculty of Information Technology, Macau University of Science and Technology, Macau, China. His research interests include wireless communications and networking, next-generation wireless local area networks, software-defined wireless networks, and physical-layer/medium access control co-design.

FANGXIN XU received the B.S. degree from the Beijing Institute of Technology, Zhuhai, China, in 2012, and the M.S. degree from the Macau University of Science and Technology, Macau, China, in 2015. His current research interests include 802.11 networks, full-duplex wireless communications, and software-defined radios.

JIE YANG received the B.S. degree from the Beijing Institute of Technology, Zhuhai, China, in 2012, and the M.S. degree from the Macau University of Science and Technology, Macau, China, in 2015. She is currently with the Faculty of Information Technology, Macau University of Science and Technology. Her current research interests include data base technique, data mining, and computer networks.

YUJUN ZHANG received the B.S. degree from Nankai University in 1999 and the Ph.D. degree from the Chinese Academy of Sciences in 2004. From 2009 to 2010, he was a Visiting Scholar with the University of California at Los Angeles, USA. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has authored or co-authored over 50 journal and conference papers. He holds 15 patents in China and has published one book on trusted network protocol. His current research interests include network security and future Internet architecture. He has served in technical committees of several national and international conferences.