Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 4015262, 19 pages
https://doi.org/10.1155/2019/4015262

Research Article

SDN-Based Centralized Downlink Scheduling with Multiple APs Cooperation in WLANs

Jianjun Lei, Ying Wang, and Ying Xia

School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Jianjun Lei; leijj@cqupt.edu.cn

Received 20 September 2019; Accepted 9 December 2019; Published 27 December 2019

Guest Editor: Rafik Zayani

Conventional DCF and RTS/CTS mechanisms perform the channel contention by a distributed and independent manner, which can lead to severe cochannel interference and low channel utilization in multiple APs dense deployment scenario. In this paper, we propose a channel scheduling cooperation algorithm called CCT-SDN (centralized concurrent transmission based on SDN) that enables multiple APs (Access Points) to perform cooperatively a centralized downlink transmission control, thus achieving higher system throughput and channel utilization by avoiding cochannel interference and implementing concurrent transmission. This design inherits the merit of the conventional distributed random channel access and adopts standardized OpenFlow protocol and Software Defined Network (SDN) architecture to make a centralized concurrent downlink traffic transmission decision among APs. Meanwhile, we also present a novel neighborhood relation storage scheme called SPRIM to enhance the retrieving efficiency of SDN controller, which enables CCT-SDN to perform a real-time control. Moreover, we also develop a theoretical model to prove the improvement of CCT-SDN. Furthermore, our solution does not require any modifications to existing ubiquitous 802.11 terminal devices and thus is likely to be widely deployed. Finally, extensive simulation results on Mininet-WiFi verify that CCT-SDN can achieve significant performance in terms of aggregate throughput, channel utilization, and packet loss rate in different deployment scenarios.

1. Introduction

Wireless local area networks (WLANs) have gained significant attention in both industry and academia [1] due to their flexibility of deployment and cost efficiency. With the dramatic increasing popularity of IEEE 802.11 technology, it is widespread in practice to deploy multiple APs (Access Points) with partially overlapping areas for providing ubiquitous continuous coverage. However, this results in serious cochannel or adjacent-channel interference and severely decreases the overall system performance, especially in dense deployment scenarios. The fundamental reason for this channel inefficiency is the fact that the basic Media Access Control (MAC) protocol utilizes the legacy distributed coordination function (DCF). The DCF based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) does not always make correct sensing decisions in the presence of exposed/hidden terminal problem [2] and also lacks concern to the asymmetry of data traffic on uplink and downlink in many applications. The virtual carrier sensing mechanism such as RTS/CTS uses the RTS or CTS frames to reserve the medium for the actual data transmission, which can handle the hidden terminal problem. However, it brings some extra transmission overhead and does not solve the exposed terminal problem in single-channel condition. For instance, for DCF or RTS/CTS, a transmission is deferred as long as the node senses the channel in status of busy. However, in certain cases, this new transmission may occur and it does not affect the ongoing transmission because their respective receivers do not locate in the same radio range. Namely, many potential valid transmissions are blocked unnecessarily, thus resulting in the waste of rare wireless channel resource. Additionally, in many deployments, the traffic of uplink and downlink is often severely asymmetrical

and several studies also report that the downlink traffic is about four times over the uplink traffic [3, 4]. Nevertheless, the legacy DCF mechanism often provides an equal opportunity for all APs and STAs (stations) to access the sharing channel, which causes severe AP congestion and performance degradation [5]. Basic DCF has been proven to be severely unsuitable for dense WLANs. Therefore, optimization of downlink traffic is often very critical to improve the whole system performance and has become the most important research issue especially in dense deployment scenarios.

Some existing methods such as smart antennas [6], carrier sense adaptation, and transmission power control [7] improve channel efficiency by a distributed concurrent scheduling manner. Therein, each node makes the concurrent transmission decision based on its relevant and local link information. Although these distributed schemes can construct partial concurrent transmission links, due to the lack of global perspective, it is hard to maintain immediately the transmission links once a node joins or leaves the network, which seriously affects the performance and stability of the network.

In this paper, we consider a centralized approach to schedule the downlink and improve the channel efficiency. Inevitably, the main challenge of centralized approach is to update or retrieve appropriately the status information of relevant nodes for convenience to make rapid decision and support real-time concurrent transmissions, especially in multiple APs deployment scenarios. Due to the Software Defined Networking (SDN) offering the more efficient configuration, higher flexibility, and finer control [8], we propose a CCT-SDN algorithm that is based on the SDN architecture and can provide an efficient cochannel concurrent transmission procedure. However, as mentioned above, an inherent challenge in SDN lies in the fact that the SDN controller needs to obtain the real-time network state information (NSI), which includes all transmission state information (TSI) and neighborhood relation information (NRI). The NSI is usually imperfect due to the latency caused by the decision procedure of the central SDN controller, thus leading to the impossible implementation for centralized channel control algorithm. In order to tackle this problem, we also present an efficient information processing algorithm called SPRIM to generate neighborhood relation matrix (NRM), which enables SDN controller to update and maintain timely transmission list (TL) and receiving list (RL) according to TSI through its data plane. During the transmission, the concurrent transmission decision module can be triggered and make a real-time decision, while SDN controller receives a data transmission request from any AP. In detail, when receiving a data transmission request from the AP, the SDN controller will determine whether the new link will interfere with the ongoing link according to NRM, TL, and RL. This design enables more downlink concurrent transmissions and avoids unnecessary collision, thus improving significantly the channel utilization.

The main contributions of this paper are summarized as follows:

(i) Firstly, we design a programmable three-tier SDWN architecture to implement a centralized concurrent transmission scheme. Meanwhile, the CCT-SDN does not need any modification for STAs and has backward compatibility with the legacy WiFi network.

(ii) Secondly, due to the efficiency of the decision algorithm of controller impact on the accuracy of NSI, we put forward SPRIM to enhance decision efficiency by efficient information storing and retrieving mechanism, which makes it possible to implement the centralized CCT-SDN.

(iii) Thirdly, we establish a numerical analysis model for CCT-SDN. It is theoretically proven that the system throughput and channel utilization can be improved by increasing the downlink concurrent transmission and mitigating the channel collision.

(iv) Lastly, we evaluate extensively the performance of CCT-SDN in many different application scenarios including different downlink traffic ratio, overlapping degree, number of nodes, and so forth, and simulation results verify that CCT-SDN can achieve considerable performance gain in throughput, packet loss rate, and channel utilization.

The rest of this paper is organized as follows. We summarize related work in Section 2. The system architecture and concurrent transmission algorithm are presented in Section 3. In Section 4, we describe the numerical analysis model. The performance simulations are carried out in Section 5. Finally, we conclude this paper in Section 6.

2. Related Work

Due to the scarcity of wireless resources, how to effectively utilize wireless resources is the most popular research issue. Currently, many state-of-the-art extensions to traditional DCF have been proposed to mitigate cochannel interference in dense WLANs, such as cochannel concurrency transmission [6, 9–12] and transmission power control (TPC) [13–16]. In [9], Vutukuru et al. propose CMAP algorithm to build a conflict map for each node by using empirical observations of packet loss and the map differentiating manner between interfering and noninterfering links. All nodes can decide whether to transmit data immediately by listening to the ongoing transmissions and consulting the map. However, the CMAP can suffer from severe redundant retransmissions when acknowledging packet lost during concurrent transmissions. Hence, in [10], Yao et al. propose IRMA to use a signature detection method combating the control frame collision problem existing in [9], which designs a new NAV updating scheme in MAC layer to differentiate the interference range of different transmission links, thus promoting the concurrency of all noninterfering links. Nevertheless, it brings high extra overhead when the network topology changes frequently. In [11], they further propose an Interference-Aware Power Control (IAPC) protocol to avoid interference by transmitting CTS and

exploit concurrent transmission by letting CTS deliver the transmission power information to the neighboring nodes. However, IAPC needs each node to maintain the control information and signatures, which increases the nodes' burden. The C2SMA/CA [6] is another cochannel concurrent transmission scheme by estimating interference based on position information. The concurrent transmission may not be always efficient enough due to the fact that its operations are based on the assumption of omnidirectional transmission. In [12], the authors extend C2SMA/CA and propose a new concurrency-decision-making scheme based on beamforming for cochannel directional concurrent transmission. In [13], Zhang et al. propose PRKS to ensure the required link reliability and achieve a channel spatial reuse through local distributed coordination, which increases the burden of node to maintain local signal maps. Other studies on improving the channel utilization in WLANs include TPC technology and its extensions. In [14], the authors present a spatial reuse algorithm integrating with Dynamic Sensitivity Control (DSC) and transmission power control (TPC), which only supports two-BSS (Basic Service Set) network operating and uplink traffic scheduling. In [15], the authors mitigate the interference by optimizing the transmission power and Clear Channel Assessment (CCA) threshold jointly. However, the power adjusting algorithm for STA-specific CCA threshold is too complicated to deploy in reality. In [16], the authors focus on CST (Carrier Sensing Threshold) adaptation while maintaining optimal data rate to achieve high aggregate throughput. Because the CST adaptation can detect Signal of Interest formed by short communication links in dense network and eventually avoid collisions; in [17], they further propose another CST adaptation scheme that eliminates hidden terminal and exposes terminal problems by considering margin value. However, these two CST adaptation schemes can result in starvation for some nodes in dense environment. In [18], Liu et al. propose an Optimal Node Activation Multiple Access (ONAMA) protocol, which adopts a Distributed Maximal Independent Set (DMIS) algorithm to generate a scheduling slot table but may bring bigger extra cost for each node due to storing much intermediate states for pipelined precomputation. In [19], Zhao et al. propose a concurrent transmission mechanism (CTM) to mitigate pan-hidden/exposed-node problems, which constrains the interference among the same type of frames by a strict time schedule mechanism. It is similar to the traditional RTS/CTS and the efficiency will degrade when transmitting many small data packets.

In recent years, SDN has been a promising alternative architecture and has attracted significant interest from the academic and the industrial communities [20]. It decouples the network control and data forwarding functions and provides a global network perspective for a variety of applications [21]. Thus, it also can be adopted easily as a novel paradigm for WLANs, which enables the controller to mitigate signal interference and improve further the channel efficiency by a centralized framework. For instance, in [22], a SDWN-based framework called OpenRF is proposed to manage MIMO (multiple-input multiple-out) interference among APs, which adjusts the relative power used to transmit the traffic on each of the AP's antennas to degrade the cochannel interference. In [23], the authors adopt a centralized SDWN architecture to schedule downlink for multiple APs in WLANs, where all activated downlinks are cooperated to reduce the occurrence of collision by combining the DCF and a centralized queue scheduling algorithm for APs. Except these works applying the SDN architecture to mitigate the interference, other studies based on SDWN architecture are also proposed to solve the problems such as load balancing [24, 25], service differentiation [26–28], and resource allocation [29]. In this paper, borrowing the function separation idea of the SDN architecture, we propose CCT-SDN to schedule intensively all downlinks among APs for solving the existing cochannel interference problems in multiple APs deployment scenarios. CCT-SDN aims at performing concurrent transmission and avoiding possible collisions for all downlinks, utilizing the centralized control and global network perspective supported by the SDN controller. By integrating a fast retrieving algorithm, the controller can decide in real time whether or not a downlink packet can be transmitted by an AP. Deferring from [23], the CCT-SDN can tackle the exposed/hidden terminal problem simultaneously by looking up simply the table (NSI) and without any extra requirement supported from other protocols such as DCF, and also no modification is required to existing 802.11 STAs; thus it is easy to be widely deployed.

3. System Framework and Algorithm Description

In this section, we present the system framework and relevant algorithms of CCT-SDN. Different from traditional wireless network architecture, the CCT-SDN adopts the SDWN-based framework for scheduling downlink traffic for the concurrent transmission. And then we also describe the details of implementing CCT-SDN algorithm.

3.1. System Architecture. In CCT-SDN algorithm, the access control of uplink traffic from STAs to AP depends on CSMA/CA mechanism of each STA, but the downlink traffic data transmission from AP to STA is determined by the concurrent transmission decision modular in the controller. Thus, the CCT-SDN builds a three-tier blocks architecture and interfaces based on a SDN paradigm, which includes control plane and data plane. They communicate by a southbound interface protocol called OpenFlow [30]. As shown in Figure 1, the concurrent transmission decision modular is standardized as one of northbound applications and supported by some core modules such as NRM, transmission list (TL), and receiving list (RL). NRM contains the neighbor relation information of all nodes in the entire network, and TL and RL can be maintained and updated in

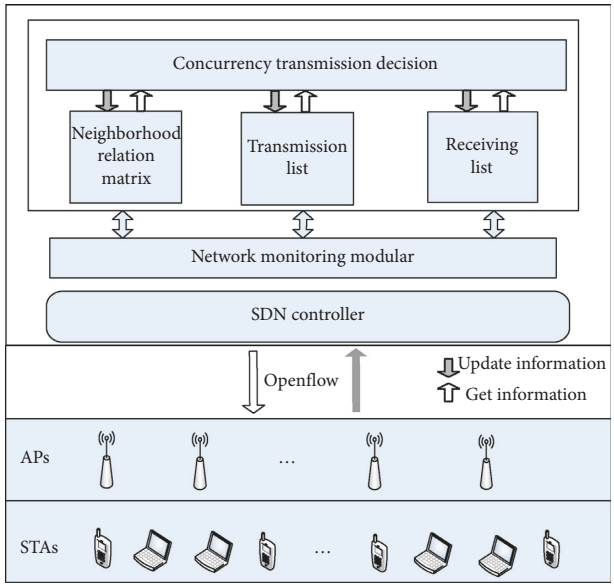


Figure 1: Framework of CCT-SDN.

real time by network monitoring modular embedded in the SDN controller. The controller as the core of control plane orchestrates the underlying physical wireless network entities. Since the OpenFlow standardizes the communication protocol between SDN switch and SDN controller [31], through it, the controller can not only send the concurrent transmission decision to the data plane but also obtain some information from data plane such as neighbor relation, transmission status, and receiving status of each node. The bottom tier of architecture is the data plane consisting of some APs, which not only can receive instructions from the controller but also can actively report events to the controller. According to this centralized downlink scheduling scheme, the CCT-SDN can manipulate some potential available links and void the invisible collision by some global information collected by the central SDN controller, thus improving the channel utilization.

3.2. CCT-SDN Algorithm. This section presents a detailed description of CCT-SDN algorithm. We firstly introduce the system model. Based on the model, we elaborate the concurrent transmission algorithm (CCT-SDN). Then, we describe the maintaining and updating mechanisms of RL and TL. Finally, we propose a SPRIM algorithm that can generate the NRM agilely according to NRI and significantly improve the efficiency of the controller and enable it to make the transmission decision in real time.

3.2.1. System Model. We consider an infrastructure-based network consisting of m APs and n competing STAs and they are deployed on the same channel. Hence, the total number of APs and STAs is N_{node} ($N_{\text{node}} = m + n$), and each AP is associated nearly with n/m STAs. All APs and STAs have the same transmission power and the transmission rate of each STA is determined via link adaptation mechanism. All APs are wired and controlled by an OpenFlow SDN controller. The signal coverage areas between adjacent APs overlap each other and all STAs are randomly distributed within the simulation area and each STA is associated with one AP based on its RSSI. For simplicity, we assume that the interference relationship between STAs is a binary phenomenon and each STA always has sufficient data to transmit. Finally, in order to focus on link scheduling, we also assume that it performs in a relative stable network and the interference graph can be achieved by some existing passive interference graph constructing technologies such as [32].

We define A as a node's information matrix; $A \in \{1, m+n\}1 \times (n+m)$, where

$$a_{1,i} = \begin{cases} k, & k \in (1, n), \text{ the } k \text{ represent the ID of APs,} \\ h, & h \in (n+1, n+m), \text{ the } h \text{ represent the ID of STAs.} \end{cases} \tag{1}$$

3.2.2. Concurrent Transmission Decision Algorithm. In CCT-SDN, the data transmission of the uplink utilizes the DCF mechanism with BEB [33]. However, the downlink transmission decision of APs is made by the controller governed by our concurrent transmission decision algorithm. Hence, the AP just sends the transmission request to the controller before it sends the data, even though the channel is busy. Figures 2(a)–2(d) show the cases for CCT-SDN algorithm for performing the data concurrent transmission for exposed terminal case and avoiding the data collision for hidden terminal case in two scenarios: loose coupling and tight coupling, respectively. The tight coupling scenario indicates that not only APs exhibits some signal coverage overlapping area but also they can sense the carrier signal each other. In loose coupling scenario, APs only have partial signal coverage overlapping area. As aforementioned, the SDN controller can make a concurrent transmission decision for the requested AP based on the determination that this new transmission packet can be delivered to the receiver successfully and does not interfere in the ongoing transmission. This detail will be presented in the following section.

To better explain our design, we first formulate the cases shown in Figure 2. We use a set U to store the information of nodes and assume that the indexes of AP2 and STA2 are i and j in set U, respectively. In addition, we define $R(i)$ and $T(j)$ as the receiving matrix of the i-th node and the transmitting matrix of the j-th node in U. And $F[i]$ and $L[i]$ are the indexes of the first and last neighbor node of sending node in set U. Similarly, $F[j]$ and $L[j]$ are the indexes of the first and last neighbor node of receiving node in set U. We define a variable $F_{i,j}$, which is expressed as follows:

$$F_{i,j} = C_{V,R,i} + T_{V,T,j}, \tag{2}$$

where $C_{V,R,i}$ represents whether one of neighbor nodes of the i-th node is receiving data and $T_{V,T,j}$ represents whether one of neighbor nodes of the j-th node is sending data. They are given by

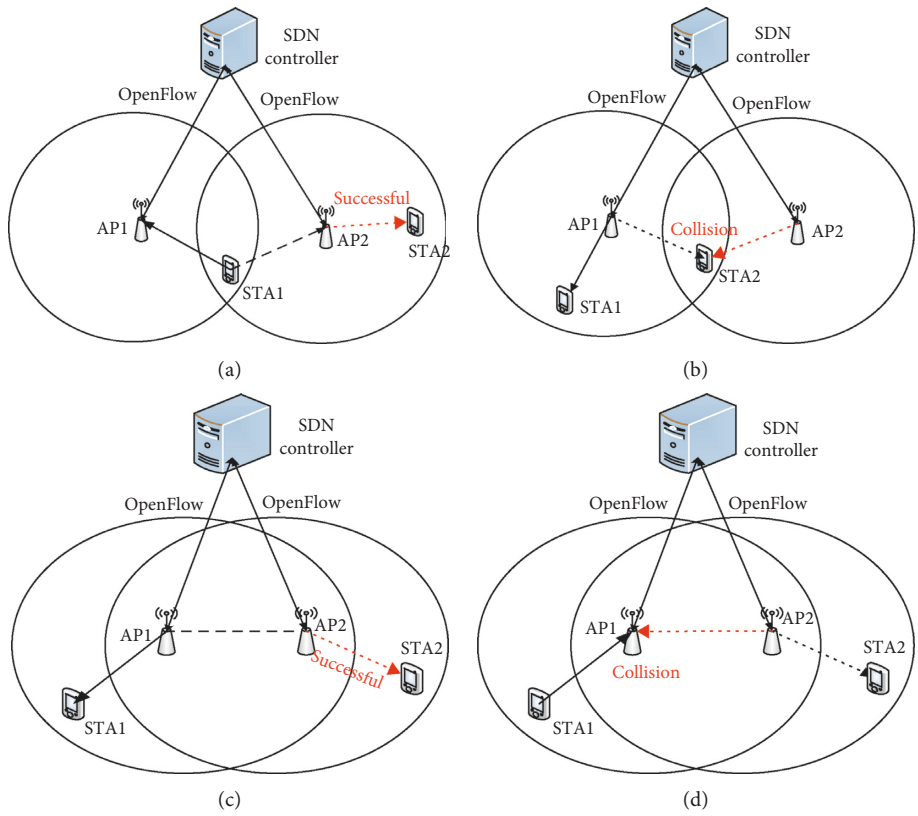


FIGURE 2: Hidden/exposed terminal problems in loose and tight coupling scenarios. (a) Exposed terminal case in loose coupling. (b) Hidden terminal case in loose coupling. (c) Hidden terminal case in tight coupling. (d) Exposed terminal case in tight coupling.

$$
\begin{aligned}
C_{V,R,i} &= V(i) \times \mathrm{RT}(i) \\
&= V(i)_{11} \times \mathrm{RT}(i)_{11} + \cdots + V(i)_{1(L[i]-F[i]+1)} \\
&\quad \times \mathrm{RT}(i)_{(L[i]-F[i]+1)1} \\
&= \sum_{k=1}^{L[i]-F[i]+1} V(i)_{1k} \times \mathrm{RT}(i)_{k1}, \\
T_{V,T,j} &= V(j) \times \mathrm{TT}(j) \\
&= V(j)_{11} \times \mathrm{TT}(j)_{11} + \cdots + V(j)_{1(L[j]-F[j]+1)} \\
&\quad \times \mathrm{TT}(j)_{(L[j]-F[j]+1)1} \\
&= \sum_{k=1}^{(L[j]-F[j]+1)} V(j)_{1k} \times \mathrm{TT}(j)_{k1}.
\end{aligned} \tag{3}
$$

$V(i)$ and $\mathrm{RT}(i)$ are the neighbor relation matrix and transposed matrix of $R(i)$ for the i-th node. And $V(j)$ and $\mathrm{TT}(j)$ are the neighbor relation matrix and transposed matrix of $T(j)$ for the j-th node. Clearly, $C_{V,R,i} = 1$ when any neighbor is transmitting data; otherwise $C_{V,R,i} = 0$. Similarly, $T_{V,R,j} = 1$ when any neighbor is receiving data; otherwise, $T_{V,R,j} = 0$. Therefore, shown specifically in Figure 2(a) case, even though the ongoing transmission exists from STA1 to AP1, we also can enable the concurrency transmission from AP2 to STA2 if $F_{i,j} = 0$. However, in DCF, this concurrency is not permitted due to its CDMA/CA mechanism.

To implement the concurrent transmission, the AP will send all transmission requests to the SDN controller, including the information of receiver too. According to the concurrent strategy, the SDN controller can make the concurrent transmission decision and respond to the AP immediately. The pseudocode for the concurrency transmission procedure is shown in Algorithm 1.

The concurrent transmission decision algorithm is triggered by the transmission request event from AP. At first, the SDN controller gets the neighbor relation matrix of sender $V[i]$ and the neighbor relation matrix of receiver $V[j]$ (line 1). Afterwards, it calculates $R(i)$ and $T(j)$ according to $V[i]$ and $V[j]$ (lines 2 and 3). Then, the transposed matrix $\mathrm{RT}(i)$ and $\mathrm{TT}(j)$ can be calculated by $R(i)$ and $T(j)$ (lines 4 and 5). Finally, the SDN controller can get $F_{i,j}$ via (2) (line 6). If $F_{i,j} = 0$, this means that no sender's neighbor nodes are receiving data and no receiver's neighbor nodes are sending data. Hence the new transmission from the AP is acknowledged and the transmission information and the receiving status information of receiver for AP are updated via calling Algorithm 2 (lines 7 and 8).

3.2.3. Maintaining TL (Transmission List) and RL (Receiving List). TL and RL are two sets storing the transmission status and receiving status of nodes in U, respectively. If $\mathrm{TL}[i] = 0$, it denotes that the i-th node in U is sending data, and if $\mathrm{RL}[i] = 0$, it denotes that the i-th node in U is receiving data. The SDN controller updates the TL and RL in real time through the OpenFlow message from AP. Hence, the receiving matrix $R(i)$ and transmission matrix $T(i)$ of the i-th node in

Require: a AP send a transmission request to the SDN controller
Input: NRM, TL, RL, F, L, V, sender is $U[i]$, and receiver is $U[j]$
Output: concurrent transmission instruction (CTI) or no response
Temporary variables: CTI, RT (i) and TT (j) are the transposed matrix of R (i) and T (j) respectively
(1) get V (i) and V (j) from set V
(2) get $F[i]$ and $L[i]$ from set F and L, $R(i) \longleftarrow$ RL $[F[i] : L[i]]$
(3) get $F[j]$ and $L[j]$ from set F and L, $T(j) \longleftarrow$ TL $[F[j] : L[j]]$
(4) RT (i) $\longleftarrow$ the transposed matrix of R (i)
(5) TT (j) $\longleftarrow$ the transposed matrix of T (j)
(6) $F_{i,j} = V(i) * \mathrm{RT}(i) + V(j) * \mathrm{TT}(j)$
(7) *if* $F_{i,j} == 0$ then
(8) Enable CTI to AP and run Algorithm 2 to change the transmission and receiving status
(9) *else*
(10) no response to AP
(11) *end if*

Algorithm 1: Concurrency transmission decision algorithm.

Require: This algorithm runs all the time.
Input: U, sender $U[i]$, receiver $U[j]$
Update: Set TL and RL store the transmission and receiving status of nodes in U
Temporary variables: RL and TL
(1) initialize RL = RL $[0] * |U|$ and TL = TL $[0] * |U|$
(2) *when* status changing event arrives do
(3) *if* SDN controller receives a data transmission start instruction then
(4) TL $[i] \longleftarrow 1$
(5) RL $[j] \longleftarrow 1$
(6) *else if* SDN controller receives a data transmission end instruction then
(7) TL $[i] \longleftarrow 0$
(8) RL $[j] \longleftarrow 0$
(9) *end if*
(10) update the TL and RL stored in SDN controller

Algorithm 2: Maintaining and updating RL and TL.

U are $R(i) = [\mathrm{RL}[F[i]]\, \mathrm{RL}[F[i]+1] \cdots \mathrm{RL}[L[i]-1]\, \mathrm{RL}[L[i]]]$ and $T(i) = [\mathrm{TL}[F[i]]\, \mathrm{TL}[F[i]+1] \cdots \mathrm{TL}[L[i]-1]\, \mathrm{TL}[L[i]]]$, respectively. Algorithm 2 depicts the iterative procedure.

Algorithm 2 is introduced, firstly, to maintain and update the status of the nodes' current transmission and receiving; it will initialize two sets of RL and TL with length $|U|$ (line 1). Once the SDN controller receives a status changing event from AP, it will update the sender's transmission and receiver's receiving status (between lines 2 and 9). Synchronously, the SDN controller will update the TL and RL too (line 10). Algorithm 2 always runs and can provide real-time node status information for the concurrency transmission decision.

3.2.4. SPRIM Algorithm. As the number of nodes in the network dramatically increases, more and more NRIs are available. The NRI plays an important role in CCT-SDN, which influences concurrent transmission decisions. Correspondingly, the efficiency of CCT-SDN also affects the accuracy of NSI. Therefore, it is very important to enhance the storage and retrieval mechanism of NRI, since this cooperation is a real-time and online procedure. Aiming at this problem, we propose a SPRIM algorithm that is originated from the known PRIM [34] algorithm. The SPRIM algorithm selects the initial node based on the node degree matrix and updates the NRM according to the generated nodes' sequence rather than only the initial node in the PRIM selected randomly. The SPRIM algorithm can generate an NRM with the high concentration; namely, all nodes existing in neighbor relation can be stored as close as possible to facilitate the rapid retrieving information from bigger matrix. Hence, the SPRIM algorithm can significantly reduce the computational complexity and improve the efficiency by extracting rapidly valid data from the NRM in dense scenarios. As follows, we present some details of the SPRIM algorithm.

At first, we use an INRM (initial neighbor relation matrix) to store neighborhood relations of all nodes in the network according to set A and NRI. In CCT-SDN, the two nodes are considered to have a neighborhood relationship when the data transmissions of them will interfere with each other. Figure 3 depicts an example of matrix A and INRM.

$$A = \begin{bmatrix} \text{AP1} & \text{AP2} & \cdots & \text{STA1} & \cdots & \text{STA}m \end{bmatrix}$$

$$\text{INRM} = \begin{bmatrix} \text{dis (AP1, AP1)} & \text{dis (AP1, AP2)} & \cdots & \text{dis (AP1, STA1)} & \cdots & \text{dis (AP1, STA}m\text{)} \\ \text{dis (AP2, AP1)} & \text{dis (AP2, AP2)} & \cdots & \text{dis (AP2, STA1)} & \cdots & \text{dis (AP2, STA}m\text{)} \\ \vdots & \vdots & \ddots & & \vdots & \\ \text{dis (STA1, AP1)} & \text{dis (STA1, AP2)} & . & \text{dis (STA1, STA1)} & \cdots & \text{dis (STA1, STA}m\text{)} \\ \vdots & \vdots & \ddots & & \vdots & \\ \text{dis (STA}m\text{, AP1)} & \text{dis (STA}m\text{, AP2)} & \cdots & \text{dis (STA}m\text{, STA1)} & \cdots & \text{dis (STA}m\text{, STA}m\text{)} \end{bmatrix}$$

Figure 3: Example of matrix A and INRM.

The notation dis (x, y) (both x and y belong to set A) represents the interference distance between xth node and yth node in set A, which can be calculated through the network monitoring modular governed by the SDN controller.

We also define a matrix DEV = {dev(AP1), ..., dev(APn), dev(STA1), ..., dev(STAm)} to store the number of neighbor nodes and dev (x) represents the number of neighbor nodes for the x-th node in set A; it can be obtained as follows:

$$\text{dev}(x) = \sum_{k=0}^{n+m-1} Nlx, k, \tag{4}$$

where $N_{lx,k}$ represents the neighbor relation of the x-th node and the k-th node in set A and it can be given by

$$N_{lx,k} = \begin{cases} 1, & \text{dis}(x, k) \le \text{interference threshold}, \\ 0, & \text{else}. \end{cases} \tag{5}$$

The interference threshold can be calculated as in [35]. To enhance the storage and retrieval efficiency of the NRI, we propose SPRIM algorithm, which can make the storage location of these nodes existing in the neighbor relationship to be centralized as far as possible. Therefore, based on dev (x) in (4) and the INRM, we can get a new set U via SPRIM algorithm. In order to facilitate understanding, we assume that the new matrix U = [AP2 AP3 STA1 $\cdots$ APy STAx]. Like the INRM, the NRM also can be obtained according to the set U storing the neighbor relation of nodes with high concentration degree. Figure 4 gives an example of U and NRM.

In summary, the SPRIM algorithm is formally described in Algorithm 3.

Based on set U, the SPRIM algorithm can generate neighbor relation centralization NRM by adjusting INRM. Algorithm 3 shows the details of SPRIM algorithm. Firstly, the controller adds all the nodes with no neighbor nodes into set U and removes these nodes from set I and records the index (k) of nodes in set A with the minimum degree $\text{DEV}_{\min}$ ($\text{DEV}_{\min} > 0$) (between lines 3 and 10). Then it lets A $[k]$ be the initial node V_0, adds it into set U, and removes it from set I (line 11). The algorithm operates in a loop till set U covers all nodes in set A; that is, $|U| = |A|$ (line 12). In each iteration, the SDN controller will move a node that has the minimum distance from the node in set I to set U (lines 13 and 14) and updates the index of all nodes according to set U and set A (between lines 15 and 17). Finally, the SDN controller adjusts INRM according to U and index (between lines 18 and 29). Meanwhile, the SDN controller will record F, L, and V for all nodes. The sets U, F, L, V, and the matrix NRM are regarded as the output of algorithm (line 30). This information is always maintained by the SDN controller and can be provided to the concurrent transmission decision modular.

To evaluate the SPRIM, we run simulations 1000 times for different number of nodes and APs to get the average number of zeros by a matrix V (i), which can be used to reflect the centralization of nodes (referred as θ_{degree}) with the neighboring relation. V (i) stores the neighbor relation value from the first neighbor node to the last one. We assume that $V(i) = [\text{NRM}[I\_i][F[i]]\ \text{NRM}[I\_i][F[i]+1] \cdot \text{NRM}[I\_i][L[i]]]$. Here, I_i = Index $[i]$; the index of node in U $[i]$ is stored in set A.

Therefore, θ_{degree} that indicates the degree of centralization of NRM can be given from the following equation:

$$\theta_{\text{degree}} = \frac{\sum_{\text{i=0}}^{n+m-1} NZi}{N_{\text{node}}}, \tag{6}$$

where the notation NZ_i presents the number of zeros in V (i) $(0 \le i < n + m)$; $\sum_{i=0}^{n+m-1} \text{NZ}_i$ is the total value calculated by the number of zeros in the neighbor relation value of each node V (i) $(0 \le i \le n + m - 1)$.

To facilitate understanding, we explain V (i) by an example. Assuming that NRM $[I\_i]$ = [0001100001000], here, F $[i] = 3$ and L $[i] = 9$; hence V (i) = [1100001].

Figure 5 shows the simulation results for the SPRIM and random storing algorithms under different numbers of APs and STAs scenarios. It is noticed that our SPRIM algorithm outperforms the random algorithm. Namely, the SPRIM algorithm can improve the concentration degree of NRM and hence effectively enhances the efficiency of controller and degrades the inaccuracy of NSI as much as possible by attempting to centralize as possible the nodes existing in neighbor relation when storing them.

4. Numerical Analysis

In this section, we present an analytical model and theoretically prove that CCT-SDN can improve significantly throughput and channel utilization. For convenience, we

$$U = \begin{bmatrix} \text{AP2} & \text{AP3} & \text{STA1} & \cdots & \text{AP}y & \text{STA}x \end{bmatrix}$$

$$\text{NRM} = \begin{bmatrix} \text{NR (AP1, AP2)} & \text{NR (AP1, AP3)} & \text{NR (AP1, STA1)} & \cdots & \text{NR (AP1, AP}y\text{)} & \text{NR (AP1, STA}x\text{)} \\ \text{NR (AP2, AP2)} & \text{NR (AP2, AP3)} & \text{NR (AP2, STA1)} & \cdots & \text{NR (AP2, AP}y\text{)} & \text{NR (AP2, STA}x\text{)} \\ \text{NR (AP3, AP2)} & \text{NR (AP3, AP3)} & \text{NR (AP3, STA1)} & \cdots & \text{NR (AP3, AP}y\text{)} & \text{NR (AP3, STA}x\text{)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \text{NR (STA } (m-1)\text{, AP2)} & \text{NR (STA } (m-1)\text{, AP3)} & \text{NR (STA } (m-1)\text{, STA1)} & \cdots & \text{NR (STA } (m-1)\text{, AP}y\text{)} & \text{NR (STA } (m-1)\text{, STA}x\text{)} \\ \text{NR (STA}m\text{, AP2)} & \text{NR (STA}m\text{, AP3)} & \text{NR (STA}m\text{, STA1)} & \cdots & \text{NR (STA}m\text{ , AP}y\text{)} & \text{NR (STA}m\text{, STA}x\text{)} \end{bmatrix}$$

Figure 4: Example of U and NRM.

Inputs: Set of nodes A, initial neighbor relation matrix (INRM), and Degree of all nodes in set A (DEV)
Outputs: N_{node}, Set U, Set F and L store the index of first and last neighbor node for nodes in U in NRM, respectively, Set V stores the neighbor relation value of nodes and NRM is the target neighbor relation matrix
Temporary variables: V_0, $V_{\min}$, I, Index

(1) $I \longleftarrow A$, $U \longleftarrow \emptyset$, Index $\longleftarrow \emptyset$, $K = 0$, $\text{DEV}_{\min} = \text{maximum\_value}$
(2) *for* $i \longleftarrow 0$ to $N_{\text{node}} - 1$ do
(3) *if* DEV $[i] == 0$ then
(4) $U \longleftarrow U \cup \{I\ [i]\}$, $I \longleftarrow C_I\ (I \cap \{I\ [i]\})$
(5) *else*
(6) *if* DEV $[i] \leq \text{DEV}_{\min}$ then
(7) $\text{DEV}_{\min} \longleftarrow$ DEV $[i]$, $k \longleftarrow i$
(8) end if
(9) end if
(10) *end for*
(11) $V_0 \longleftarrow A\ [k]$, $U \longleftarrow U \cup \{A\ [k]\}$, $I \longleftarrow C_I\ (I \cap \{A\ [k]\})$
(12) Repeat until number of elements in $U == N_{\text{node}}$
(13) Pick $A\ [j]$ in set I with smallest dis (i, j), $(A\ [i] \in U$ and $A\ [j] \in I)$
(14) $U \longleftarrow U \cup \{A\ [j]\}$, $I \longleftarrow C_I\ (I \cap \{A\ [j]\})$
(15) *for* $i \longleftarrow 0$ to N_{node} do
(16) Index $[i] \longleftarrow$ (Index of $U\ [i]$ in $A) \cup$ Index
(17) *end for*
(18) *for* $i \longleftarrow 0$ to N_{node} do
(19) *for* $j \longleftarrow$ to N_{node} do
(20) NRM $[i][j] \longleftarrow$ INRM $[i]$ [index $[j]$]
(21) *if* NRM $[i][j] \leq$ interference threshold then
(22) NRM [i][j] $\longleftarrow 1$
(23) *else*
(24) NRM $[i][j] \longleftarrow 0$
(25) *end if*
(26) *end for*
(27) $F\ [i] \longleftarrow$ pick the index of first neighbor node for $U\ [i]$ in NRM [Index $[i]$]
(28) $L\ [i] \longleftarrow$ pick the index of last neighbor node for $U\ [i]$ in NRM [Index $[i]$]
(29) $V\ [i] \longleftarrow$ str (NRM [index [i]] $[F\ [i] : L\ [i]]$), $V\ [i] \longleftarrow$ int $(V\ [i])$
(30) *end for*
(31) return (U, F, L, V, and NRM)

Algorithm 3: The SPRIM algorithm.

also make some reasonable assumptions based on the model mentioned before.

4.1. Throughput. Since CCT-SDN algorithm can improve the throughput in the scenarios with severe cochannel interference, we first define the probability of successful transmission as P_s and let P_{tr} denote the probability that at least one transmission occurs in the considered back-off slot. As presented in [36], we have

$$P_{\text{tr}} = 1 - (1 - \tau)^{n_*+1}, \tag{7}$$

where τ is the transmission probability of a node for a given slot. n_* is the average number of nodes associated with each AP. Let X denote a new back-off counter value selected by the tagged nodes after a packet transmission in steady state; thus

$$\tau = \frac{1}{1 + E[X]}, \tag{8}$$

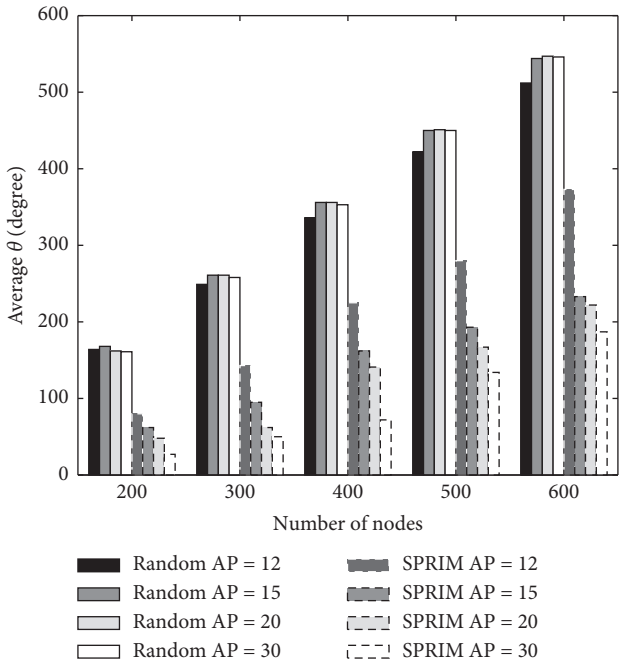


FIGURE 5: Average θ_{degree} value with various numbers of APs and STAs by using SPRIM and Random algorithms.

where $E[X]$ is the expectation of X. Let P_s denote the probability of successful transmission for legacy DCF; that is, $Ps = n_1\tau(1-\tau)^{(n_*-1)}$. Hence, the probability of successful transmission P_s^* can be approximated by

$$P_s^* = P_s + P_{sp} + P_{sc}, \tag{9}$$

where P_{sp} and P_{sc} denote the enhancing probability of successful transmission via enabling the downlink concurrent transmission and eliminating hidden terminals, respectively. Then P_{sp} and P_{sc} can be given as

$$\begin{aligned} P_{sp} &= (1-\tau)^{NA+E[n_s]}, \\ P_{sc} &= \frac{E[no](1-\tau)^{n_*+NA+E[n_s]}}{n_1}, \end{aligned} \tag{10}$$

where N_A denotes the number of neighbor APs for one AP. $E[n_s]$ represents the expectation of the number of neighbor nodes for one STA. $E[n_o]$ is the expectation of the number of STAs in overlapping areas. Therefore, (9) can be rewritten as

$$\begin{aligned} P_s^* = {} & n_*\tau(1-\tau)^{(n_*-1)} + (1-\tau)^{NA+E[n_s]} \\ & + \frac{E[n_o](1-\tau)^{n_*+NA+E[n_s]}}{n_*}. \end{aligned} \tag{11}$$

Analogously, we also can obtain the collision probability $P_c = P_{tr} - P_s - P_{sc}$. As a result, the aggregate throughput per BSS (Basic Service Set) can be calculated by

$$T^* = \frac{P_s^* E[L]}{\left(P_s^* - P_{sp}\right)T_s + P_c T_c + \left(1-P_{tr}\right)T_i}, \tag{12}$$

where $E[L]$ denotes the average packet payload size. T_s and T_c denote the average length of time periods for a successful packet and a collision packet transmission, respectively. Analogously, T_i is the duration of an empty slot. Hence, T_s and T_c are derived by

$$\begin{aligned} T_s &= \text{DIFS} + \text{SIFS} + \delta + T_{\text{payload}} + \delta + T_{\text{ACK}}, \\ T_c &= \text{DIFS} + \delta, \end{aligned} \tag{13}$$

where T_{payload} and T_{ACK} are transmission time for payload and ACK frame. DIFS and SIFS are Distributed and Short Inter-Frame Spaces. The notation δ denotes the propagation delay. Consequently, by substituting the above correlation formulas in (12), we can rewrite T^* as

$$\begin{aligned} T^* &= \frac{E[L]}{T_s - \left((1-\tau)^{NA+E[n_s]}T_s - \left(1-(1-\tau)^{n_*+1} - B_0\right)T_c - (1-\tau)^{n_*+1}T_i/(1-\tau)^{NA+E[ns]} + B_0\right)} \\ B_0 &= n_*\tau(1-\tau)^{(n_*-1)} + \frac{E[n_o](1-\tau)^{n_*+NA+E[n_s]}}{n_*}. \end{aligned} \tag{14}$$

Therefore, the system throughput for our algorithm can be calculated as $\sum_{i=1}^{m} T^*$. In addition, the throughput T in the legacy 802.11 DCF under cochannel deployment can be denoted as

$$T = \frac{P_s E[L]}{P_s T_s + \left(P_{tr} - P_s\right)T_c + \left(1-P_{tr}\right)T_i}. \tag{15}$$

It is noticed that T^* in (14) is a little different from the throughput T shown in (15), which is due to the difference of successful transmission probability for CCT-SDN and DCF. The legacy 802.11 DCF neither improves P_s via enabling concurrent transmission nor reduces collision probability P_c caused by hidden terminal. Hence, P_s in legacy DCF can be expressed as

$$\begin{aligned} P_s = {} & \left(1 - \frac{E[n_o]}{n_*}\right)(n_* - E[n_o])\tau(1-\tau)^{(n_*-1)} \\ & + \frac{E[n_o]}{n_*}E[n_o]\tau(1-\tau)^{E[n_s]}. \end{aligned} \tag{16}$$

4.2. Channel Utilization. In CCT-SDN, another target is to improve the channel utilization by increasing the probability of successful transmission. The actual channel utilization is very complicated because it is involved in traffic arrival mode and the packet size. Hence, as simplified in [37], we denote the channel utilization μ^* as

$$\mu^* = \frac{P_s^* T_{\text{payload}}}{\left(P_s^* - P_{sp}\right)T_s + P_c T_c + \left(1 - P_{tr}\right)T_i}. \quad (17)$$

Similarly, by substituting P_s^* in (16), the channel utilization can be rewritten as

$$\mu^* = \frac{T_{\text{payload}}}{T_s - \left((1-\tau)^{NA+E[n_s]}T_s - \left(1-(1-\tau)^{n_*+1} - B_0\right)Tc - (1-\tau)^{n_*+1}T_i/(1-\tau)^{NA+E[n_s]} + B_0\right)}, \quad (18)$$

$$B_0 = n_*\tau(1-\tau)^{(n_*-1)} + \frac{E[n_o](1-\tau)^{n_*+NA+E[n_s]}}{n_*}.$$

It is evident that, for given T_{payload}, P_{tr}, T_i, T_s, and T_c, the channel utilization μ^* can be obtained by (18). Therefore, the models of CCT-SDN and DCF can be evaluated theoretically by a given set of typical parameters, which are depicted in Figure 6.

We simulate a scenario with 2 APs and varying number of nodes to evaluate the CCT-SDN. We also change the percentage of nodes in the overlapping area (0.1, 0.3, and 0.5) and set the network in saturated condition ($\tau = 0.075$). The system throughput and channel utilization for CCT-SDN and DCF models are plotted in Figures 6(a) and 6(b), respectively. Obviously, as the number of nodes increases, the system throughput and channel utilization of both CCT-SDN and DCF decrease. However, CCT-SDN always surpasses DCF regardless as certain level of $E[n_0]$, which indicates that CCT-SDN can solve the exposed/hidden terminals problems and improve the system performance by the cooperation of the SDN controller.

5. Performance Evaluations

This section presents the simulation methodology for evaluating CCT-SDN under two different deployment scenarios: loose coupling and tight coupling. Further, the number of APs, number of stations in overlapping area, and the percentage of downlink traffic are also considered in our simulations. To benchmark the performance, three types of mechanisms are compared: legacy DCF, RTS/CTS, and CCT-SDN. We examine the performance of these algorithms in terms of aggregate throughput, channel utilization, and packet loss rate. Moreover, except verifying the feasibility of CCT-SDN, we also will validate its effectiveness in different application scenarios by deploying more APs and varying network density.

5.1. Evaluation Methodology. We choose Mininet-WiFi [38] to perform simulations in several different network density scenarios. The SDN controller RYU 1.2 runs on Ubuntu Kylin-14.04 with the dual 4-core CPU and 16G RAM Dell Server supported, and all APs are deployed on the same channel. iPerf is used to generate simultaneously the UDP traffic with a constant payload size of 1470 bytes in one packet. We use IEEE 802.11 g (with 54 Mbps data rate) as the physical layer protocol and assume that frames are lost only due to the channel collisions. The other parameters used in simulations are summarized in Table 1.

We consider a network with (1) 2 APs deployed in an area of 330 m × 200 m (loose coupling scenario) or 280 m × 200 m (tight coupling scenario) and (2) 4 APs deployed in an area of 330 m × 350 m (loose coupling scenario) or 280 m × 350 m (tight coupling scenario). Each AP is associated with 10 STAs with generating saturated data traffic to the AP. In tight coupling scenario, APs are uniformly placed in the area and they can sense each other. Correspondingly, in loose coupling scenario, APs only have some common signal coverage overlapping area. Therefore, in simulation, we also consider the different percentages of STAs (from 20% to 60%) deployed in the overlapping area, since they can generate different level cochannel interference. These scenarios are illustrated in Figure 7. Since the CCT-SDN only schedules the downlink traffic concurrent transmissions (the downlink traffic is dominant in the most applications), we investigate the gain by setting different ratios of downlink to total traffic (50%, 65%, and 80%) in simulations. In addition, the simulation results are derived from 20 simulations of 100 seconds in length.

5.2. Simulation in Basic Network Deployment

5.2.1. Different Number of STAs in Overlapping Area. Our main aim is to validate that CCT-SDN is able to maximize the system throughput and channel utilization while minimizing the packet loss rate in multiple APs WLANs. To this end, we first consider the case of 2 or 4 APs with different number of STAs in the overlapping area (each AP with fixed 10 associating STAs) and compute the system throughput, channel utilization, and packet loss rate. The results are presented in Figures 8–10.

Figure 8 shows the variation of the aggregate throughput of two schemes for each BSS versus the different percentage of STAs in overlapping areas, as well as different ratios downlink data traffic to total traffic (50%, 65%, and 80%).

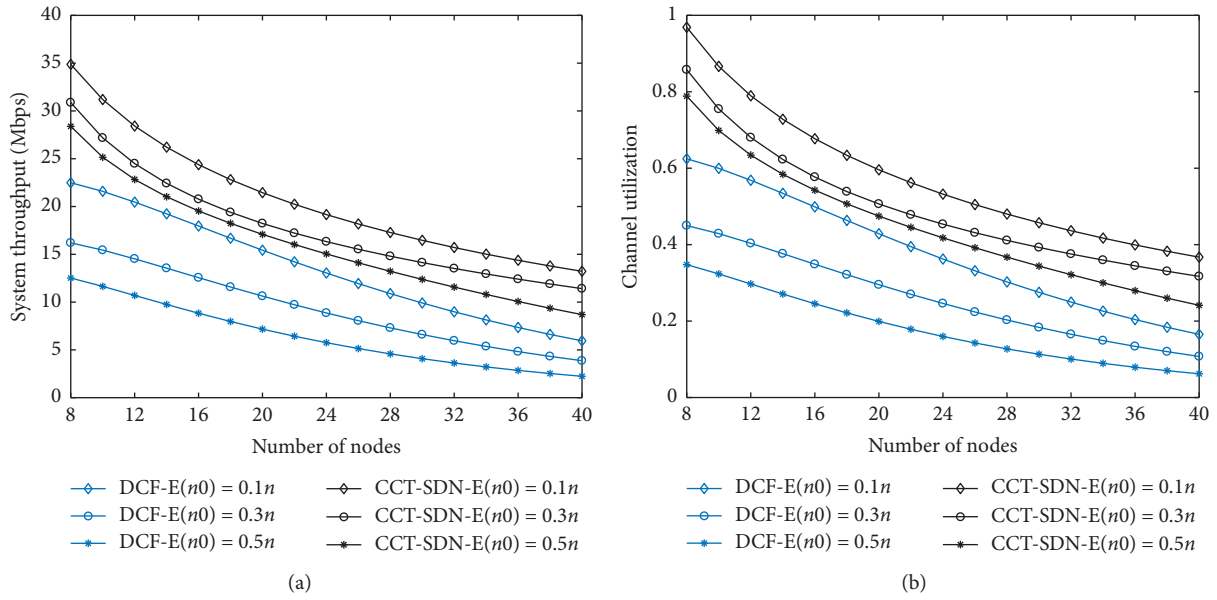


Figure 6: System throughput and channel utilization versus the number of nodes in the network. (a) System throughput. (b) Channel utilization.

Table 1: Experimental parameters.

Parameters	Value	Parameters	Value
Simulated time	100 seconds	Slot time	9 us
Frequency channel	2.4 GHz	DIFS	34 us
Data rate	6–54 Mbps	SIFS	16 us
Propagation loss model	Log distance	Propagation delay	1 us
Tx power	16 dbm	W_0	16
MAC header	224 bits	W_m	1024
PHY header	192 bits	ACK	112 bits

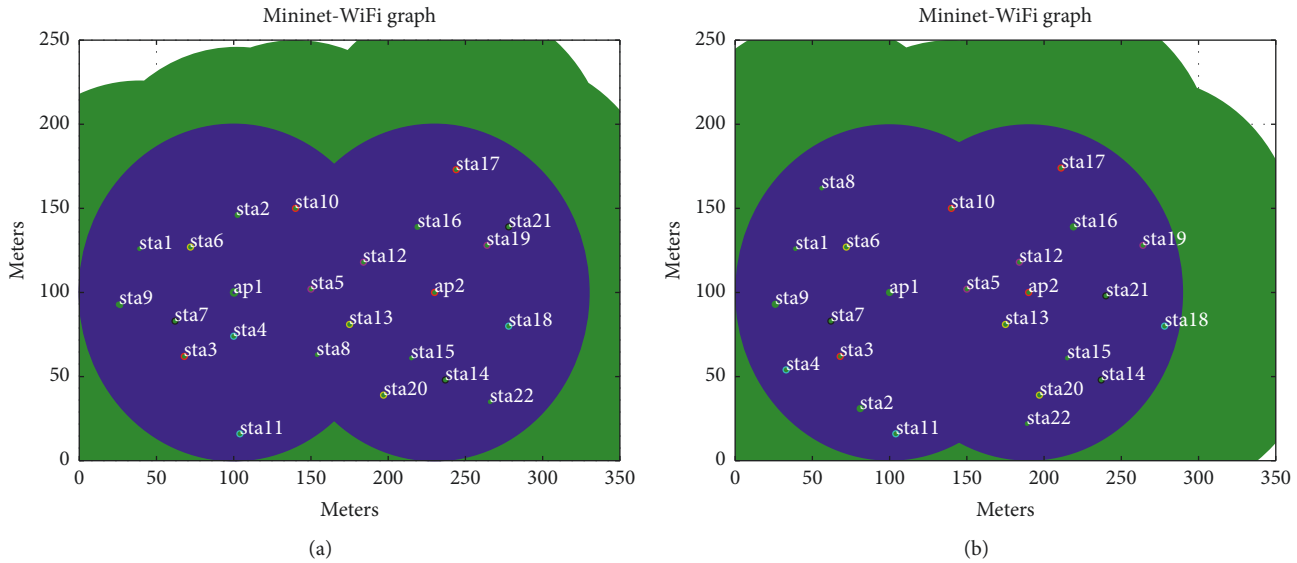
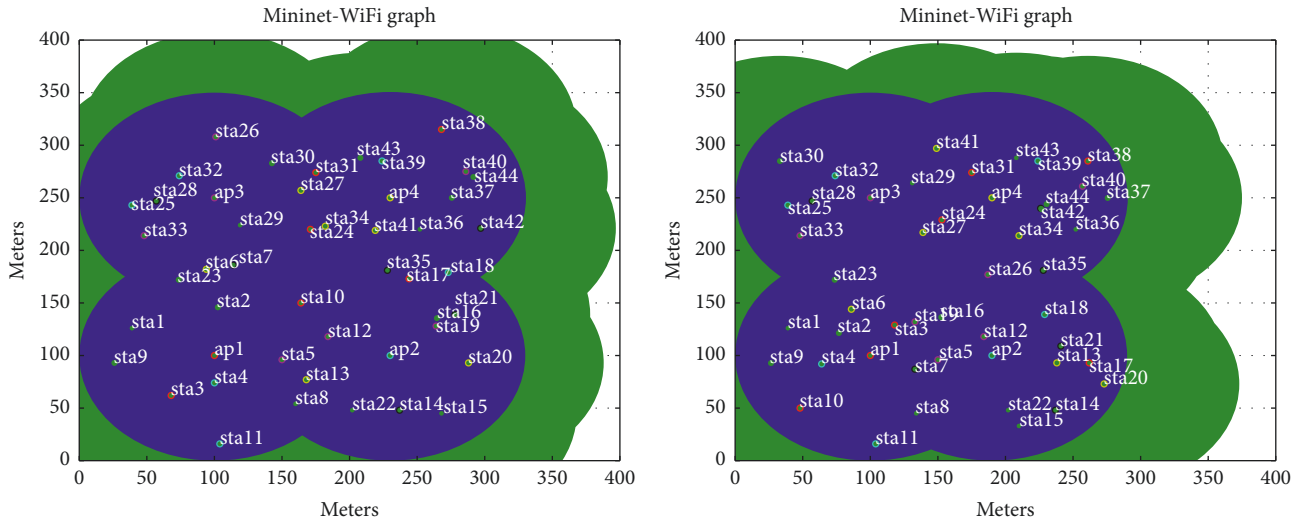
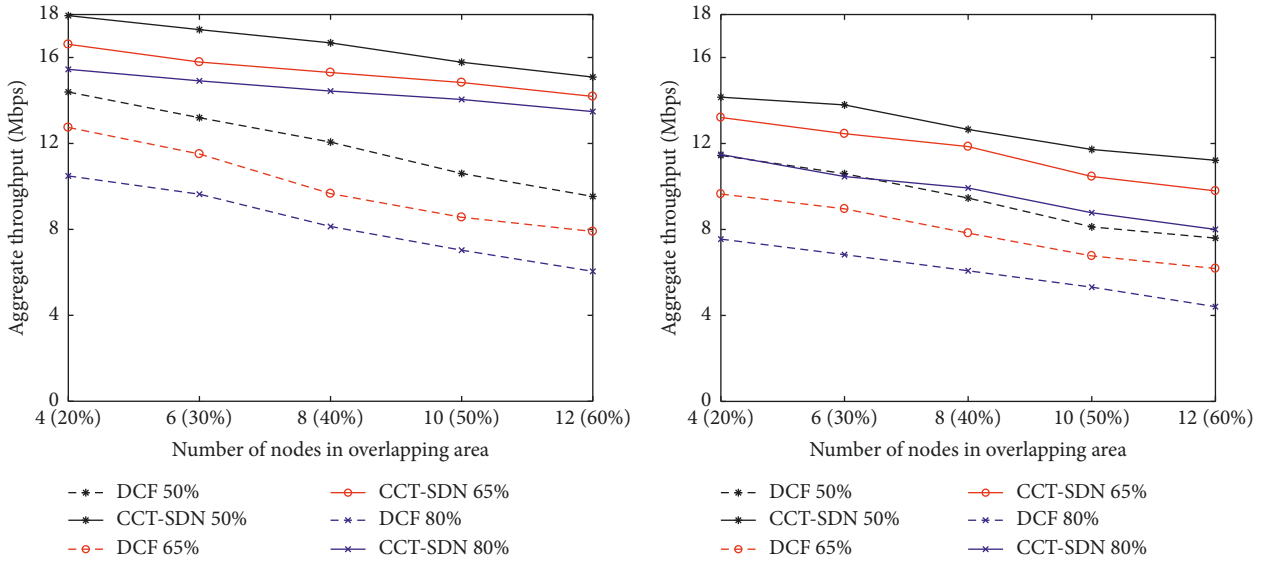


Figure 7: Continued.



(c) (d)

Figure 7: Network topology with 2 or 4 APs deployed in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario, (b) 2 APs in tight coupling scenario, (c) 4 APs in loose coupling scenario, and (d) 4 APs in tight coupling scenario.



(a) (b)

Figure 8: Continued.

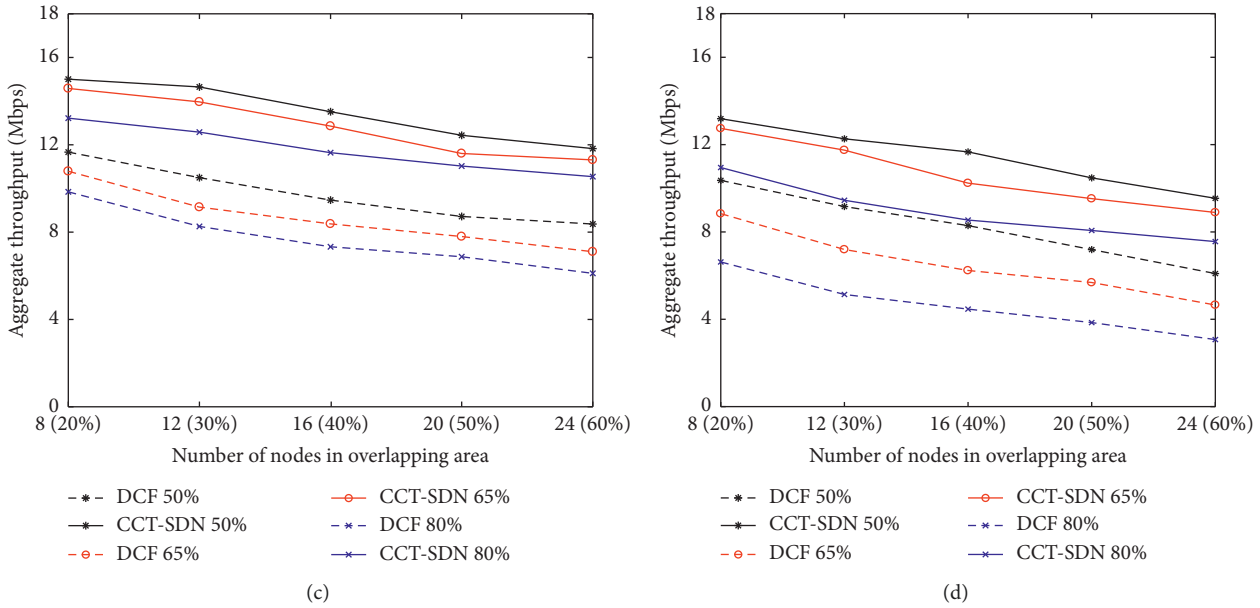


Figure 8: Aggregate throughput versus number of STAs in overlapping area for 2 and 4 APs in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario, (b) 2 APs in tight coupling scenario, (c) 4 APs in loose coupling scenario, and (d) 4 APs in tight coupling scenario.

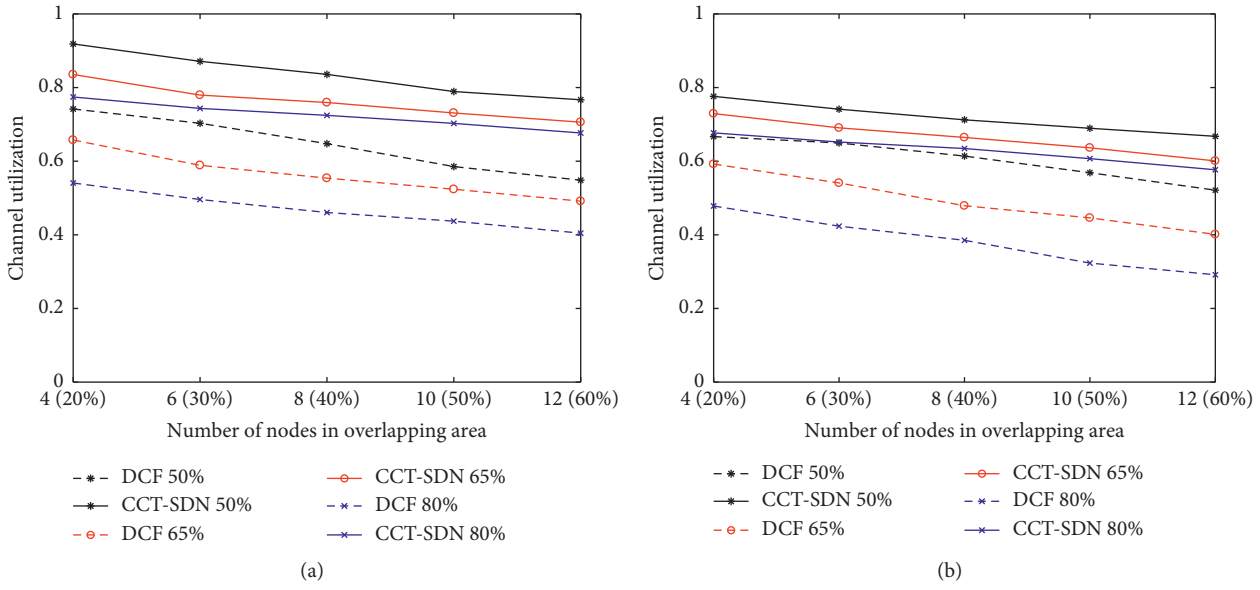


Figure 9: Continued.

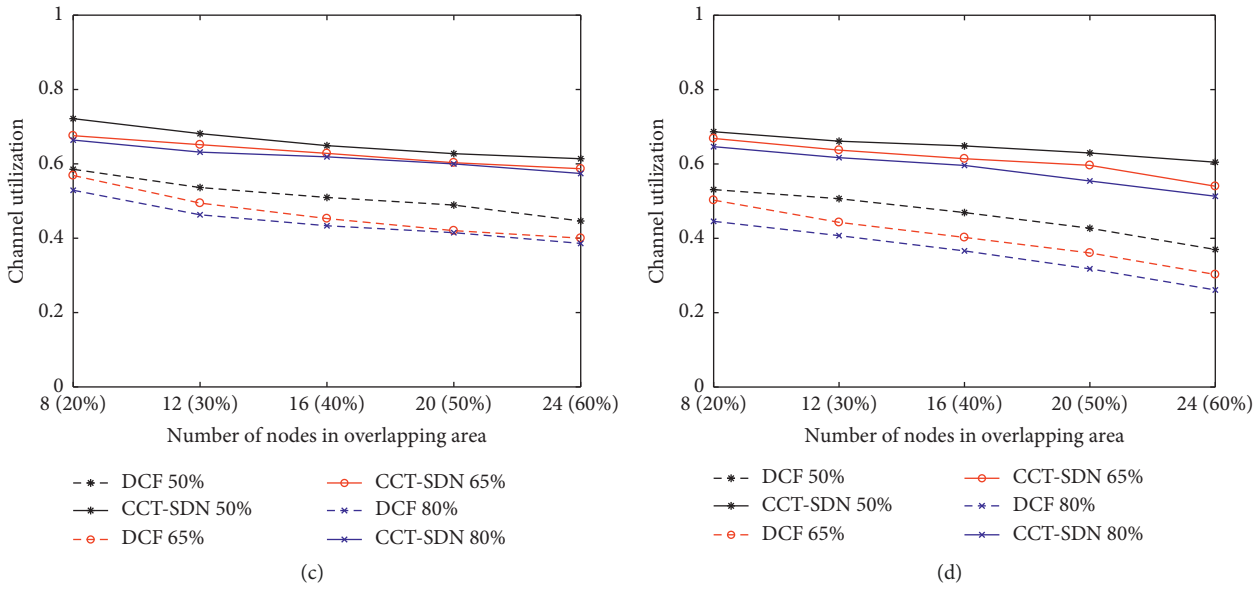


Figure 9: Channel utilization versus number of STAs in overlapping area for 2 and 4 APs in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario, (b) 2 APs in tight coupling scenario, (c) 4 APs in loose coupling scenario, and (d) 4 APs in tight coupling scenario.

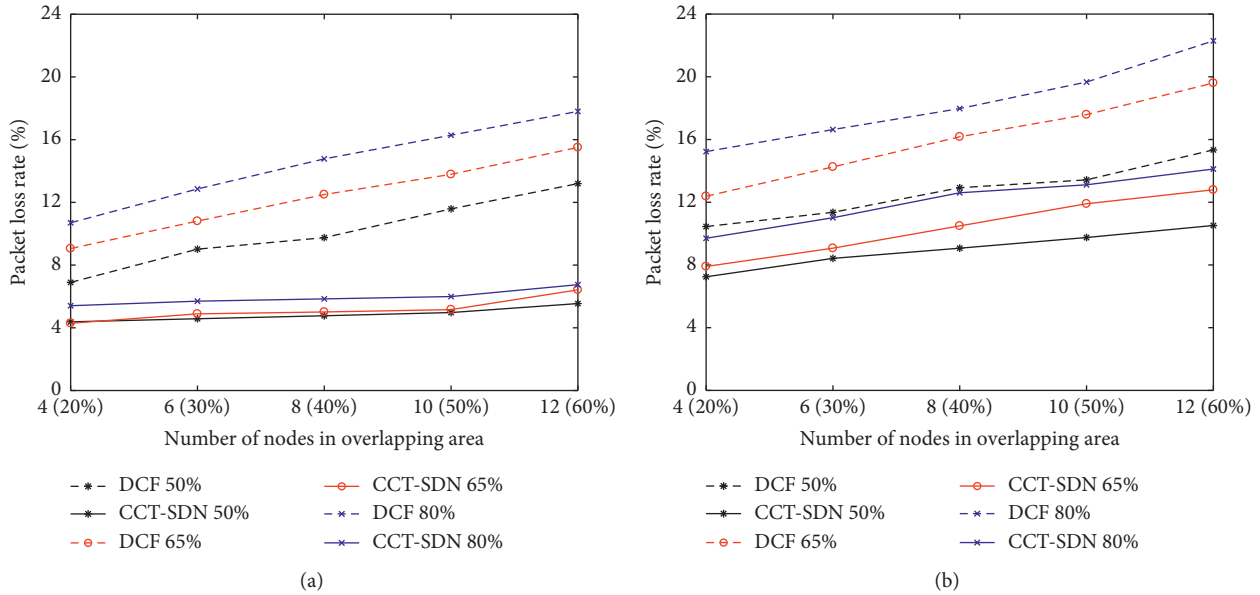


Figure 10: Continued.

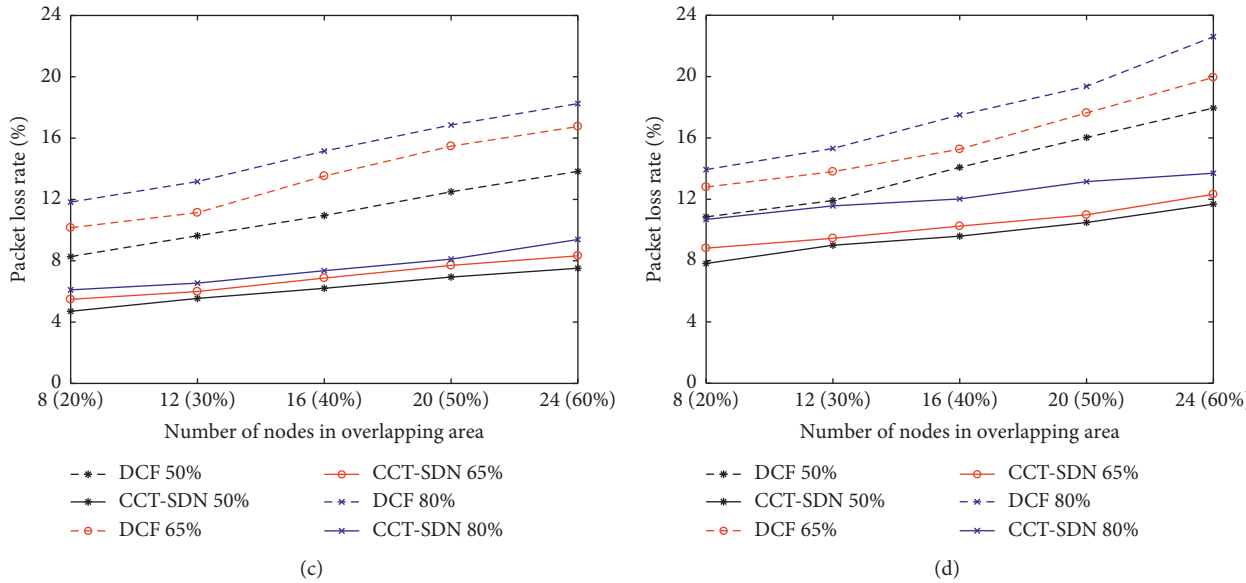


Figure 10: Packet loss rate versus number of STAs in overlapping area for 2 and 4 APs in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario, (b) 2 APs in tight coupling scenario, (c) 4 APs in loose coupling scenario, and (d) 4 APs in tight coupling scenario.

The result indicates that the aggregate throughput decreases with the increasing of number of STAs in overlapping areas, which is due to the deterioration of cochannel interference. Nevertheless, CCT-SDN succeeds in providing higher aggregate throughput for all BSSs regardless of their overlapping degree (denote different number of STAs in overlapping areas). Specifically, for 2 APs scenario shown in Figures 8(a) and 8(b), the aggregate throughput gains of CCT-SDN algorithm compared with default DCF algorithm range approximately from 24.7% to 58.3% in loose coupling scenario and from 23.6% to 47.7% in tight coupling scenario when the downlink traffic ratio is 50%. This phenomenon can also be observed when the number of APs gets larger. More APs and STAs can also cause more contentions and collisions, which may slightly decrease the throughput when the number of APs is 4, as shown in Figures 8(c) and 8(d). However, the CCT-SDN also can improve the probability of successful transmission by enabling concurrent transmission, especially in high-density areas with more exposed terminal existing. In contrast, the throughput of legacy DCF is degraded severely due to a large number of unnecessary back-offs occurring. Therefore, it can be concluded that the CCT-SDN can improve significantly the performance especially in high-density scenario.

Figure 9 illustrates the channel utilization for 2 and 4 APs in loose and tight coupling scenarios. As shown, CCT-SDN algorithm also achieves higher channel utilization than DCF in loose and tight coupling scenarios and in different ratio downlink traffic scenarios. Though the channel utilization of CCT-SDN also decreases with the increase of number of STAs in overlapping areas, it still achieves about 28% (2 APs scenario) and 23% (4 APs scenario) performance improvement over DCF when the percentage of STAs in overlapping areas is 60% in tight coupling scenario. Reasonably, the CCT-SDN enables lots of concurrent transmissions and also avoids the downlink transmission collision by addressing the exposed/hidden terminal problem, thus utilizing the channel more efficiently.

It is well known that the packet loss is mainly caused by transmission collisions and the network congestion. More STAs in multiple APs coverage overlapping areas may result in the higher cochannel interference, which can cause more packet losses. Hence, we also conduct some simulations to validate that CCT-SDN performs a better behavior by considering comprehensively these factors, which are shown in Figure 10. For 2 APs experiment, the packet loss rate by using CCT-SDN can be reduced by 59.9% over DCF under tight coupling scenario when the percentage of STAs in overlapping areas is 60%. Similar results can also be obtained in 4 APs experiment. The main reason is that the CCT-SDN enables downlink concurrent transmissions and reduces reasonable back-off time to relieve the congestion at AP; meanwhile, it also prevents some potential collision transmission from APs, thus effectively decreasing the packet loss rate.

In what follows, we study the effect of different network density and assess the effectiveness of CCT-SDN algorithm under 2 APs operating in different coupling scenario as well.

5.2.2. Different Network Density Scenarios. Based on the previous experiments, we further analyze how the performance of CCT-SDN varies when the number of STAs associated with APs changes. For this purpose, we consider the case of 2 APs, with a fixed number of STAs in overlapping areas (30%). We set the number of STAs in the whole network from 10 to 70 (5 to 35 STAs associated with each AP) and compare the CCT-SDN with DCF and RTS/CTS algorithm.

Figure 11 plots the average aggregate throughput for each AP versus the number of its associating STAs varying

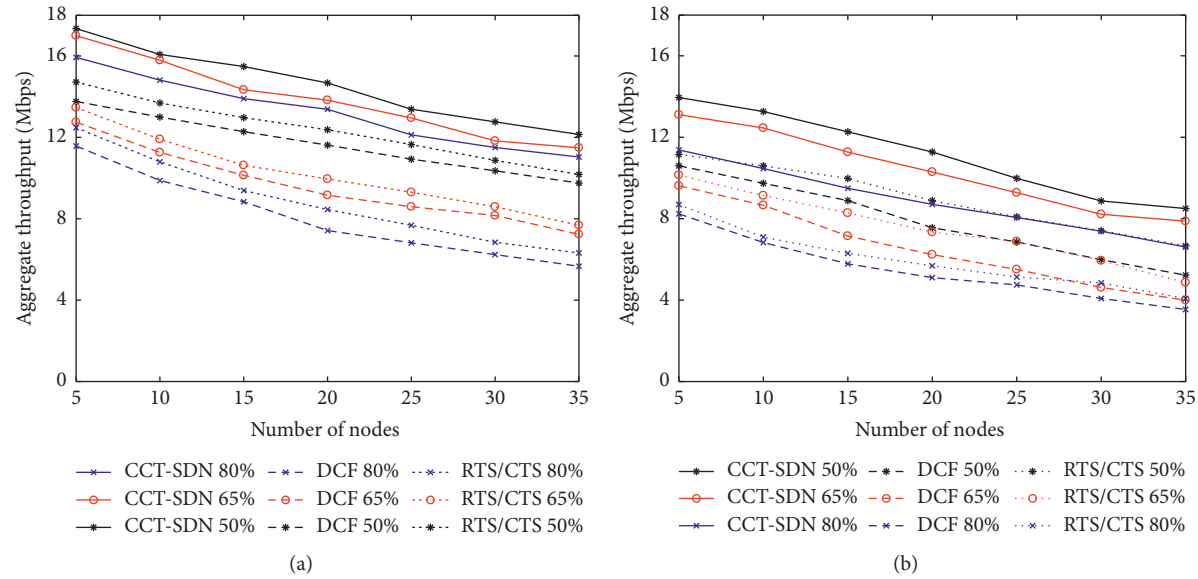


Figure 11: Aggregate throughput versus different network density in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario; (b) 2 APs in tight coupling scenario.

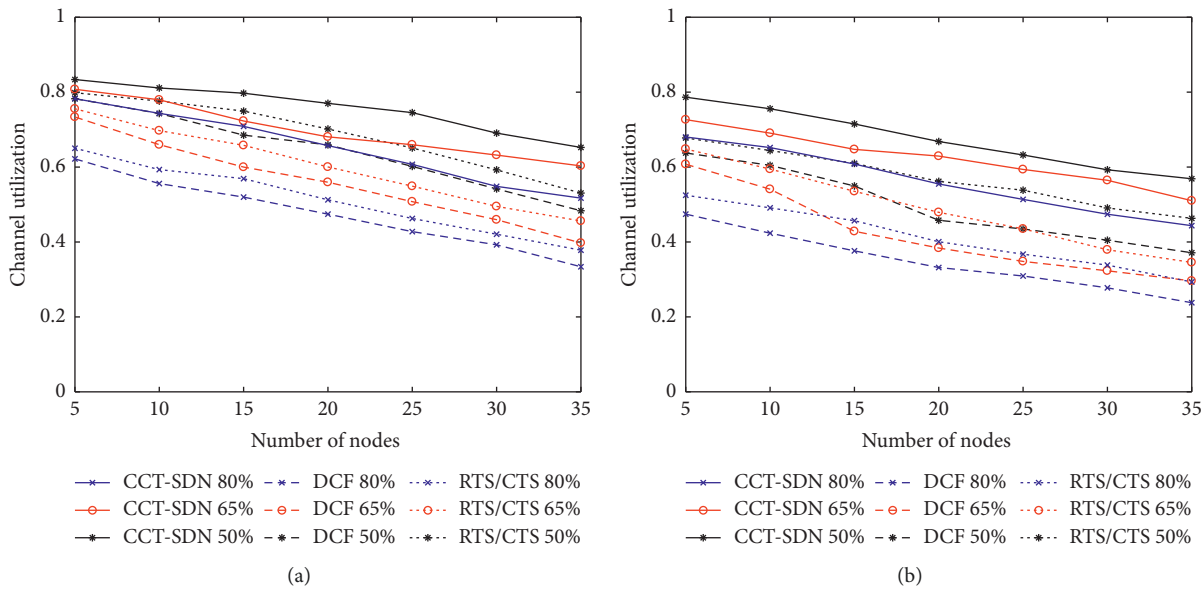


Figure 12: Channel utilization versus different network density in loose coupling and tight coupling scenarios: (a) 2 APs in loose coupling scenario; (b) 2 APs in tight coupling scenario.

from 5 to 35 for CCT-SDN, DCF, and RTS/CTS algorithms. From this figure, it can be found that the throughputs of all schemes decrease as the number of STAs increases. However, the CCT-SDN always performs better in both loose coupling and tight coupling scenarios. For example, when the percentage of clients in overlapping areas is increased to 35% in tight coupling scenarios, the CCT-SDN achieves the performance improvement about 26% over DCF and 20% over RTS/CTS. The reason is that more STAs in overlapping area can cause more interference and contentions, which results in degrading the throughput for three schemes. Nonetheless, CCT-SDN relieves the collisions by tackling exposed/hidden terminal problems simultaneously, and RTS/CTS scheme only focuses on the hidden terminal problem and brings extra transmission overhead.

Figure 12 repeats Figure 11, except that the y-axis represents the channel utilization. Similar results can be reserved and the CCT-SDN can obtain 53% and 23% improvement over DCF and RTS/CTS, respectively, at 35 STAs in tight coupling network. The results are also reasonable that more concurrent transmissions can be operated by the CCT-SDN when more exposed terminals are existing in

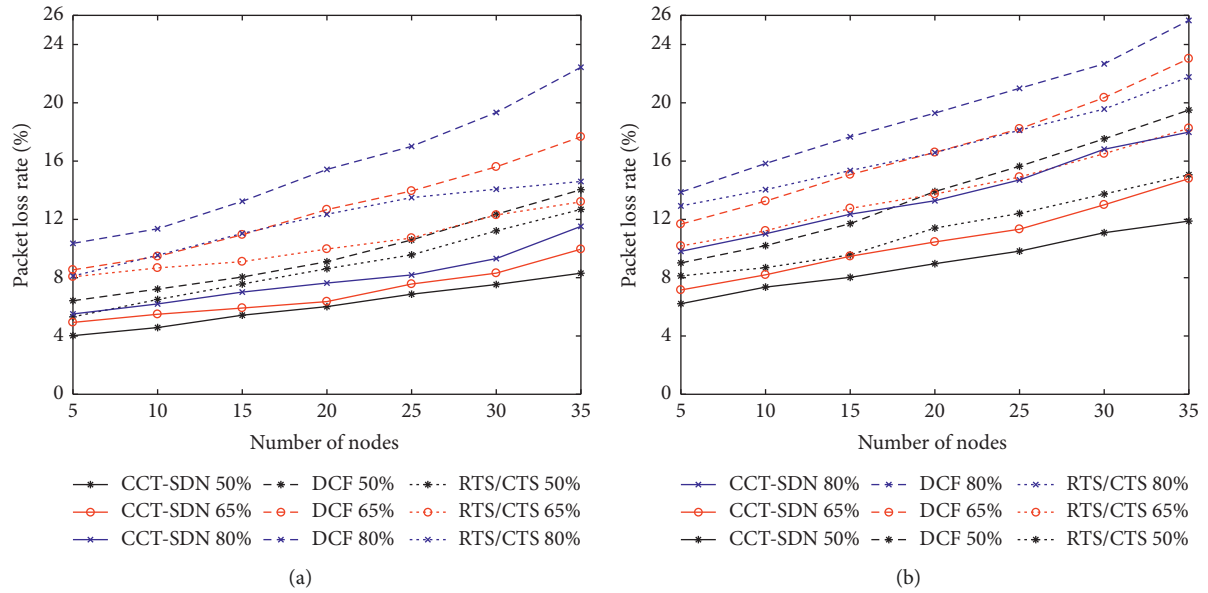


Figure 13: Packet loss rate versus different network density in loose coupling and tight coupling scenarios. (a) 2 APs in loose coupling scenario; (b) 2 APs in tight coupling scenario.

network. However, the legacy DCF and RTS/CTS schemes cannot utilize effectively channel resources due to lots of unnecessary back-off times, especially in severe cochannel interference scenarios.

At last, shown similarly in Figure 13, the CCT-SDN can outperform the DCF and RTS/CTS in packet loss rate calculated by each AP. Likely, when number of STAs is 35, the packet loss rate of CCT-SDN reduces by 20% over RTS/CTS and 40% over DCF in loose coupling scenario, and the performance improvement for packet loss rate ranges from 17% to 24% for RTS/CTS even in the case of 80% downlink traffic tight coupling scenario. Meanwhile, CCT-SDN also can relieve the link congestion due to its avoidance for unnecessary channel access back-off and improvement of concurrent transmission capability, especially in dense and heavy load network.

6. Conclusions

Legacy 802.11 DCF mechanisms use a simple distributed mechanism to coordinate the channel access and data transmission among all nodes. However, with the various multimedia applications being deployed and the network density increasing, the cochannel interference caused by the hidden/exposed terminal problems deteriorates severely the network performance, which obstructs the improvement of the system throughput and is very hard to be tackled by this purely distributed control architecture. In this paper, borrowing programmable idea and the function separation of SDN, we propose the CCT-SDN mechanism to address these problems. Our mechanism employs a centralized SDWN architecture to control comprehensively the transmission behavior of APs via evaluating the cochannel interference and is able to perform a collision-free downlink schedule for all APs. The CCT-SDN is simple and there is no requirement for any modifications for existing STAs, and it can be deployed flexibly in any scale applications. Meanwhile, we introduce a SPRIM algorithm to enhance the retrieving efficiency of the controller, which enables CCT-SDN to perform a real-time concurrent transmission control. In addition, we also establish a theoretical model to analyze the effectiveness of SPRIM algorithm and CCT-SDN mechanism. The extensive simulations clearly demonstrate the significant improvement of the concurrent transmission scheme over its legacy counterpart. The CCT-SDN could be considered as a viable replacement of current WiFi architecture and provides novel ideas for designing multiple APs deployment in some QoS-based applications. For our future work, we also plan to assess and implement CCT-SDN in a big-scale real-life test-bed.

Data Availability

The data used to support the findings of this study are available upon request to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Science Foundation of China (61602073) and Chongqing Postgraduate Research and Innovation Project (CYS19260).

References

[1] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined

wireless networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[2] L. B. Jiang and S. C. Liew, "Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 34–49, 2007.

[3] X. Yang, "IEEE 802.11N: enhancements for higher throughput in wireless LANs," *IEEE Wireless Communications*, vol. 12, no. 6, pp. 82–91, 2005.

[4] IEEE, *IEEE P802.11 Wireless LANs Usage Models*, IEEE 802.11-03/802r17, 2003.

[5] W. Wang, Y. Chen, Q. Zhang, K. Wu, and J. Zhang, "Less transmissions, more throughput: bringing carpool to public WLANs," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1168–1181, 2016.

[6] X. He, S. Lakshmanan, R. Sivakumar, and F. Y. Li, "C2SMA/CA: enabling co-channel concurrency in WLANs using positional information," in *Proceedings of the 2013 IEEE wireless communications and networking conference (WCNC '13)*, pp. 848–853, Las Vegas, NV, USA, March 2013.

[7] B. Alawieh, Y. Zhang, C. Assi, and H. Mouftah, "Improving spatial reuse in multihop wireless networks-a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 71–91, 2009.

[8] M. Rick, "A safe, efficient update protocol for openflow networks," in *Proceedings of the Workshop on Hot Topics in Software Defined Networks, HotSDN '12*, pp. 61–66, Helsinki, Finland, August 2012.

[9] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *Proceedings of the 5th Usenix Symposium on Networked Systems Design & Implementation, NSDI '08*, pp. 59–72, San Francisco, CA, USA, March 2008.

[10] J. Yao, T. Xiong, J. Zhang, and W. Lou, "On eliminating the exposed terminal problem using signature detection," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2034–2047, 2016.

[11] J. Yao, W. Lou, C. Yang et al., "Efficient interference-aware power control in wireless ad hoc networks," in *Proceedings of the 2017 IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[12] H. Zhang, X. Liu, C. li et al., "Scheduling with Predictable link reliability for wireless networked control," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6135–6150, 2017.

[13] Z. Jia, X. He, and F. Y. Li, "Enable concurrent transmissions with beamforming for broadband wireless access in CSMA/CA-based WLANs," in *Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps '14)*, pp. 1075–1080, Austin, TX, USA, December 2014.

[14] O. Oteri, F. La Sita, R. Yang, M. Ghosh, and R. Olesen, "Improved spatial reuse for dense 802.11 WLANs," in *Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps' 15)*, pp. 1–6, San Diego, CA, USA, December 2015.

[15] K. Shin, I. Park, J. Hong, D. Har, and D.-h. Cho, "Per-node throughput enhancement in Wi-Fi densenets," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 118–125, 2015.

[16] I. Jamil, L. Cariou, and J. Helard, "Improving the capacity of future IEEE 802.11 high efficiency WLANs," in *Proceedings of the 21st International Conference on Telecommunications (ICT)*, pp. 303–307, Lisbon, Portugal, May 2014.

[17] I. Jamil, L. Cariou, and J. Helard, "Efficient MAC protocols optimization for future high density WLANs," in *Proceedings of the 2015 IEEE Wireless Communications & Networking Conference*, pp. 1054–1059, Las Vegas, NV, USA, March 2015.

[18] X. H. Liu, Y. Chen, and H. Zhang, "A maximal concurrency and low latency distributed scheduling protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, no. 8, 2015.

[19] X. J. Zhao, S. G. Chen, J. Z. Guo et al., "Concurrent transmission mechanism to mitigate pan-exposed-node problems in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, 2017.

[20] K. T. Foerster, S. Schmid, and S. Vissicchio, "Survey of consistent software-defined network updates," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1435–1416, 2018.

[21] F. A. P. De Figueiredo, X. Jiao, W. Liu, and I. Moerman, "Radio hardware virtualization for software-defined wireless networks," *Wireless Personal Communications*, vol. 100, no. 1, pp. 113–126, 2018.

[22] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *Proceedings of the 2013 ACM Conference on SIGCOMM (SIGCOMM '13)*, pp. 387–398, Hong Kong, China, August 2013.

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

[24] C. Lin, W. Tsai, M. Tsai, and Y. Cai, "Adaptive load-balancing scheme through wireless SDN-based association control," in *Proceedings of the 31st International Conference on Advanced Information Networking and Applications (AINA '17)*, pp. 546–553, Taipei, Taiwan, March 2017.

[25] Y. D. Lin, C. C. Wang, Yi-Jen Lu, Y.-C. Lai, and H.-C. Yang, "Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks," *Wireless Networks*, vol. 24, no. 8, pp. 2711–2823, 2017.

[26] L. Suresh and S. Z. Julius, "Towards programmable enterprise WLANS with Odin," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks(HotSDN '12)*, pp. 115–120, New York, NY, USA, August 2012.

[27] S. Z. Julius and C. Mayer, "Open SDWN: programmatic control over home and enterprise WiFi," in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research Article (SOSR '15)*, pp. 1–12, Santa Clara, CA, USA, July 2015.

[28] C. Xu, W. Jin, G. Zhao, H. Tianfield, S. Yu, and Y. Qu, "A novel multipath-transmission supported software defined wireless network architecture," *IEEE Access*, vol. 5, no. 99, pp. 2111–2125, 2017.

[29] B. Cao, Y. Li, C. Wang, G. Feng, S. Qin, and Y. Zhou, "Resource allocation in software defined wireless networks," *IEEE Network*, vol. 31, no. 1, pp. 44–51, 2017.

[30] A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow switching: data plane performance," in *Proceedings of the 2010 IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.

[31] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: a survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[32] V. Shrivastava, S. Rayanchu, S. Banerjee, and K. Papagiannaki, "PIE in the sky: online passive interference estimation for enterprise WLANs," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation (NSDI '11)*, pp. 337–350, Boston, MA, USA, March 2011.

[33] M. Karaca, S. Bastani, and B. Landfeldt, "Modifying backoff freezing mechanism to optimize dense IEEE 802.11 networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9470–9482, 2017.

[34] O. T. Arogundade, B. Sobowale, and A. T. Akinwale, "Prim algorithm approach to improving local access network in rural areas," *International Journal of Computer Theory and Engineering*, vol. 3, no. 3, pp. 413–417, 2011.

[35] K. Ramachandran, R. Kokku, H. Honghai Zhang, and M. Gruteser, "Symphony: synchronous two-phase rate and power control in 802.11 WLANs," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1289–1302, 2010.

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

[37] Y.-W. Kuo and J.-H. Huang, "A CSMA-based MAC protocol for WLANs with automatic synchronization capability to provide hard quality of service guarantees," *Computer Networks*, vol. 127, no. 9, pp. 31–42, 2017.

[38] R. D. R. Fontes and C. E. Rothenberg, "Mininet-WiFi: a platform for hybrid physical-virtual software-defined wireless networking research," in *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*, pp. 607-608, Florianopolis, Brazil, August 2016.