Performance of Multi-Channel IEEE 802.11 WLANs with Bidirectional Flow Control - Extended Report*

Suong H. Nguyen, Lachlan L. H. Andrew, Hai L. Vu
Centre for Advanced Internet Architectures, Technical Report 130730B
Swinburne University of Technology
Melbourne, Australia
hsnguyen@swin.edu.au, landrew@swin.edu.au, hvu@swin.edu.au

***Abstract*—To increase throughput, wireless LANs often combine multiple frequency channels. We investigate three options for using two channels: using the wider channel to increase the bitrate, sharing data over two independent bidirectional channels, and allocating one channel to the uplink and one to the downlink. To enable a realistic evaluation of bidirectional traffic, we incorporate a simple model of TCP flow control. We develop a tractable model of flow control in IEEE 802.11 networks with both upload and download traffic, which we then use to compare the three channel sharing options. The use of a wider combined channel, which is the default in IEEE 801.11n, provides the lowest throughput, and the use of independent bidirectional channels provides the highest.**

I. Introduction

Due to the prevalence of laptops and smart phones with built-in WiFi [1], wireless local area networks (WLANs) have become a popular means of Internet access. This places increased demand on the available capacity, and so engineers are continually seeking ways to increase the throughput while maintaining compatibility with existing standards.

One of the features in the current IEEE 802.11 standard increases throughput by allowing stations to use double the physical bandwidth when possible [2]. This allows the data rate to be doubled, which helps to increase network throughput. However, there are several techniques for using two channels concurrently while still keeping the standard medium access control (MAC) protocol. We investigate the performance of these options, which will allow designers to choose the right technique for a particular scenario. We focus on the case of two channels, but expect both the techniques and the qualitative conclusions to apply more generally.

There has been substantial research on how to use multiple channels concurrently in an effective way in IEEE 802.11 wireless networks. Many of those propose multi-channel MAC protocols for wireless ad hoc networks, which consider single-transceiver stations [3], [4], [5], [6] or multi-transceiver stations [7], [8], [9]. Unlike those, our report does not aim to propose a new multi-channel MAC protocol but studies the utilization of multiple channels in a single hop infrastructure WLANs, where an access point (AP) is a centralized controller to inform stations of the working mode and schedule traffic over channels. Then, stations still use the standard IEEE 802.11 MAC protocol to gain channel access.

Most of this research has focused on the MAC layer in isolation, and has evaluated the performance with simplistic traffic patterns such as Poisson packet arrivals or saturated stations. However, most Internet applications use sliding window flow control, notably the transmission control protocol (TCP), to provide reliability. In this work, we introduce a simple model of flow control into 802.11 models. Its novelty is the ability to model networks with both upload and download stations.

There is a considerable body of literature studying sophisticated models of TCP's congestion control operating over simple models of wireless networks [10], [11], [12], [13], [14]. These studies mainly characterize TCP evolution either at the packet-based level or at the macroscopic level and evaluate the TCP throughput in the presence of congestion and loss caused by bottleneck links and noisy channels [15]. These models are useful to understand the impact of the network and TCP parameters on the TCP performance; however, they do not describe the interplay between TCP and the MAC protocol.

*This report is an extended version of "Performance of Multi-Channel IEEE 802.11 WLANs with Bidirectional Flow Control" accepted as a short paper in IEEE LCN 2013, October 2013.

There has also been considerable work focusing on modifying the standard MAC models to capture the interaction with simple models of TCP, most notably its window flow control [16], [17], [15], [18], [19], [20], [21], [22] The standard assumptions in these models include there being no loss due to buffer overflow and TCP timeout, an ideal physical channel, long-lived flows so that the system reaches an equilibrium, and a small RTT. This means that they only consider the flow control mechanism of TCP, not its congestion control. Our model is of this second type.

A common approach used in many of these latter models is first to determine the average number of backlogged stations in the network at any time and then to apply the saturated MAC model with this number of backlogged stations. These models differ in the method to estimate the number of backlogged stations, which can be classified into Markov chain-based ones [16], [17], [15], [18], [19] or non Markov chain-based ones [20], [21], [22]. Compared with non Markov chain models, Markov chain ones are much more complex due to the need of solving a Markov chain with large number of states.

Among non Markov chain models, that of Sakurai and Hanly [21] is simple but very accurate; however, it only models a network with either TCP upload flows or TCP download flows. It works by determining the probability a station will transmit in a given time slot (the "attempt probability") from that of the AP. Using a similar idea, we propose a tractable non Markov chain model of IEEE 802.11 WLANs with both upload and download flows which does not require significant changes to standard models of the IEEE 802.11 MAC. We show that our model is very accurate under a wide range of scenarios, and identify some conditions where the model's assumptions do not hold. For tractability, our model also uses the above standard assumptions as in most prior work; its major contribution is the extension to bidirectional traffic.

Having derived this new model, we then apply it to our study of the performance of three natural modes to utilize two channels concurrently. Unlike [23], we do not consider the changes in physical layer performance when we change channel width, and only consider the MAC performance. The first scheme we consider is called "channel bonding" in the IEEE 802.11n standard and 40/40 in [24], and involves the use of one MAC instance (i.e. independent queue and backoff/contention process) on a channel that has twice the physical modulation rate and so is twice the bandwidth of a standard slot. In this scheme, overheads such as backoff require the full wide channel to be idle, and we find that it consequently performs worst in many scenarios. The second scheme we consider uses a technique more similar to channel bonding in wired networks, in which there are two separate physical channels, each with its own MAC instance, over which traffic is split. In this case, while one channel is backing off, the other channel(s) can be in use. As may be expected from the results of [25], we find that this gives higher throughput. The third scheme we consider separates the uplink transmissions from the downlink transmissions, called 20+20 in [24]. (Note that "downlink" transmissions are transmissions by the AP and "uplink" transmissions are transmissions by the stations; these are not to be confused with "down*load*" transmissions, which are all transmissions associated with flows in which data packets are sent on the downlink and TCP ACKs are sent on the uplink, and "up*load*" transmissions in which data packets are sent on the uplink and TCP ACKs are sent on the downlink.) Although this has the capacity to eliminate collisions on the downlink, on which only the AP transmits, its performance turns out to be worse than that of the two bidirectional channels.

The rest of the report is organized as follows. First, a brief description of IEEE 802.11 protocol and TCP is provided in Section II. Then, three different modes to use two channels concurrently are described in Section III. In Section IV, we present a model of IEEE 802.11 infrastructure WLANs with TCP upload and download flows to study the performance of these three modes. Section V provides numerical results. Finally, we offer concluding remarks in Section VI.

II. Background

Here we will first briefly describe the Distributed Coordination Function (DCF) channel access mechanism defined in the IEEE 802.11 standard [2]. Then, we will provide some background about TCP.

A. 802.11 DCF

The DCF channel access mechanism enables users to contend for the common wireless channel using a carrier sense multiple access mechanism with collision avoidance (CSMA/CA). To reduce collisions, it employs both sensing of the channel to detect channel activity and truncated binary exponential backoff (BEB) to randomize the start times of packet transmissions. When a packet arrives to an idle source, the source senses the channel for a period DIFS. If it is idle during this whole time, the packet is transmitted immediately. Otherwise, the source waits until the channel is continuously idle for DIFS, and then starts a backoff process. A backoff counter is initialized to a random integer uniformly distributed between 0 and (CW-1), where CW is the current contention window. For each new transmission, CW is initialized to $\mathrm{CW}_{\min}$ and doubles after each unsuccessful transmission until it reaches $\mathrm{CW}_{\max}$, after which it remains constant until the packet is either successfully received or a retry limit is exceeded. The backoff counter is decreased by one at every idle slot time and frozen during periods of channel activity. Decrementing is resumed after the expiration of a DIFS after a channel activity period ends. When the backoff counter reaches zero, the frame is transmitted. An acknowledgment (ACK) is sent back from the receiver after a Short Inter-Frame Space (SIFS) for every successful frame reception. If an ACK is not received, the source increases CW as described above, and attempts again until the retry limit is reached. After receiving an ACK, the source performs a "post-backoff" process with CW set to $\mathrm{CW}_{\min}$ before being allowed to restart the above procedure. This prevents back-to-back frame transmission.

B. TCP

TCP is a reliable window flow control protocol at the transport layer. Being a window flow control protocol means that it maintains a window size w that defines a strict upper bound on the amount of unacknowledged data that can be in transit between a given sender-receiver pair. Unlike many window flow control schemes, TCP allows "delayed acknowledgements" [26], in which a single acknowledgement packet can acknowledge multiple data packets. In wired networks, this provides limited benefit because acknowledgement packets are quite small. However, in 802.11 networks it provides a considerable benefit, because there is a substantial fixed cost for sending a packet, however small it is. TCP's window size is governed by the window size advertised by the receiver, $awnd$, used by flow control mechanism and the congestion window, $cwnd$, used by congestion control mechanism. TCP's flow control prevents the sender from overrunning the receiver's buffer by implementing the policy that allows the sender to send new packets only after receiving the acknowledgement for the previous packet. Moreover, the congestion control implements algorithms such as slow start and congestion avoidance to prevent the TCP senders from overrunning the resources of the network.

In many older operating systems, the default $awnd$ is not large (for example, for Microsoft Windows 95/98/NT it was 8760 bytes and for Microsoft Windows 2000, it was 17520 bytes); therefore, the sender was never be able to fill up the network pipe [21]. Although modern operating systems have substantially larger values of $awnd$, network speeds are also increasing, and so there will remain many cases in which $awnd$ limits dominates $cwnd$. Besides, a sufficiently small $awnd$ can prevent buffer overflow at the network bottleneck. (See for example [27]). In this case, the equilibrium state can be characterized by a window size fixed at $awnd$. Our model in the next section considers this case.

III. DIFFERENT MODES OF UTILIZING TWO CHANNELS

Given that two channels can be used concurrently as defined in the current IEEE 802.11 standard [2], there are several ways to utilize this feature. In this report, we are interested in evaluating three natural modes which are described as follows.

A. Mode 0

In this mode, two adjacent channels are coupled to form a single channel with double the bandwidth to be shared between upload and download TCP traffic. As a result, PHY data rate is doubled; however, this mode may have high collision overhead due to all traffic sharing the same channel which causes bottleneck at the AP. The advantage of this mode is that it requires only one transceiver per station. Besides, this mode is currently the default one in the IEEE 802.11 standard [2].

B. Mode 1

This mode involves balancing upload and download traffic over two channels by splitting both types of traffic equally on two channels. This can help to reduce collision on each channel and to utilize the channel more efficiently due to reduction in the cost of protocol overhead. However, the disadvantage of this mode is that it requires two transceivers per station.

C. Mode 2

In this mode, all uplink traffic is sent on one channel (i.e. Channel 1) and all downlink traffic is sent on the other channel (i.e. Channel 2). This means that all packets sent from wireless stations (not the AP) are sent on Channel 1 while all packets sent from the AP are sent on Channel 2. This mode helps to solve the issue of bottleneck at the AP mentioned above, which may improve network performance. As with Mode 1, two radio front ends are required per station.

To study and compare the performance of these three modes, we build a model of IEEE 802.11 WLANs with TCP traffic, which will be described in the next section.

IV. Model of 802.11 WLANs with upload and download TCP flows

In this section, we will propose a tractable model of IEEE 802.11 WLANs with $N_u \geq 1$ wireless stations (STA) uploading TCP traffic and $N_d \geq 1$ stations downloading TCP traffic through an access point (AP).

In the proposed model, we focus on only the flow control mechanism of TCP traffic, i.e, TCP self-clocking mechanism using TCP ACKs. This is the case for TCP when there is no packet loss due to buffer overflow and TCP time-outs and RTTs are negligible. Besides, the channel condition is ideal so that packet corruption is negligible. Moreover, data from the socket receive buffer of an application is read at the rate which it is received from the network. This means the maximum TCP receive window size is always advertised in TCP ACK packets.

A. Notations

Let S_u and S_d denote the total upload and download throughput, respectively, measured in packets per second. Similarly, S_u^a and S_d^a are the rates at which TCP ACK packets are sent in upload and download flows. Note that S_u^a flows on the downlink, and S_d^a flows on the uplink.

The backoff mechanism imposes a slotted structure on time, with slot sizes independently distributed as a random variable Y, which is σ if the slot is idle or longer if a transmission is attempted.

Let τ_u^{STA} and τ_d^{STA}, respectively, be the attempt probabilities of an upload STA and an download STA.

Let τ_u^{AP} and τ_d^{AP} be the probabilities an AP attempts to transmit TCP ACKs and data, respectively. Also denote the total attempt probability of the AP by τ^{AP}.

Let p_u^{STA} and p_d^{STA} be the collision probabilities of an upload STA and an download STA. Also denote the collision probability of the AP by p^{AP}.

Let $W_{\min}$ is the minimum contention window used by all stations and the AP.

Stations emit data packets of constant size L_{data} and TCP ACK packets of constant size of L_{ack}.

Each packet is attempted up to K times, with the jth attempt occurring after a backoff of $U_j \sim U[0, 2^{\min(j,m)} W_{\min} - 1]$ slots, where $W_{\min}$ is called the contention window and m is the maximum number of times a STA or the AP doubles its contention window.

Let $\mathbb{E}[.]$ denote the mean value of a random variable.

Denote the duration that a data packet and a TCP ACK packet occupy the channel by T_{data} and T_{ack}, respectively.

Note that in this report, we will use ACK to refer to an acknowledgment packet at the MAC layer and TCP ACK to refer to a TCP acknowledgment packet at the transport layer.

B. Model

Here we first describe a model for WLANs with uploading and downloading stations sharing the same channel. This model can be directly used to analyze two modes 0 and 1. In particular, in Mode 0, there is only one channel with double the bandwidth shared among all stations and the AP. Similarly, each channel in Mode 1 is also shared among stations and the AP; therefore, we can apply the model for each of two channels in Mode 1. Then, in Section IV-B1, we will show how this model can be modified to model Mode 2 where uplink traffic from wireless stations is sent on one channel and downlink traffic from the AP is sent on the other channel.

We model only the flow control component of TCP. Because the RTT is assumed to be small and packet loss is considered negligible, this simply enforces a relationship between the rate of transmission of data packets and acknowledgements. Specifically, delayed acknowledgements [26] require that one TCP ACK be sent for every D data packets. Hence

$$S_u^a = S_u/D \tag{1a}$$

$$S_d^a = S_d/D \tag{1b}$$

In this model, all data packets and/or all TCP acknowledgement packets flow through the AP transmit buffer. Therefore, we assume that the AP transmit buffer never empties. Then, the attempt probability of the AP can be determined as that of a saturated source derived in [28]:

$$\begin{aligned}\tau^{\text{AP}} =& 2(1-(p^{\text{AP}})^{K+1})/\\ &\Big[W_{\min}(1-(2p^{\text{AP}})^{m+1})(1-p^{\text{AP}})/(1-2p^{\text{AP}})\\ &+(2^m W_{\min}+1)(1-(p^{\text{AP}})^{K+1})\\ &-2^m W_{\min}(1-(p^{\text{AP}})^{m+1})\Big]\end{aligned} \tag{2}$$

At the AP, the transmission attempt probability for each type of packet (data or acknowledgement) is proportional to the throughput of that class, since the collisions are independent of the type of packet. Hence, by (1),

$$\tau_u^{\text{AP}} = \frac{S_u/D}{S_u/D + S_d}\tau^{\text{AP}} \tag{3a}$$

$$\tau_d^{\text{AP}} = \frac{S_d}{S_u/D + S_d}\tau^{\text{AP}}. \tag{3b}$$

Without delayed acknowledgements ($D = 1$), the number of data packets are equal to the number of ACK packets for each upload and download TCP flow.

Similar to [21], we assume that when an equilibrium state is reached, the combined effect of all upload stations is to yield a sequence of successfully transmitted TCP data packets with the average spacing (t_u^{cycle}) equal to ($1/D$) times that of the sequence of successfully transmitted TCP ACK packets at the AP. This means that there is one TCP data packet sent by a given upload station every period of $t_u^{cycle} N_u$. Then, the attempt probability of an upload station can be determined from the attempt probability τ_u^{AP} as follows. First,

$$\tau_u^{\text{STA}} = \frac{\mathbb{E}[\text{Number of attempts per data packet}]}{\mathbb{E}[\text{Number of slots per } (t_u^{cycle} N_u)]} = \frac{\mathbb{E}[\text{Number of attempts per data packet}]}{\left(N_u \frac{\mathbb{E}[\text{Number of attempts per TCP ACK packet}]}{D\tau_u^{\text{AP}}} \right)} \quad (4)$$

where the second line of the above expression comes from

$$\mathbb{E}[\text{Number of slots per } (t_u^{cycle} N_u)] = \frac{N_u}{D} \mathbb{E}[\text{Number of slots per } (t_u^{cycle} D)]$$

and

$$\tau_u^{\text{AP}} = \frac{\mathbb{E}[\text{Number of attempts per TCP ACK packet}]}{\mathbb{E}[\text{Number of slots per } (t_u^{cycle} D)]}.$$

Besides, we have

$$\begin{aligned} &\mathbb{E}[\text{Number of attempts per data packet}] \\ &= 1 + p_u^{\text{STA}} + (p_u^{\text{STA}})^2 + \cdots + (p_u^{\text{STA}})^K \\ &= \frac{1 - (p_u^{\text{STA}})^{K+1}}{1 - p_u^{\text{STA}}} \end{aligned}$$

and

$$\begin{aligned} &\mathbb{E}[\text{Number of attempts per TCP ACK packet}] \\ &= \frac{1 - (p^{\text{AP}})^{K+1}}{1 - p^{\text{AP}}}. \end{aligned}$$

Substituting those into (4) gives

$$\tau_u^{\text{STA}} = \frac{(1 - p^{\text{AP}})(1 - (p_u^{\text{STA}})^{K+1})}{(1 - p_u^{\text{STA}})(1 - (p^{\text{AP}})^{K+1})} \frac{D\tau_u^{\text{AP}}}{N_u}. \quad (5a)$$

In the calculation of the attempt probability of download stations, the roles of data and acknowledgement packets are reversed, and so delayed acknowledgements causes a division by D instead of multiplication. Thus

$$\tau_d^{\text{STA}} = \frac{(1 - p^{\text{AP}})(1 - (p_d^{\text{STA}})^{K+1})}{(1 - p_d^{\text{STA}})(1 - (p^{\text{AP}})^{K+1})} \frac{\tau_d^{\text{AP}}/D}{N_d} \quad (5b)$$

The collision probabilities p^{AP}, p_u^{STA}, and p_d^{STA} are

$$p^{\text{AP}} = 1 - (1 - \tau_u^{\text{STA}})_u^N (1 - \tau_d^{\text{STA}})_d^N \quad (6a)$$

$$p_u^{\text{STA}} = 1 - (1 - \tau_u^{\text{STA}})^{N_u - 1} (1 - \tau_d^{\text{STA}})_d^N (1 - \tau^{\text{AP}}) \quad (6b)$$

$$p_d^{\text{STA}} = 1 - (1 - \tau_u^{\text{STA}})^{N_u} (1 - \tau_d^{\text{STA}})^{N_d - 1} (1 - \tau^{\text{AP}}). \quad (6c)$$

Let $\mathbb{E}[Y]$ be the average slot duration. The total upload and download throughput S_u and S_d in packets/s are then calculated as in [29]:

$$S_u = \left(\frac{\tau_u^{\text{STA}}(1 - p_u^{\text{STA}})}{\mathbb{E}[Y]} \right) N_u \quad (7a)$$

$$S_d = \left(\frac{\tau_d^{\text{STA}}(1 - p_d^{\text{STA}})}{\mathbb{E}[Y]} \right) N_d D. \quad (7b)$$

The additional factor of D in the expression for S_d arises because each transmission (of a TCP ACK) by a downloading station causes the reception of D data packets. Here

$$\mathbb{E}[Y] = a_i \sigma + a_s T_{\text{ack}} + a_l T_{\text{data}} \quad (8)$$

where a_i, a_s and a_l are the probabilities that a slot is free, is busy due to the successful transmission of a TCP ACK packet or collision among TCP ACK packets from different stations, or is busy due to the successful transmission of a data packet or collision among at least one data packet and other packets. Then,

$$a_i = (1 - \tau_u^{\text{STA}})^{N_u} (1 - \tau_d^{\text{STA}})^{N_d} (1 - \tau^{\text{AP}}) \quad (9a)$$

$$a_s = (1 - \tau_u^{\text{STA}})^{N_u} \left(1 - \tau_d^{\text{AP}} - (1 - \tau_d^{\text{STA}})^{N_d} (1 - \tau^{\text{AP}})\right) \quad (9b)$$

$$a_l = 1 - a_i - a_s. \quad (9c)$$

To obtain S_u and S_d, we solve the system of fixed point equations (2)–(9).

1) Model for Mode 2: Recall that in mode 2, uplink traffic (i.e. data packets from upload stations and TCP ACK packets from download stations) is sent on one channel and downlink traffic (i.e. data packets and TCP ACK packets from the AP) is sent on the other channel. To determine the total download and total upload throughput in this mode, we also assume that the AP always has a packet waiting to be sent. This means that the downlink channel is the bottleneck and the throughput of the network is determined by the throughput obtained on the downlink channel. Therefore, the model for Mode 2 only considers the downlink channel as follows.

Because there is only the AP transmitting on the downlink channel, the collision probability p^{AP} is

$$p^{\text{AP}} = 0. \tag{10}$$

Note that the uplink and downlink now have separate slot structures. Let $\mathbb{E}[Y_d]$ denote the average slot time on the downlink. Instead of (7), the total upload and download throughput are thus given by

$$S_u = \frac{\tau_u^{\text{AP}}}{\mathbb{E}[Y_d]} D \tag{11a}$$

$$S_d = \frac{\tau_d^{\text{AP}}}{\mathbb{E}[Y_d]} \tag{11b}$$

where τ_u^{AP} and τ_d^{AP} are determined from (3) with τ^{AP} given by (2). Note that these differ from (7) by a factor of N_u or N_d, because τ_u^{AP} and τ_d^{AP} refer to the total probability that the AP will send to *any* upload or download station, respectively. Moreover, the coefficient D has moved to S_u rather than S_d because the expression is based on the transmissions by the AP instead of the stations, and the AP transmits TCP ACKs for the uploads whereas stations transmit them for the downloads.

Finally, $\mathbb{E}[Y_d]$ is determined from (8) with

$$a_i = 1 - \tau^{\text{AP}} \tag{12a}$$

$$a_s = \tau_u^{\text{AP}} \tag{12b}$$

$$a_l = \tau_d^{\text{AP}}. \tag{12c}$$

To obtain S_u and S_d, we solve the system of fixed point equations (2), (3), (8), (11) and (12).

V. Performance evaluation

In this section, we validate the accuracy of the proposed model and evaluate the performance of each of the three channel sharing modes described in III. To this end, we compare the numerical results obtained from the proposed model with the results obtained from the ns-2 simulation [30] in each mode. We implemented in ns-2 a multi-interface support required in Mode 2 in a similar way to [31].

We simulate an infrastructure WLAN network as described in Section IV with N_u stations uploading and N_d stations downloading TCP traffic through an access point (AP). All stations use actual TCP NewReno without delayed acknowledgements ($D = 1$), without the simplification assumptions used in the model. We consider the use of two consecutive 20MHz channels.

TABLE I

MAC and PHYS parameters for 802.11g systems

Parameter	Symbol	Value
Data bit rate	R_{data}	54 Mbps
Control bit rate	R_{ctrl}	11 Mbps
PHYS header	T_{phys}	20 μs
MAC header	L_{mac}	288 bits
ACK packet	L_{ACK}	112 bits
Slot time	σ	9 μs
Short Interframe Space	SIFS	10 μs
DCF Interframe Space	DIFS	28 μs
$\text{CW}_{\min}$	$W_{\min}$	16
Retry limit	K	7
Doubling limit	m	5

The general MAC and physical layer parameters of each channel are shown in Table I. Note that in Mode 0, two 20MHz channels are combined into one channel with double the bandwidth where we set the data and control bit rates to be twice those of a 20MHz channel (i.e. 108 and 22 Mbps, respectively).

The T_{data} and T_{ack} in (8) are

$$T_{\text{data}} = \text{DIFS} + T_{\text{data}}^p + \text{SIFS} + T_{\text{ACK}}^p$$
$$T_{\text{ack}} = \text{DIFS} + T_{\text{ack}}^p + \text{SIFS} + T_{\text{ACK}}^p$$

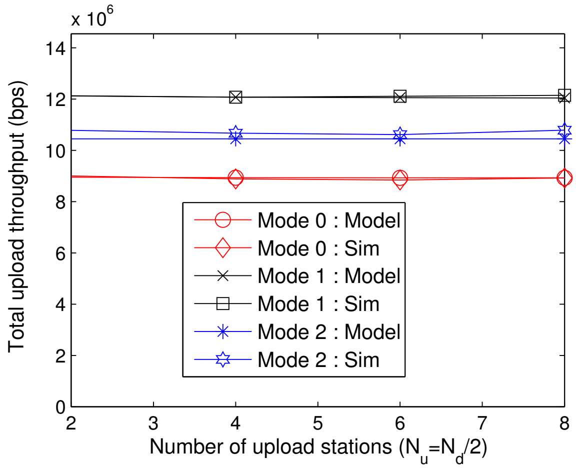
where T_{data}^p, T_{ack}^p and T_{ACK}^p are the transmission time of a data packet, a TCP ACK packet and an ACK packet of MAC layer, respectively. Those are given by

$$T_{\text{data}}^p = T_{\text{phys}} + \frac{L_{\text{mac}} + L_{\text{data}}}{R_{\text{data}}}$$
$$T_{\text{ack}}^p = T_{\text{phys}} + \frac{L_{\text{mac}} + L_{\text{ack}}}{R_{\text{data}}}$$
$$T_{\text{ACK}}^p = T_{\text{phys}} + L_{\text{ACK}}/R_{\text{ctrl}}$$

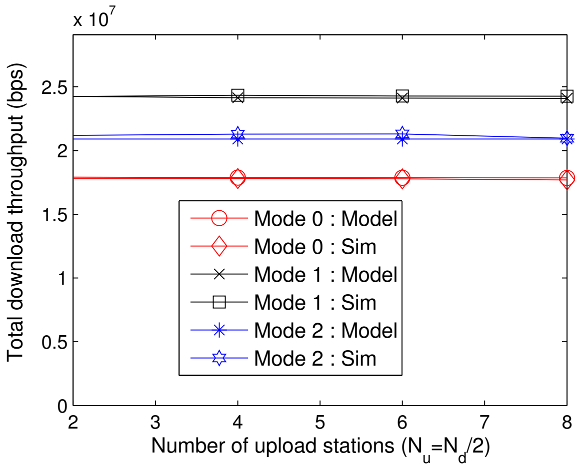
where R_{data} and R_{ctrl} are the data and control bit rates. We define the “congestion level” of a channel as the ratio

$$\text{congestion level} = \frac{\text{total time when there is a collision}}{\text{total time the channel is busy}}.$$

We will use this measure to compare the performance of the three modes. In the following, we consider scenarios where the assumptions of the model hold. In particular, the awnd of a TCP sender is set to 50 while the buffer size of a TCP receiver was chosen so that there is no buffer overflow at the receiver. Note that although the following results are for TCP without delayed ACKs, we observe that the results for delayed ACKs are qualitatively similar.



(a) Upload throughput



(b) Download throughput

Fig. 1. Total upload and download TCP throughput in bps as a function of the number of upload stations. ($L_{\text{data}} = 1040$B, $L_{\text{ack}} = 40$B, $N_u = N_d/2 = \{2, 4, 6, 8\}$.)

A. Asymmetric traffic: $N_u = N_d/2$

We first consider a network with the number of download stations being twice the number of upload stations ($N_u = N_d/2$). This reflects the fact that in practice the download traffic is typically higher than the upload traffic. The total throughput of upload and download TCP flows under the three modes are shown in Figures 1(a) and 1(b). These figures show that the model gives an accurate estimate of the total upload and download throughput under all three modes.

Moreover, it can be seen from Figure 1 that the best performance is obtained by Mode 1, which has two bidirectional channels. Compared with Mode 0 (a single wide channel), Mode 1 improves the upload and download throughput by 36%. In contrast, Mode 2 (two unidirectional channels) only improves the throughput over Mode 0 by 18%.

Note that when the number of stations increases, the total upload and download throughput under the three modes do not change. This can be explained through Figure 2 from ns-2 simulation, which shows that the congestion level of each channel under each mode does not change with the number of stations . The observation for Modes 0 and 1 is consistent with that published in prior work [15] which shows that the average number of backlogged stations at any given time does not change significantly with the number of stations and is bounded by three active stations (including the AP).

From Figures 1 and 2, one may find it counterintuitive that the throughput under Mode 1 is higher than that under Mode 2 despite the fact that the congestion level of both uplink and downlink channels under Mode 2 is smaller. This can be explained as follows. First, note that the accuracy of the model under Mode 2 implies that the assumption that AP is saturated holds in this scenario, which we have confirmed in our ns-2 simulation. This means that the throughput obtained under Mode 2 is limited by the throughput of the downlink channel where only the AP is transmitting, and the channel is never used while the AP decreases its backoff counter. This suggests that the AP in Mode 2 should use a smaller backoff level, CW_{min}. In the other two modes, each channel supports multiple stations, counting down their backoff counters in parallel, and so backoffs waste less capacity. The congestion levels are similar, since in each case the number of backlogged stations (i.e., those not prevented from transmitting by flow control) remains roughly constant in each channel.

B. Symmetric traffic: $N_u = N_d$

We now consider a network with equal number of upload and download TCP stations ($N_u = N_d$). The total throughputs of upload and download TCP flows under three modes are shown in Figures 3(a) and 3(b).

From Figures 3(a) and 3(b), it is clear that the model again gives an accurate estimate of the total upload and download throughput under both Modes 0 and 1. However, the model fails to capture those under Mode 2. Performing further investigation, we find that this is cause by the violation of the assumption that the AP is assumed to be saturated. In other words, when $N_u = N_d$, the AP is not saturated under Mode 2. To explain this, we plot the corresponding congestion level under three modes from ns-2 simulation in Figure 4. This figure shows that the congestion level of the uplink channel

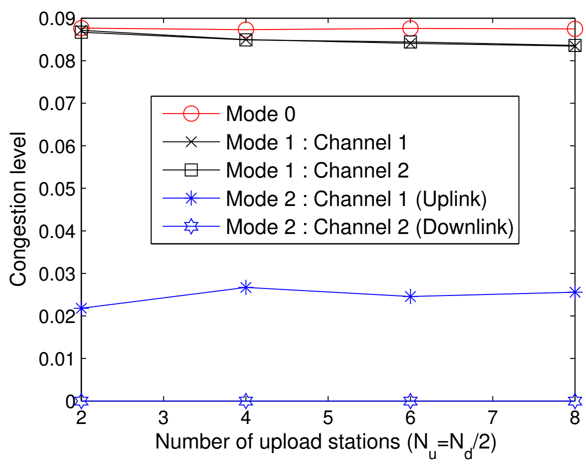
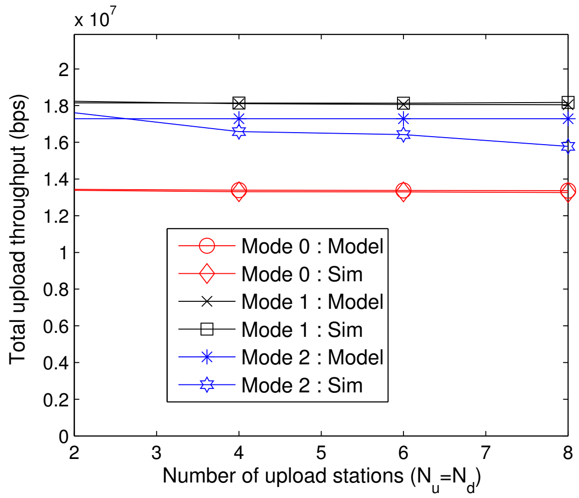
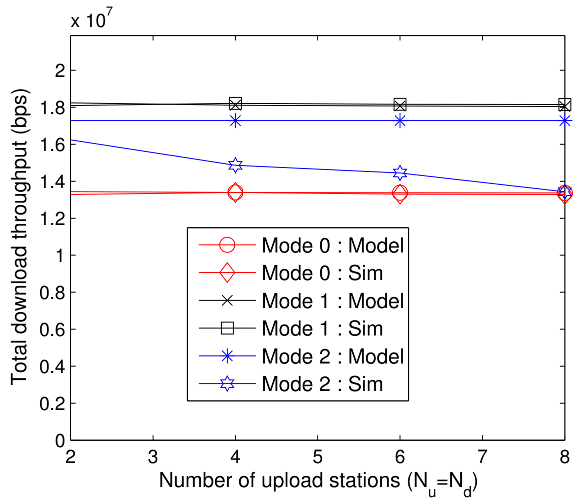


Fig. 2. Congestion level of each channel in each mode (ns-2 simulation). ($L_{\text{data}} = 1040\text{B}$, $L_{\text{ack}} = 40\text{B}$, $N_u = N_d/2 = \{2, 4, 6, 8\}$.)



(a) Upload throughput



(b) Download throughput

Fig. 3. Total upload and download TCP throughput in bps as a function of the number of upload stations. ($L_{\text{data}} = 1040\text{B}$, $L_{\text{ack}} = 40\text{B}$, $N_u = N_d = \{2, 4, 6, 8\}$.)

under Mode 2 is higher than that of each channel under Mode 0 and Mode 1. This means that the uplink channel is the bottleneck in this mode, which limits the transmission of the data packets for the upload flows and TCP ACKs for download flows. As a consequence, this prevents the AP from being saturated. Enhancing the model to describe this situation is left as future work.

Observe from Figure 3 that Mode 1 again performs the best among the three modes. Mode 2 performs better than Mode 0 in terms of both total download and upload throughput for a small number of stations. When the number of stations increases, the total download throughput decreases substantially while the total uplink throughput also decreases but by a smaller amount.

In summary, among the three modes, Mode 1 has the best performance in most cases. Furthermore, when the number of download stations is substantially higher than that of uplink stations, Mode 2 performs better than Mode 0. However, when the number of stations in each case is equal, the performance of Mode 2 decreases below that of Mode 0 when the number of stations is large. Note that similar observations are also found in other simulated scenarios.

VI. Conclusion

In this report, we have studied the performance of three modes which utilize two wireless channels concurrently. We have found that the default mode in IEEE 802.11 standard, which combines two channels into a channel with double the bandwidth, performs worst among those in most scenarios. Furthermore, we have also found that the load balancing mode performs well over a wide range of scenarios. To assist with this study, we proposed a tractable model of TCP traffic in IEEE 802.11 WLANs and shown that it gives accurate estimation of TCP throughput under scenarios in which TCP only performs flow control, and the AP is the bottleneck.

This study paves the way for future studies of a wider range of channel allocation policies. For example, in the case that one channel is used for uplink traffic and one for downlink traffic, it may be beneficial to modify the MAC to make the downlink channel more aggressive, since the AP will be the only station transmitting, in the absence of other nearby APs.

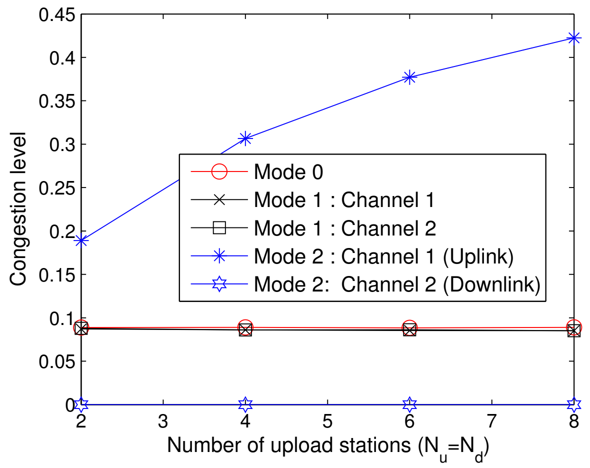


Fig. 4. Congestion level of each channel in each mode (ns-2 simulation). ($L_{\text{data}} = 1040$B, $L_{\text{ack}} = 40$B, $N_u = N_d = \{2, 4, 6, 8\}$.)

References

[1] Wi-Fi Alliance Press Releases. "Wi-Fi innovations and user enthusiasm propel continued sales growth, " (Online). Available at ⟨http://www.wi-fi.org/media/press-releases/⟩.

[2] "IEEE Std., IEEE 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control Quality of Service Enhancements", September 2005.

[3] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multi channel MAC protocol with on-demand channel assignment for multihop mobile ad hoc networks", *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pp. 232237, 2000.

[4] J. Chen, S. Sheu and C. Yang, "A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs," in *Proc. 14th IEEE Intl Symp. Personal, Indoor and Mobile Radio Comm.*, vol. 3, pp. 2291-2296, 2003.

[5] I. Wormsbecke and C. Williamson, "On Channel Selection Strategies for Multi-Channel MAC Protocols in Wireless Ad Hoc Networks," in *Proc. of the 2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 212-220, 2006.

[6] P. Tan and M. C. Chan, "AMCM: Adaptive multi-channel MAC protocol for IEEE 802.11 wireless networks, " in *Proc. 3rd Int. Conf. BROADNETS*, pp. 1–10, 2006.

[7] A. Nasipuri and S. Das, "A Multichannel CSMA MAC Protocol for Mobile Multihop Networks," in *Proc. of IEEE WCNC*, pp. 1402–1406, 1999.

[8] C. Xu, Y. Xu, J. Gao, and J. zhou, "Multi-Transceiver based MAC Protocol for IEEE 802.11 Ad Hoc Networks," in *Proc. of IEEE WCNM*, pp. 804-807, 2005.

[9] S. Mangold and J. Habetha, "IEEE 802.11 contention-based medium access for multiple channels," in *Proc. of the 11th European Wireless Conference*, vol. 2, pp. 522–526, 2005.

[10] A. Abouzeid, S. Roy, and M. Azizoglou, "Comprehensive Performance Analysis of a TCP Session over a Wireless Fading Link with Queueing," *IEEE Trans. Wireless Comm.*, vol. 2, no. 2, pp. 344-356, Mar. 2003.

[11] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," *Proc. IEEE INFOCOM*, vol. 2, pp. 863-872, 2003.

[12] M. Rossi, R. Vicenzi, and M. Zorzi, "Accurate Analysis of TCP on Channels with Memory and Finite Round-Trip Delay," *IEEE Trans. Wireless Comm.*, vol. 3, no. 2, pp. 627-640, Mar. 2004.

[13] M. Garetto, R. Lo Cigno, M. Meo, and M. Marsan, "Closed Queueing Network Models of Interacting Long-Lived TCP Flows," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 300-311, Apr. 2004.

[14] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses," *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 356-369, Apr. 2005.

[15] R. Bruno, M. Conti, and E. Gregori, "Throughput analysis and measurements in IEEE 802.11 WLANs with TCP and UDP traffic flows," *IEEE Transactions on Mobile Computing*, 7(2):171-186, 2008.

[16] R. Bruno, M. Conti, and E. Gregori, "Analytical Modeling of TCP Clients in Wi-Fi Hot Spot Networks, " in *Proc. IFIP-TC6 Networking Conf.*, pp. 626-637, May 2004.

[17] J. Yu and S. Choi, "Modeling and Analysis of TCP Dynamics over IEEE 802.11 WLAN," in *Proc. Fourth IFIP/IEEE Ann. Intl Conf. Wireless On-Demand Network Systems and Services (WONS07)*, Jan. 2007.

[18] B. Bellalta, M. Meo, and M. Oliver, "Comprehensive Analytical Models to Evaluate the TCP Performance in 802.11 WLANs," *The 4th Wired/Wireless Internet Communications (LNCS)*, Bern, Switzerland, May 2006.

[19] G. Kuriakose, S. Harsha , A. Kumar, and V. Sharma, "Analytical models for capacity estimation of IEEE 802.11 WLANs using DCF for internet applications," *Wireless Networks*, vol.15 no.2, pp. 259-277, February 2009.

[20] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New Insights from a Fixed-Point Analysis of Single-Cell IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, vol. 3, pp. 1550-1561, Mar. 2005.

[21] T. Sakurai and S. Hanly, "Modelling TCP flows over an 802.11 wireless LAN, " in *Proc. 11th European Wireless Conference*, pp. 2-7, 2005.

[22] D. Miorandi, A. Kherani, and E. Altman, "A Queueing Model for HTTP Traffic over IEEE 802.11 WLANs," *J. Elsevier Computer Networks*, vol. 50, no. 1, pp. 63-79, Jan. 2006.

[23] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "Case for Adapting Channel width in Wireless Networks," in *Proc. ACM SIGCOMM*, pp. 135–146, 2008.

[24] R. Shravan, S. Vivek, B. Suman, and C. Ranveer, "Fluid: improving throughputs in enterprise wireless LANs through flexible channelization", *IEEE Trans. Mobile Computing*, vol. 11, no. 9, pp. 1455–1469, Sep. 2012.

[25] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained Channel Access in Wireless LAN," in *Proc. ACM SIGCOMM*, pp. 147–158, 2010.

[26] R. Braden, "RFC-1122: Requirements for internet hosts", *Request for Comments*, 1989.

[27] L. L. H. Andrew, S. V. Hanly and R. G. Mukhtar, "Active queue management for fair resource allocation in wireless networks", *IEEE Trans. Mobile Computing*, vol. 7, no. 2, pp. 231–246, 2008.

[28] S. H. Nguyen, H. L. Vu and L. L. H. Andrew, "Performance Analysis of IEEE 802.11 WLANs With Saturated and Unsaturated Sources, " *IEEE Trans. Vehicular Technology*, vol. 61, no. 1, pp. 333–345, 2012.

[29] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.
[30] "The network simulator ns-2," Available at ⟨http://www.isi.edu/nsnam/ns/⟩.
[31] "Adding Multiple Interface Support in NS-2," Available at ⟨http://personales.unican.es/aguerocr/files/ucMultiIfacesSupport.pdf⟩.