Access Point Selection for Multi-Rate IEEE 802.11 Wireless LANs

Shicong Yang, Michael Krishnan and Avideh Zakhor
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Email: {yangshic, mkrishna, avz}@eecs.berkeley.edu

***Abstract*—Access Point (AP) selection is important in WLANs as it affects the throughput of the joining station (STA). In this paper, we propose a class of AP selection algorithms to maximize the joining STA's expected throughput by considering interference at STAs, and transmit opportunities (TXOPs) at APs. Specifically, we collect a binary-valued local channel occupancy signal, called busy-idle (BI) signal, at each node and require the APs to periodically broadcast their BI signal and a quantity representing their TXOPs. This enables the joining STA to estimate throughput from candidate APs before selecting one. We use NS-2 simulations to demonstrate the effectiveness of our algorithms for saturated UDP and TCP downlink traffic, and compare them with received signal power (rxpwr) algorithm, load based algorithm, and Fukuda algorithm. For a random topology consisting of 24 APs and 60 STAs, our algorithms increase the joining STAs average throughput by as much as 42% and 24% compared to rxpwr for UDP and TCP respectively. In addition, The achieved average throughput is 90% and 93% of that obtained via the optimal selection. We also show that, in contrast to rxpwr, the throughput of proposed algorithms remain close to optimal with the increase in AP or STA density.**

I. Introduction

Wireless local area networks (WLANs) have gained increasing popularity due to their convenience as compared to wireline infrastructure. As such, WLANs are becoming the preferred technology of high-speed broadband access in homes, offices, and other hotspots. Each WLAN access point (AP) forms a Basic Service Set (BSS), and multiple BSSs can overlap to form an Extended Service Set (ESS) to provide seamless handoff for stations (STAs). Due to the dense deployment of WLANs and the use of ESS to provide roaming services, it is common for STAs to have multiple available APs to choose from. In addition, nearby BSSs often experience inter-BSS co-channel interference due to the limited number of orthogonal channels. An inappropriate AP selection typically leads to compromised service, thus it is imperative for an STA to identify and select the AP that provides the highest throughput to improve user experience.

AP selection policy is not specified in IEEE 802.11 standards. Currently the strongest received power ($rxpwr$) algorithm is the most widely used. While this strategy is straightforward and easy to implement, it is ineffective especially in hyper dense deployment scenarios where adjacent APs could use the same channel. For example, as shown in Figure 1, consider two nearby co-channel APs who cannot sense each other, and an "joining" STA within range of both APs. Assume without loss of generality, the joining STA is closer to AP 2 and hence experiences a higher received signal power from it as compared to AP 1. The $rxpwr$ would select AP 2,

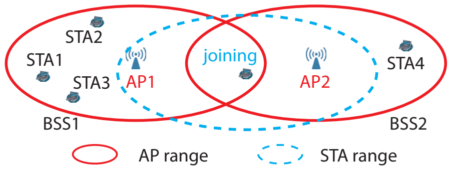


Figure 1. APs with Overlapping BSSs.

although it could experience more interference and a lower throughput. Furthermore, $rxpwr$ cannot take into account the difference between the potential transmit opportunities (TXOPs) from the two APs, which is another important factor in determining throughput. It is clear from this example that $rxpwr$ is sub-optimal for dense deployment of WLANs, and that interference, collisions and TXOPs should also be taken into account when selecting among multiple APs.

Our simulation results in Section IV also demonstrate that, for a fixed number of APs in a given region, the throughput achieved by $rxpwr$ decreases as the STA density increases. To alleviate this problem, more APs are typically deployed to provide higher throughput to each STA, which creates more hyper dense WLANs. Even for scenarios with a fixed number of users in a given area, more APs are typically added in order to achieve higher throughput per STA. An inherent problem with WLANs is that there is no closed-loop power control mechanism to adjust cell size of APs. Therefore, as the AP density increases, the joining STA can select from an increasing number of candidate APs, making the AP selection issue even more important.

To address these issues, a variety of schemes have been proposed in the literature. Some optimize potential throughput [1]–[6], whereas others propose various load balancing algorithms [7]–[10]. Furthermore, a number of approaches take interference and collision into account for selecting APs [11]–[14]. Last but not least, some researchers approach AP selection problem from fairness point of view [15]–[18].

In this paper, we propose a class of AP selection algorithms to maximize a joining STA's DL expected throughput. Our proposed AP selection metrics not only consider TXOPs at APs, but also take into account the inter-BSS interference with a more accurate collision estimation technique. We use the framework in [19], which provides a method to estimate the collision probability for UL traffic at a given STA. The basic idea behind [19] is that all STAs and APs continually measure the spatial channel occupancy around them, with APs periodically broadcasting a compressed binary-valued busy-idle (BI) signal to indicate their local channel occupancy to all associated STAs. Each STA can then estimate UL collision probability by comparing its local BI signal with that of the

AP's. We extend this framework to estimate DL collision probability, and then compute decision metrics at the joining STA to select an AP.

Throughout the paper, we use STA to refer to a non-AP station, and use *node* to refer to either an AP or an STA. The remainder of the paper is organized as follows: Section II discusses our packet loss model, and the method to estimate each component of packet loss; Section III describes our proposed algorithms; Section IV presents the performance evaluations, and Section V concludes the paper.

II. Packet Loss Modeling and Estimation

We categorize packet loss in WLANs into two classes: collisions and channel errors. A collision is defined as a packet failure at the intended receiver due to interference from other transmitters which are in close proximity to the receiver. A channel error is defined as an unsuccessful decoding of a packet due to low received SNR, which is caused by large path loss or deep multipath fade, given that the packet does not suffer from collisions. The probability of total packet loss can be expressed as:

$$P_L = 1 - (1 - P_C)(1 - P_e) \tag{1}$$

where P_C is the packet loss probability due to collisions, and P_e is the packet loss probability due to channel error given that the packet does not experience collisions. Equivalently, the packet success rate P_S is given by:

$$P_S = 1 - P_L = (1 - P_C)(1 - P_e) \tag{2}$$

In this paper, we assume none of the packets suffering from collisions are captured, and are therefore assumed to be lost.

Krishnan *et al.* proposed a framework to estimate UL collision probabilities at STAs, using the local channel occupancy at the STA as well as the periodically broadcasted BI signal from its AP reflecting the AP's local channel occupancy [19]. We can generalize the estimator in [19] to estimate the collision probability on link (Tx, Rx) as follows:

$$P_C(Tx, Rx) = f(BI_{Tx}, BI_{Rx}) \tag{3}$$

where BI_{Tx} and BI_{Rx} are BI signals collected at the transmitter and the receiver, respectively. For DL, suppose AP i is the Tx and STA j is the Rx, hence:

$$P_C(i, j) = f(BI_{AP_i}, BI_{STA_j}) \tag{4}$$

We classify collisions into three types: direction collisions (DCs), staggered collisions of type 1 (SC1), and type 2 (SC2) [19]. A DC for a given node is a collision in which the node under consideration starts transmitting at the same time as other nodes. An SC1 for a given node is a collision in which the node under consideration transmits first and is then interrupted by a hidden node. An SC2 for a given node is a collision in which the node under consideration interrupts the transmission of a hidden node. Intuitively, for the node under consideration, an SC2 occurs when another node is already transmitting to the intended receiver before the node starts to transmit, a DC occurs when another node starts transmitting at the same time the node starts to transmit, and an SC1 occurs when another node starts transmitting later than, but interrupts, the node's transmission. Based on the above description, $(1 - P_C)$ can be expanded into [19]:

$$(1 - P_C) = (1 - P_{SC2})(1 - P_{DC})(1 - P_{SC1}) \tag{5}$$

where P_{SC2} denotes the probability of SC2, P_{DC} denotes the probability of DCs given that it does not experience SC2, and P_{SC1} denotes the probability of SC1 given that it experiences neither SC2 nor DC [19]. Due to the way collisions are counted, SC2 is the dominant type of collision for high traffic scenarios [19], and can thus be used to approximate the total collision probability in a traffic-saturated WLAN as:

$$P_C(i, j) \approx P_{SC2}(i, j) = \frac{\sum_t \mathbb{1}\{BI_{APi}(t) = 0, BI_{STAj}(t) = 1\}}{\sum_t \mathbb{1}\{BI_{APi}(t) = 0\}} \tag{6}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. The intuition is that this is the probability that the channel is busy at the STA given that it is idle at the AP; hence if at time t a packet was transmitted by the AP when AP senses the channel to be idle, i.e., $BI_{APi}(t) = 0$, it would have experienced collision at the STA with probability $P_C(i, j)$.

An 802.11 packet uses PHY modulation rate R_{PHY} for preamble and PLCP header, and potentially higher modulation rates R_{MAC} for MAC frame. The probability of channel error for packets from AP i to STA j can be expressed as [20]:

$$\begin{aligned} P_e(i, j) = 1 - &(1 - BER_{R_{\text{PHY}}}(SNR_{ij}))^{L_{\text{PHY}}} \\ &(1 - BER_{R_{\text{MAC}}}(SNR_{ij}))^{L_{\text{MAC}}} \end{aligned} \tag{7}$$

where L_{PHY} and L_{MAC} are the lengths of the preamble and PLCP header, and MAC frame, respectively. $BER_R(SNR)$ denotes the bit error rate assuming it is a known function of modulation rate R and SNR. SNR_{ij} can be estimated as:

$$SNR_{ij} = Pr_{ij}/Noise \tag{8}$$

where Pr_{ij} is the received power of beacons from AP i to STA j, and $Noise$ is the thermal noise to be estimated from:

$$Noise(\text{dBm}) = -174 + 10\log_{10}(W) + N_f \tag{9}$$

where W is the bandwidth of wireless transmission, and N_f is the noise figure of the wireless system, which is a property of hardware. Substituting Equation (9) into Equation (8), SNR_{ij} can be estimated and consequently the channel error probability $P_e(i, j)$ can be computed as Equation (7).

With the estimates of collision probability $P_C(i, j)$ and channel error probability $P_e(i, j)$, the total loss probability $P_L(i, j)$ can be computed as Equation (1). We use $P_L(i, j)$ to estimate average backoff time if STA j associates with AP i, and to compute our proposed decision metric in Section III.

III. Proposed AP Selection Algorithm

In this section we describe a class of AP selection algorithms which take into account the TXOPs, MAC rates, interference and collisions at the STA. We begin by describing our system model. We assume WLAN operates in infrastructure mode with DCF, and hence no RTS/CTS is used. All traffic flows have the same priority, and packets have Poisson arrival whose rate depends on the application layer data rate. When serving MAC Service Data Units (MSDUs), an AP does not switch to a new MSDU until the previous MSDU is successful or dropped due to its retransmission limit being exceeded. The network is assumed to be saturated with UDP or TCP DL traffic. The MAC rate is determined by the path loss from an STA to its serving AP, and no rate adaptation is assumed to be used. In this analysis we assume APs to be on the same channel, and focus on one STA j joining the network while all other STAs are already associated to and exchanging traffic with their desired APs. For ease of notation, we use $P_C(i)$, $P_e(i)$ and $P_S(i)$ in place of $P_C(i, j)$, $P_e(i, j)$ and $P_S(i, j)$, respectively, since only one joining STA j is considered.

In our proposed algorithm, both APs and the joining STA record their BI signals at a resolution of $10\mu sec$ as suggested in [19]; this sampling period provides a good balance between estimation error and transmission overhead. APs broadcast the BI signals every $3sec$ with the overhead to send BI signal being about 3% in the 802.11b network [19]. Before associating to any AP, the joining STA stays idle and records its local BI signal for the first $3sec$.

Our approach to AP selection is to maximize DL expected throughput (eTP):

$$AP_{sel} = arg \max_{i \in \mathcal{A}} (eTP(i)) \quad (10)$$

where $\mathcal{A}$ is the set of candidate APs that the joining STA can choose from. We define $eTP(i)$ as:

$$\begin{aligned} eTP(i) :&= \frac{\text{total successful MSDU in bits from AP } i \text{ to joining STA}}{\text{total time}} \\ &= \frac{\text{total time to send MSDU by AP } i \text{ to joining STA}}{\text{total time}} \\ &\times \frac{\text{total successful MSDU in bits from AP } i \text{ to joining STA}}{\text{total time to send MSDU by AP } i \text{ to joining STA}} \quad (11) \\ &= t_{\text{alloc}}(i) \times TP_{\text{MAC}}(i) \quad (12) \end{aligned}$$

where we denote the first term in Equation (11) as $t_{\text{alloc}}(i)$, representing the percentage of channel time that AP i can allocate to the joining STA, and denote the second term as $TP_{\text{MAC}}(i)$, representing the expected MAC layer throughput from AP i to the joining STA. Note the MSDU in Equation (11) is also known as MAC payload, and we use them interchangeably in this paper. We refer the second term as the MAC layer throughput because it counts only the successfully delivered MAC payloads at a given STA. In contrast, PHY rate is the one used by an AP to modulate packets to an STA, which does not take backoff, header and other overhead into account. The potential TXOP from AP i is captured in $t_{\text{alloc}}(i)$ term, while the effect of MAC rate, interference and collisions is considered in both $t_{\text{alloc}}(i)$ and $TP_{\text{MAC}}(i)$. We elaborate on how to estimate these two terms in the following sections.

A. Estimating $TP_{MAC}(i)$

The $TP_{\text{MAC}}(i)$ from AP i to the joining STA is defined as successful number of MAC payload bits transmitted over the time that AP i spent for delivering those data, including packet transmission time and all associated overhead time. $TP_{\text{MAC}}(i)$ can be expressed as:

$$\begin{aligned} TP_{\text{MAC}}(i) :&= \frac{\text{total successful MSDU in bits from AP } i \text{ to joining STA}}{\text{total time to send MSDU by AP } i \text{ to joining STA}} \\ &= \frac{\sum_m L_i(m) \times \mathbb{1}\{A_i(m)\}}{\sum_m t_i(m)} \quad (13) \end{aligned}$$

where $L_i(m)$ is the MSDU size in bits from AP i on the mth Physical layer Protocol Data Unit (PPDU) transmission, $t_i(m)$ is the time that AP i spent on the mth PPDU transmission, backoff and protocol overhead, $A_i(m)$ is the event that the mth PPDU sent by AP i to the joining STA is successful, and $\mathbb{1}\{\cdot\}$ is the indicator function defined by:

$$\mathbb{1}\{A_i(m)\} = \begin{cases} 1 & \text{if } m\text{th PPDU sent by AP } i \text{ succeeds} \\ 0 & \text{if } m\text{th PPDU sent by AP } i \text{ fails} \end{cases} \quad (14)$$

If we assume the maximum MAC payload size L is used for each packet, Equation (13) can be rewritten as:

$$TP_{\text{MAC}}(i) = \frac{L \times \sum_m \mathbb{1}\{A_i(m)\}}{\sum_m t_i(m)} = \frac{L \times P_S(i)}{t(i)} \quad (15)$$

where $P_S(i)$ is the packet success probability from AP i to the joining STA given by Equation (2), $t(i)$ is the average time that AP i allocates to the joining STA for one packet transmission including backoff time, PPDU transmission time and protocol overhead, given by:

$$t(i) = t_p(i) + t_{OH}(i) \quad (16)$$

where $t_p(i)$ is the time for AP i to transmit MAC payload to the joining STA, and $t_{OH}(i)$ is the average overhead of one MSDU transmission from AP i. Substituting Equation (16) into (15) and rearranging the terms, we obtain:

$$\begin{aligned} TP_{\text{MAC}}(i) &= \frac{L}{t_p(i)} \times P_S(i) \times \frac{t_p(i)}{t_p(i) + t_{OH}(i)} \\ &= R_{\text{MAC}}(i) \times P_S(i) \times \frac{t_p(i)}{t_p(i) + t_{OH}(i)} \end{aligned} \quad (17)$$

where $R_{\text{MAC}}(i) = L/t_p(i)$ is the rate used by AP i to modulate MAC payload. Substituting Equation (2) into (17), we obtain the following expression for $TP_{\text{MAC}}(i)$ at AP i:

$$TP_{\text{MAC}}(i) = R_{\text{MAC}}(i) \times (1 - P_C(i)) \times (1 - P_e(i)) \times \frac{t_p(i)}{t_p(i) + t_{OH}(i)} \quad (18)$$

where $P_C(i)$ and $P_e(i)$ are the DL collision probability and channel error probability given by Equations (6) and (7), respectively. Next we explain how to estimate each component in Equation (18) in order to optimally select the AP.

$R_{\text{MAC}}(i)$ depends on SNR_i from AP i to the joining STA. Assuming the function to map SNR to MAC rate is known, the MAC rate $R_{\text{MAC}}(i)$ used by AP i can be predicted as long as SNR is estimated as in Equation (8).

$t_p(i)$ is the time to transmit MAC payload, i.e., MSDU. It depends on the payload size L in bits and MAC rate $R_{\text{MAC}}(i)$:

$$t_p(i) = L/R_{\text{MAC}}(i) \quad (19)$$

Maximum payload is typically used in WLANs to improve transmission efficiency.

$t_{OH}(i)$ is the average overhead time for AP i to deliver an MSDU for one time, which includes preamble, PLCP header, MAC header and CRC, inter-frame spacing time, possible ACK time, and backoff time. The derivation is given in [21].

Once $t_{OH}(i)$ and the corresponding values in Equations (6), (7), and (19) are estimated by the joining STA, Equation (18) can be used to evaluate $TP_{\text{MAC}}(i)$.

B. Estimating $t_{alloc}(i)$

The $t_{\text{alloc}}(i)$ is a unit-less term representing the expected percentage of channel time that AP i can allocate to the joining STA. We propose four different ways to estimate $t_{\text{alloc}}(i)$, resulting in four different decision metrics for selecting AP:

1) TP_{MAC}: $t_{\text{alloc}}(i)$ is a constant.
2) eTP_n: $t_{\text{alloc}}(i)$ is estimated by the number of STAs that are already associated with AP i.
3) eTP_r: $t_{\text{alloc}}(i)$ is estimated by the MAC rates of STAs that are already associated with AP i.
4) eTP_t: $t_{\text{alloc}}(i)$ is estimated by calculating the average waiting time between consecutive unique MSDUs from AP i to the joining STA.

We now describe each method, and provide detailed evaluations in Section IV.

1) TP_{MAC}: The simplest way is to estimate t_{alloc} as a constant. Without loss of generality, we can set $t_{\text{alloc}} = 1$. Substituting this into Equation (12), the metric becomes:

$$eTP(i) = t_{\text{alloc}}(i) \times TP_{\text{MAC}}(i) = TP_{\text{MAC}}(i) \quad (20)$$

which is equivalent to TP_{MAC}. We denote this decision metric as TP_{MAC} in subsequent sections.

2) eTP_n: In this case we denote t_{alloc} as t^n_{alloc}. Let $N_{\text{assoc}}(i)$ be the number of STAs that are already associated with AP i, which can be broadcasted by the AP along with the BI signals. Assuming that APs can allocate equal amount of channel time to each associated STA, t^n_{alloc} can be computed as:

$$t^n_{\text{alloc}}(i) = 1/(N_{\text{assoc}}(i) + 1) \tag{21}$$

Substituting Equation (21) into Equation (12), we obtain:

$$eTP_n(i) = TP_{\text{MAC}}(i)/(N_{\text{assoc}}(i) + 1) \tag{22}$$

3) eTP_r: In this case we denote t_{alloc} as t^r_{alloc}. Let $R_i(k)$ be the MAC rate from AP i to the STA k. Assuming that APs can transmit equal amount of data in bits to each associated STA, t^r_{alloc} can be estimated as:

$$t^r_{\text{alloc}}(i) = \frac{1/R_{\text{MAC}}(i)}{R_{\text{MAC}}(i) + \sum_{k\in\mathcal{S}_i} 1/R_i(k)} \tag{23}$$

where R_{MAC} is estimated as explained in Section III-A, $\mathcal{S}_i$ is the set of STAs that are already associated with AP i. Intuitively speaking, the higher MAC rate one STA can get from AP i, the less time AP i spends to transmit packets to the STA, and hence the less channel time AP i would allocate to the STA. Specifically for the joining STA, the higher MAC rate other associated STAs can get from AP i, the more channel time AP i can allocate to the joining STA.

Substituting Equation (23) into Equation (12), we obtain:

$$eTP_r(i) = \frac{1/R_{\text{MAC}}(i)}{1/R_{\text{MAC}}(i) + \sum_{k\in\mathcal{S}_i} 1/R_i(k)} \times TP_{\text{MAC}}(i) \tag{24}$$

Note the quantity $\sum_{k\in\mathcal{S}_i} 1/R_i(k)$ can be computed by AP i and broadcast along with BI signals.

4) eTP_t: In this case, we denote eTP as eTP_t and t_{alloc} as t^t_{alloc}. In practice, $t_{\text{alloc}}(i)$ is not only determined by the number of associated STAs and the MAC rate of those STAs, but also determined by the packet success probability to each STA. For example, if one STA has excessively low success probability, the AP would have to keep retransmitting to it and hence spend less time on other STAs. $t_{\text{alloc}}(i)$ is also affected by how often AP i has to wait for the transmissions of other APs and STAs due to CSMA. The following analysis assumes that upper layer applications are sending saturating traffic, and the MAC layer serves each traffic flow in a round-Robin fashion.

In order to estimate $t^t_{\text{alloc}}(i)$, we introduce the concept of the expected waiting time $t^{\text{after}}_w(i)$ between two consecutive unique MSDUs for traffic from AP i to the joining STA if the joining STA selects AP i as its serving AP. By definition, $t^t_{\text{alloc}}(i)$ is the ratio between two quantities:

$$t^t_{\text{alloc}}(i) = t(i)/t^{\text{after}}_w(i) \tag{25}$$

where $t(i)$ is given by Equation (16). We now describe how to estimate $t^{\text{after}}_w(i)$.

$t^{\text{after}}_w(i)$ depends on the amount of channel time AP i needs to allocate to the joining STA and other associated STAs, as well as the amount of time AP i has to wait when other nodes are active. This quantity cannot be measured before AP i starts transmitting to the joining STA. However, assuming only one new STA joins the network, we can estimate $t^{\text{after}}_w(i)$ as:

$$t^{\text{after}}_w(i) = \begin{cases} t^{\text{before}}_w(i) & \text{if } t_{\text{idle}}(i) > t_u(i) \\ t^{\text{before}}_w(i) + t_u(i) - t_{\text{idle}}(i) & \text{if } t_{\text{idle}}(i) < t_u(i) \end{cases} \tag{26}$$

where $t_u(i)$ is defined to be the average time spent by AP i to deliver one unique MSDU to the joining STA, which includes backoff time, possible retransmissions, and other protocol related overheads, $t^{\text{before}}_w(i)$ is the minimum over all associated STAs' average waiting time to serve two consecutive unique MSDUs to the same STA from AP i before the joining STA selects any AP, and $t_{\text{idle}}(i)$ is the minimum over all associated STAs' average idle time at AP i for a duration of $t^{\text{before}}_w(i)$ before the joining STA selects any AP. Both $t^{\text{before}}_w(i)$ and $t_{\text{idle}}(i)$ can be computed at AP i, and be transmitted to the joining STA as additional fields in BI signals. The intuition in computing $t^{\text{after}}_w(i)$ is as follows: if AP i is not too busy and has enough idle time to serve packets to the joining STA without sacrificing the TXOPs of other already associated STAs, the average waiting times $t^{\text{before}}_w(i)$ and $t^{\text{after}}_w(i)$ should be the same before and after the joining STA enters the network; otherwise, all STAs served by AP i experience longer waiting times.

To estimate $t_u(i)$, we need to determine how many retries X_i are required to deliver one unique MSDU from AP i to the joining STA. The distribution of X_i is the stationary distribution π of the MC model shown in [21].Let $T_u(i)$ be the random variable for the time spent by AP i to send one MSDU to the joining STA until the MSDU is successful after consecutive retries or until the MSDU is dropped due to exceeding retransmission limit. $T_u(i)$ can be written as:

$$T_u(i) = \sum_{x=0}^{X_i} t(i) \tag{27}$$

where $t(i)$ is the average time that AP i allocates to the joining STA for one PPDU transmission given by Equation (16). Then $t_u(i)$ is the expected value of $T_u(i)$:

$$t_u(i) = E[T_u(i)] = E[\sum_{x=0}^{X_i} t(i)] \tag{28}$$

Expanding Equation (28), we obtain:

$$t_u(i) = E[X_i]t(i) \tag{29}$$

where

$$E[X_i] = \sum_{n=0}^{N-1} (n+1)\pi(n) \tag{30}$$

$t^{\text{before}}_w(i)$ depends on a number of factors such as the number of STAs associated with AP i, interference duration that can pause backoff due to CSMA, as well as the MAC rate and the packet loss probability to each associated STA. Instead of estimating all quantities, we require AP i to monitor $t_w(i,k)$ which is the average duration between two consecutive unique MSDUs transmitted by AP i to STA k. Then:

$$t^{\text{before}}_w(i) = \min_{k\in\mathcal{S}_i} t_w(i,k) \tag{31}$$

where $\mathcal{S}_i$ is the set of STAs associated with AP i. Intuitively $t_w(i,k)$ can be described as follows: Assume upper layer applications are sending saturated traffic, all traffic streams have the same priority, and an AP does not switch to a new MSDU until the previous MSDU is successful or dropped due to exceeding retry limit. Then the average behavior of AP i can be modeled as transmitting MSDUs to each of its associated STA in a round-Robin fashion. The intuition for $t_w(i,k)$ is that on average AP i will be served one unique MSDU to STA k within a time period of $t_w(i,k)$, as shown in Figure 2.

$t_{\text{idle}}(i)$ can be computed at AP i as follows: Define the total idle time $t^{\text{total}}_{\text{idle}}(i, t_{\text{count}})$ at AP i to be the period of time when

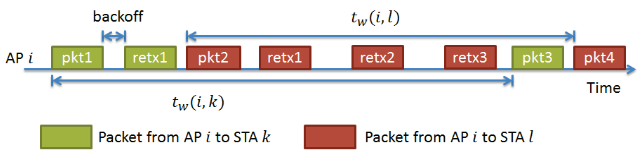


Figure 2. Illustration of $t_w(i,k)$.

Table I. SIMULATED RATE CONTROL ALGORITHM

Distance [m]	[0, 15)	[15, 20)	[20, 25)	[25, 32)
Rate [Mbps]	11	5.5	2	1

AP i is ready to transmit but there is no packet in the queue during a certain observation period denoted by t_{count}. We opt to choose t_{count} to be the same amount of time to collect BI signals, which is $3sec$ in this work. Let $N_{\text{MSDU}}(i,k,t_{\text{count}})$ be the number of MSDUs transmitted from AP i to STA j during t_{count}. Then $t_{\text{idle}}(i)$ is given by:

$$t_{\text{idle}}(i) = \frac{t_{\text{idle}}^{\text{total}}(i, t_{\text{count}})}{\max_{k\in\mathcal{S}_i} N_{\text{MSDU}}(i,k,t_{\text{count}})} \quad (32)$$

In estimating $t_{\text{idle}}(i)$ in the above equation, we assume t_{count} to be large enough for the estimate to be independent of t_{count}.

Once the joining STA receives estimated $t_w^{\text{after}}(i)$ and $t_{\text{idle}}(i)$ from AP i, given by Equations (26) and (32) respectively, $t_w^{\text{after}}(i)$ can be computed as in Equation (26). Combining with $t_u(i)$ as in Equation (29), we can compute the decision metric in this case as:

$$eTP_t(i) = \frac{t(i)}{t_w^{\text{after}}(i)} \times TP_{\text{MAC}}(i) \quad (33)$$

C. AP Selection Algorithm

We have described four AP decision metrics, namely TP_{MAC}, eTP_n, eTP_r, and eTP_t. In practice, one of the four metrics is used to estimate $eTP(i)$. The proposed AP selection algorithm is summarized in Algorithm 1.

Algorithm 1 AP Selection Algorithm

```
The joining STA scans and obtains candidate AP set 𝒜
Collect local BI for 3 seconds
Receive BI signal and other fields from all APs in 𝒜
for each AP i ∈ 𝒜 do
    compute eTP(i)
end for
The joining STA selects AP with largest eTP
```

IV. PERFORMANCE EVALUATION

A. Simulation Methodology

We use NS-2.31 to simulate 802.11b WLANs in infrastructure mode. This can be easily extended to other standards, such as 802.11a, g and n. The simulator has been modified to compute collision probability as in [19]. The range of transmission is about 32m, and no fading or shadowing is used in path loss model. Each DL stream consists of traffic from a Constant Bit Rate (CBR) application which is generating packets at a rate that saturates the network. UDP or TCP is used as the transport layer protocol. MAC retry limit is set to be 10. Both the transmit and receive antennae have 0 dB gain. A distance based rate control is used as shown in Table I.

In all simulations, both APs and STAs are randomly placed in a 110 m × 110 m region mimicking typical hyper dense scenarios observed in practice. The number of simulated APs and STAs are shown in Table II. To reduce simulation overhead, we pre-generate three different sets of random locations for each given number of APs with constraints: 1) the APs cover at least 95% of simulation area, 2) the APs have a minimum separation distance depending on number of APs as shown in Table II. Each simulation trial places all APs according to one of the pre-specified random locations. All APs are on the same channel to simulate hyper dense scenario. The STAs are placed at random according to a spatial Poisson process.

Table II. SIMULATED TOPOLOGY PARAMETERS

# AP	8	8	16	16	16	24	24	32
# STA	20	40	20	40	60	40	60	40
Minimum AP distance [m]	30	30	20	20	20	10	10	10

Table III. EXAMPLE RESULTS: NON-OPTIMAL SELECTIONS

Non-Optimal Selection		Algorithm							
		rxpwr	Load	Fukuda	TP_{MAC}	eTPn	eTPr	eTPt	optimal
8AP 40STA	UDP	17%	70%	25%	15%	14%	21%	13%	0%
	TCP	16%	77%	23%	10%	10%	17%	21%	0%
16AP 40STA	UDP	45%	69%	47%	32%	30%	28%	17%	0%
	TCP	29%	76%	54%	19%	21%	24%	36%	0%
16AP 60STA	UDP	44%	68%	50%	35%	32%	33%	22%	0%
	TCP	31%	73%	53%	23%	23%	28%	43%	0%
24AP 60STA	UDP	54%	83%	50%	29%	27%	27%	21%	0%
	TCP	45%	83%	56%	21%	21%	24%	39%	0%

Table IV. EXAMPLE RESULTS: THROUGHPUT

Ave TP [kbps]		Algorithm							
		rxpwr	Load	Fukuda	TP_{MAC}	eTPn	eTPr	eTPt	optimal
8AP 40STA	UDP	269.5	110.3	265.2	269.7	277.5	276.5	290.1	296.1
	TCP	584.5	106.6	549.7	592.2	597.3	561.0	553.8	610.8
16AP 40STA	UDP	686.7	448.4	664.7	744.5	806.0	822.6	885.6	918.7
	TCP	687.8	305.5	526.3	728.0	741.1	717.2	626.3	830.4
16AP 60STA	UDP	390.2	319.1	424.2	431.5	492.2	496.0	546.6	570.6
	TCP	468.4	255.0	373.2	483.2	504.0	489.2	401.7	560.2
24AP 60STA	UDP	551.5	295.5	631.6	732.6	781.6	788.9	840.4	865.9
	TCP	590.9	204.6	475.4	716.6	733.6	706.4	588.3	790.2

To determine the ground truth, for each simulation trial we fix AP and STA locations and run the simulations to compute the throughput of the joining STA under the exact same conditions except that the joining STA associates with different APs, in order to determine the highest throughput AP which we call "optimum". Next we run the AP selection algorithms with proposed metrics and compare their selections with those obtained from three baselines:

1) The strongest received power ($rxpwr$) algorithm: the AP with the strongest received power is chosen.
2) A simple load balancing (Load) algorithm: the AP with fewest number of associated STAs is chosen.
3) Fukuda algorithm in [8].

For each pair of number of APs and STAs shown in Table II, we run 900 trials and discard the trials whose optimal throughput is less than 1kbps. We call these discarded trials "invalid". We refer to a *non-optimal* AP selection for a given algorithm as a valid trial in which the algorithm does not result in the same AP as the optimum. The simulation time is 50 seconds.

B. Comparing Algorithms

Using extensive simulations, we found that among the three baselines, i.e., $rxpwr$, load based, and Fukuda algorithms, $rxpwr$ achieves the best performance with both UDP and TCP traffic for almost all topologies listed in Table II. Detailed simulation results demonstrating this for the selected topologies are shown in Tables III and IV. Therefore we use $rxpwr$ as the base to evaluate our proposed algorithms.

For UDP traffic, all proposed algorithms outperform $rxpwr$ in both non-optimal selection and average throughput, with

Table V. eTP_n COMPARED TO $rxpwr$ AND $optimal$

Topology		UDP			TCP		
#AP	#STA	Reduction in Non-Optimal Selection vs. rxpwr	TP Gain vs. rxpwr	% of Optimal TP Achieved	Reduction in Non-Optimal Selection vs. rxpwr	TP Gain vs. rxpwr	% of Optimal TP Achieved
8	20	-48%	2%	98%	-55%	2%	98%
8	40	-21%	3%	94%	-37%	2%	98%
16	20	-38%	16%	92%	-30%	5%	90%
16	40	-33%	17%	88%	-29%	8%	89%
16	60	-27%	26%	86%	-27%	8%	90%
24	40	-52%	31%	91%	-50%	24%	90%
24	60	-50%	42%	90%	-54%	24%	93%
32	40	-63%	68%	93%	-62%	52%	88%

eTP_t being the best, as shown in Tables III and IV. The same tables also show that eTP_n performs the best among all proposed algorithms for TCP traffic. It is interesting to note that eTP_t works well with UDP traffic but not with TCP. The main reason is that TCP congestion control can regulate the traffic flow at transport layer in the presence of packet losses, which violates the assumption for eTP_t that on average an AP can transmit MSDUs to each of its associated STA in a round-Robin fashion. STAs associated to the same AP typically experience different loss probabilities, and therefore TCP congestion control would respond and achieve different steady state traffic rate to each STA. In NS-2 mobilenode model, packets from multiple TCP flows arrive at the same interface queue, and the MAC layer serves one packet at a time from the head of that queue. If the TCP layer traffic rates to each STA are different, the packets from different TCP flows do not have the same proportion in the interface queue, and thus the round-Robin service behavior assumption falls apart.

In practice, most networks carry both UDP and TCP traffic. As such we focus on analyzing eTP_n since it achieves reasonable performance for both UDP and TCP traffic.[1] More simulation results for eTP_n with UDP and TCP traffic under different topologies are summarized in Table V. Columns three and six in Table V show the reduction in non-optimal AP selections made by eTP_n as compared to $rxpwr$. As seen, eTP_n can reduce the non-optimal selections by more than 21% and 27% in all cases for UDP and TCP, respectively. Columns four and seven in Table V summarize the average throughput gain of the joining STA obtained by eTP_n compared to $rxpwr$. In general, eTP_n achieves positive throughput gain over $rxpwr$ in every simulated topology, with up to 68% and 52% improvement in UDP and TCP, respectively. Columns five and eight in Table V show the percentage of achieved throughput by eTP_n compared to optimal, which is at least 86% and 88% for UDP and TCP, respectively.

C. Increasing Node Density

To evaluate the throughput performance of eTP_n compared to $rxpwr$, Load, Fukuda and optimal as a function of node density, we plot the average UDP and TCP throughput of the joining STA over all valid simulation trials obtained by different selection algorithms in Figure 3. As seen in Figures 3(a) and 3(b), the average throughput for all AP selection algorithms drops with number of STAs, or equivalently the STA density. For UDP, if more APs are placed in a WLAN to accommodate the increase in the number of STAs, the $rxpwr$ and Fukuda throughput drops with total node density, while

[1]Due to the limitations of the traffic models in NS-2.31, we could not simulate mixed UDP and TCP traffic. This will be our future work.

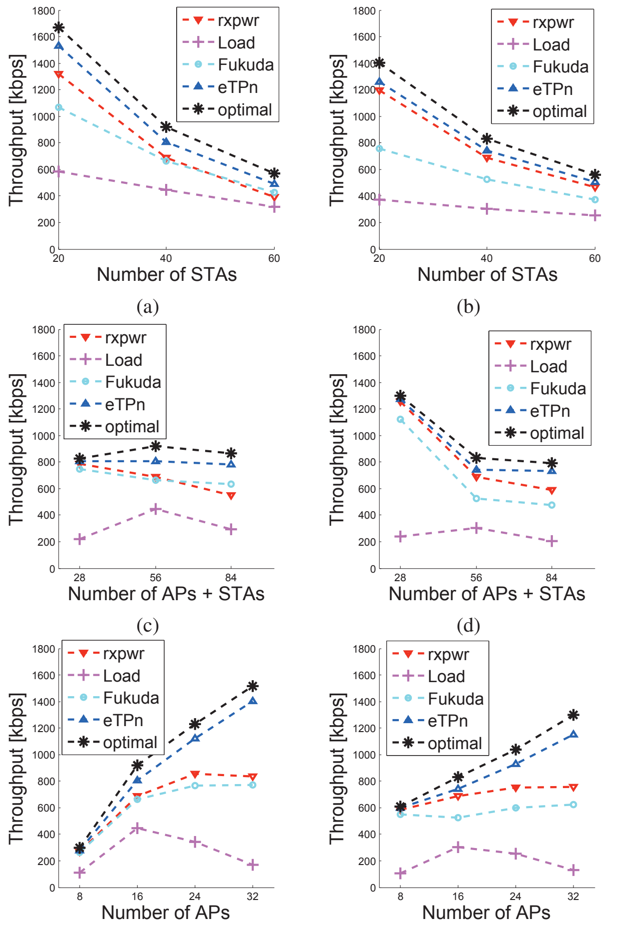
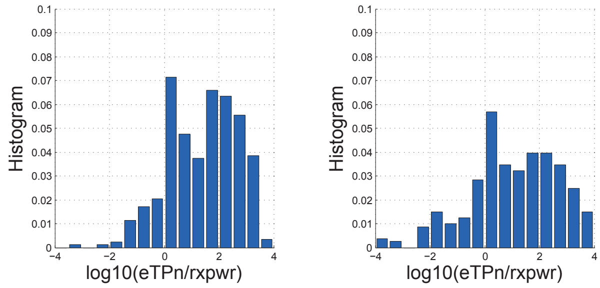


Figure 3. Average throughput of the joining STA achieved by different AP selection algorithms as a function of: (a) UDP traffic: STA density with 16 APs; (b) TCP traffic: STA density with 16 APs; (c) UDP traffic: total node density with STA to AP ratio fixed at 5:2; (d) TCP traffic: total node density with STA to AP ratio fixed at 5:2; (e) UDP traffic: AP density with 40 STAs; (f) TCP traffic: AP density with 40 STAs.

the eTP_n throughput stays almost invariant and is close to the optimal throughput, as shown in Figure 3(c). Specifically in Figure 3(c), the UDP throughput gain of eTP_n over $rxpwr$ increases with total node density, growing from 2% for 28 nodes to 42% for 84 nodes. Similarly, in Figure 3(d) the TCP throughput gain also grows with total node density from 2% for 28 nodes to 24% for 84 nodes. Furthermore, eTP_n outperforms all other baseline AP selection algorithms, and achieves at least 88% of optimal throughput in all cases shown in Figure 3(c) and 3(d). As seen in Figure 3(e), for UDP traffic, if the AP density is increased while STA density is fixed, the $rxpwr$, Load and Fukuda throughput exhibit a diminishing return, while the optimal and eTP_n throughput both increase almost linearly.[2] In particular, the UDP throughput gain for eTP_n over $rxpwr$ increases with AP density, growing from

[2]Nonetheless, we speculate that the optimal throughput for the joining STA will eventually reach diminishing return when the number of APs exceeds a certain threshold. This is because as the number of APs increases in a fixed area, the average distance between APs decreases, and the spatial frequency reuse of this WLAN decreases, which means more APs have to contend for the same amount of TXOPs. With fewer TXOPs for each AP, the throughput for each STA is likely to decrease. With proper AP selection, the threshold of diminishing return on average throughput for the joining STA is postponed to larger number of APs.



(a)

(b)

Figure 4. Histogram of the logarithm of ratio of eTP_n over $rxpwr$ throughput when decisions are different for 24 APs and 60 STAs: (a) UDP; (b) TCP.

3% for 8 APs to 68% for 32 APs. For TCP traffic shown in Figure 3(f), the TCP throughput achieved by eTP_n also increases almost linearly, and the gain over $rxpwr$ grows with AP density from 2% for 8 APs to 52% for 32 APs.

So far, we have examined average throughput over all valid simulation trials, which is not a representative of the throughput distribution. As such, it is informative to examine throughput gain only for scenarios where the AP selection by our proposed method is different from that of $rxpwr$. This is shown in Figure 4 where we plot the histogram of the logarithm of the ratio of eTP_n over $rxpwr$ throughput of the joining STA for trials in which their selections are different, for 24 APs and 60 STAs with UDP and TCP, respectively. In UDP case, eTP_n achieves higher (lower) throughput compared to $rxpwr$ in 37% (5%) of the trials. The selections are the same for the remaining 57% of trials, and hence both algorithms result in identical throughput. In TCP case, eTP_n achieves higher (lower) throughput compared to $rxpwr$ in 34% (7%) of the trials. In the remaining 59% of trials they select the same AP. The histograms for other topologies show similar trends. For all simulated topologies, the number of trials in which eTP_n achieves throughput gain over $rxpwr$ is larger than those with throughput loss.

V. Conclusion

In this paper we proposed a class of AP selection algorithms which consider TXOPs, inter-BSS interference and collisions. This is achieved by exploiting BI signals both at the AP and the joining STA, as well as additional information such as the number of associated STAs, sum of inverse of MAC rates, and average waiting and idle times.

Our proposed AP selection algorithms can reduce the percentage of non-optimal selections and improve the average throughput of the joining STA in all tested scenarios for saturated UDP and TCP DL traffic. In particular, eTP_n achieves overall the best performance among all proposed algorithms for UDP and TCP traffic, and outperforms $rxpwr$, Load and Fukuda as the node density increases, be it AP density, STA density, or both. For a random topology with 24 APs and 60 STAs, the average throughput gain achieved by eTP_n as compared to $rxpwr$ is 42% and 24% for UDP and TCP, respectively. In this topology, eTP_n achieves as much as 90% and 93% of the optimal throughput for UDP and TCP, respectively. It is interesting to observe that throughput gain can be achieved by adding more APs and performing proper AP selection, even if the APs are on the same channel.

Future work includes verifying performance of $eTPn$ in mixed UDP and TCP traffic simulations, extending the current work to AP selection for UL traffic, examining the impact of our AP selection algorithm on aggregate network throughput, and extending current static algorithm to dynamic AP selection in which existing STAs can switch from one AP to another.

References

[1] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved Access Point Selection," in *MobiSys*, 2006.

[2] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley, "Facilitating Access Point Selection in IEEE 802.11 Wireless Networks," in *ACM SIGCOMM Conference on Internet Measurement*, 2005.

[3] K. Sundaresan and K. Papagiannaki, "The Need for Cross-layer Information in Access Point Selection Algorithms," in *ACM SIGCOMM Conference on Internet Measurement*, 2006.

[4] M. Abusubaih, J. Gross, S. Wiethoelter, and A. Wolisz, "On Access Point Selection in IEEE 802.11 Wireless Local Area Networks," in *IEEE Conference on Local Computer Networks*, Nov 2006.

[5] S. Miyata, T. Murase, and K. Yamaoka, "Characteristic analysis of an access-point selection for user throughput and system optimization based on user cooperative moving," in *IEEE CCNC*, Jan 2012.

[6] L. Luo, D. Raychaudhuri, H. Liu, M. Wu, and D. Li, "Improving End-to-End Performance of Wireless Mesh Networks through Smart Association," in *IEEE WCNC 2008*, Mar 2008.

[7] G. Judd and P. Steenkiste, "Fixing 802. 11 access point selection," *ACM SIGCOMM Computer Communication Review*, 2002.

[8] Y. Fukuda and Y. Oie, "Decentralized access point selection architecture for wireless lans," *IEICE transactions on communications*, vol. 90, no. 9, pp. 2513–2523, 2007.

[9] J. C. Chen, T. C. Chen, T. Zhang, and E. van den Berg, "Effective AP Selection and Load Balancing in IEEE 802.11 Wireless LANs," in *IEEE GLOBECOM*, Dec 2006.

[10] P. Bahl, M. Hajiaghayi, K. Jain, S. Mirrokni, L. Qiu, and A. Saberi, "Cell Breathing in Wireless LANs: Algorithms and Evaluation," *Mobile Computing, IEEE Transactions on*, Feb 2007.

[11] Y. Fukuda, M. Honjo, and Y. Oie, "Development of Access Point Selection Architecture with Avoiding Interference for WLANs," in *IEEE PIMRC*, Mar 2006.

[12] L. Du, Y. Bai, and L. Chen, "Access Point Selection Strategy for Large-scale Wireless Local Area Networks," in *IEEE WCNC*, Mar 2007.

[13] M. Abusubaih and A. Wolisz, "Interference-Aware Decentralized Access Point Selection Policy for Multi-Rate IEEE 802.11," in *IEEE PIMRC*, Sep 2008.

[14] J. W. Jang, Y. S. Lim, and C. K. Kim, "Traffic-aware decentralized AP selection for multi-rate in WLANs," in *IEEE ICACT*, Feb 2010.

[15] Y. Bejerano, S. jae Han, and L. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control," *IEEE/ACM Trans. on Networking*, vol. 15, no. 3, pp. 560–573, June 2007.

[16] H. Gong, K. Nahm, and J. Kim, "Distributed Fair Access Point Selection for Multi-Rate IEEE 802.11 WLANs," in *IEEE CCNC*, Jan 2008.

[17] W. Zhou and D. Qiao, "Fulfillment-Based Fairness: A New Fairness Notion for Multi-AP Wireless Hotspots," in *IEEE ICC*, June 2007.

[18] F. Xu, X. Zhu, C. C. Tan, Q. Li, G. Yan, and J. Wu, "Smartassoc: Decentralized access point selection algorithm to improve throughput," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, Dec. 2013.

[19] M. N. Krishnan, S. Pollin, and A. Zakhor, "Local Estimation of Probabilities of Direct and Staggered Collisions in 802.11 WLANs," in *IEEE GLOBECOM*, Dec 2009.

[20] M. N. Krishnan and A. Zakhor, "Throughput Improvement in 802.11 WLANs Using Collision Probability Estimates in Link Adaptation," in *IEEE WCNC*, Apr 2010.

[21] S. Yang, "Access Point Selection for Multi-Rate IEEE 802.11 Wireless LANs," *M.S. Thesis, University of California, Berkeley*, 2014.