Opportunistic Joint Uplink/Downlink Scheduling in WLANs *

Technical Report UIUCDCS-R-2006-2757

Joon Yoo[†‡], Haiyun Luo[†] and Chong-kwon Kim[‡]
University of Illinois at Urbana-Champaign[†]
Seoul National University[‡]
{joonyoo2,haiyun}@uiuc.edu, ckim@popeye.snu.ac.kr

ABSTRACT

Recent advances in the speed of multi-rate wireless local area networks (WLANs) and the proliferation of WLAN devices have made rate adaptive, opportunistic scheduling critical for throughput optimization. As WLAN traffic evolves to be more symmetric due to the emerging new applications such as VoWLAN, collaborative download, and peer-to-peer file sharing, opportunistic scheduling at the downlink becomes insufficient for optimized utilization of the single shared wireless channel. Furthermore, without proper scheduling on the uplink, the downlink throughput gain diminishes proportionally to the increasing number of clients that are actively transmitting on the uplink. However, opportunistic scheduling on the uplink of a WLAN is challenging because wireless channel condition is dynamic and asymmetric. Each transmitting client has to probe the access point to maintain the updated channel conditions at the access point. Moreover, the scheduling decisions must be coordinated at all clients for consistency. This paper presents JUDS, a joint uplink/downlink opportunistic scheduling for WLANs. Through synergistic integration of both the uplink and the downlink scheduling, JUDS maximizes channel diversity at significantly reduced scheduling overhead. It also enforces fair channel sharing between the downlink and uplink traffic. Through analysis and extensive QualNet simulations, we show that JUDS improves the overall throughput by up to 127% and achieves close-to-perfect fairness between uplink and downlink traffic.

1. INTRODUCTION

Rapidly increasing speed of wireless local area networks (WLANs) [1] and the dynamics of channel conditions in unlicensed frequency bands have made rate adaptation a critical component for high-performance wireless networking. Based on the channel condition, e.g., the instantaneous signal-to-noise ratio (SNR), an IEEE 802.11a/b/g [1] [2] WLAN interface can choose among 4 to 8 different rates, ranging from 1Mbps to 54Mbps. The coming new standard 802.11n [3] will offer even more available rates at a larger range, enabling fine grained channel rate control for optimized utilization of the wireless channel. Meanwhile, the increasing number of wireless users has reached the point where a wireless access point always has a choice when scheduling the traffic for multiple wireless clients. As a result, *opportunistic scheduling*, a technique that was first developed in cellular wireless networks [11] [12], has been recently applied to the downlink of a multi-rate, multi-user WLAN [13]. Opportunistic scheduling leverages multi-user diversity by scheduling the user whose instantaneous channel condition is above the average.

However, the existing work on WLAN opportunistic scheduling [13] is limited on the downlink only. This limit seriously bounds the utilization of the shared wireless channel for the following two reasons. First, while traditional Web traffic is asymmetric with a significant bias on the downlink, modern WLAN traffic is becoming more and more symmetric [15] [16] due to the emerging new network applications such as BitTorrent [14], peer-to-peer file sharing, and Voice over WLAN (VoWLAN). Since both downlink and uplink traffic shares a single wireless channel in a WLAN, poor channel unitization at the uplink directly impacts the achievable throughput at the downlink. Second, the dominating IEEE 802.11 MAC, i.e., distributed coordination function (DCF) [1], treats all senders, including both the access point and the clients, equally. Therefore, as the number of clients competing on the uplink for the shared channel increases, the bandwidth allocation for the downlink (or the access point) decreases proportionally. Without proper scheduling on the uplink, the achievable throughput gain due to downlink opportunistic scheduling *diminishes* as the number of transmitting clients served by an access point increases. This is in stark contrast to the expectation that opportunistic scheduling achieves higher throughput gain at higher multi-user diversity.

One major challenge of applying opportunistic scheduling at WLANs is the maintenance of the highly dynamic and asymmetric [18] condition of wireless channels defined in the unlicensed frequency bands. Different from the cellular wireless networks where efficient mechanisms for fine-grained channel condition feedback are built into the high-end hardware, the overhead of channel condition maintenance in a WLAN based on low-end wireless transceivers may be prohibitively high. For example, the evaluation of the opportunistic WLAN downlink scheduling [13] shows that the achievable throughput decreases with more than 3 clients involved, due to the excessive overhead of channel probing and feedback. This problem is further aggravated when a naïve opportunistic uplink scheduling is applied, since every wireless client has to probe the access point and the

*The research was supported in part by the Brain Korea 21 Project in 2005

access point has to feedback the best rate for each probing client. Furthermore, even with such updated channel conditions established, it is very difficult for all wireless clients as well as the access point to reach the consensus regarding the transmission schedule of every packet originated at different transmitters.

In this paper we present *JUDS*, the Joint Uplink and Downlink Opportunistic Scheduling for WLANs. JUDS synergistically integrates the opportunistic scheduling at both the uplink and the downlink for maximum channel utilization with significantly reduced overhead in channel condition maintenance and scheduling coordination. In more specific, JUDS doubles the channel diversity while reducing the overall scheduling overhead by half, compared with the scenario where downlink or uplink opportunistic scheduling is conducted alone. Furthermore, JUDS enforces fair bandwidth allocation between the uplink and downlink, regardless of the number of contending wireless clients. This bandwidth allocation policy ensures that the performance gain of opportunistic scheduling increases as the number of clients competing for the shared channel increases, consistent with the expectation.

We make three key contributions in this paper. First, we reveal the fundamental limit of the downlink opportunistic scheduling in a WLAN, and present the first WLAN joint uplink/downlink opportunistic scheduling. Our design leverages the maximum diversity of the shared wireless channel in a multi-user WLAN. Second, we present the details of the jointly uplink/downlink scheduling protocols. JUDS protocols exploit the unique characteristics of the broadcast wireless channel and the repetitive operating cycles of the scheduling algorithm. As a result, JUDS implementation eliminates a number of unnecessary signaling in popular CSMA/CA wireless MAC, e.g., the per-frame acknowledgements. Although these signaling messages are generally small, the constant per-frame PHY and MAC overhead plus the mandatory inter-frame spacing leads to an overhead up to 15$\sim$70% [1] [2]. Finally, the performance of JUDS has been evaluated through both analysis and simulations in QualNet [20]. Our evaluation shows that the unitization of the shared wireless channel is almost doubled and the fairness between uplink and downlink traffic is close-to-perfect.

The rest of the paper is organized as follows. In Section 2 we will review and compare with the related work. Section 3 gives an overview of JUDS design. Section 4 provides the detailed description of the probing stage of JUDS and Section 5 shows the scheduling and data transmission stage of JUDS. Section 6 describes the performance of the protocol through analysis, and Section 7 shows the performance via simulations. We finally conclude this paper in Section 8.

2. RELATED WORK

Many MAC protocols have been proposed [4] [5] [6] to exploit the multi-rate capability at the physical layer. The Auto Rate Fallback (ARF) [4] is the one typically implemented in commercial 802.11 products. ARF chooses to raise or lower its transmission rate according to consecutive transmission successes or failures, respectively. In the Receiver Based Auto Rate (RBAR) [5], the receiver selects an adequate transmission rate according to the channel quality measured from the received request-to-send (RTS) frame. It then piggybacks the selected data rate in the CTS frame. The Opportunistic Auto Rate (OAR) [6] improves the channel unitization by allowing a client to hold the wireless channel for an extended period when the achievable data rate is high. Those designs enable intelligent rate adaptation between a specific pair of sender and receiver. They do not explicitly leverage the channel diversity due to multiple clients.

Opportunistic scheduling optimizes the utilization of the wireless channel shared among multiple users. The higher the user diversity, represented by a larger number of users and a larger channel variation, the higher the potential performance gain. Opportunistic scheduling was designed and applied in cellular wireless data networks, e.g., HDR [11] [12]. In HDR, the channel conditions are measured by the pilot signal sent by the base station to each individual user. The users then feed back the channel condition simultaneously via the CDMA uplink. In a WLAN based on low-end wireless transceivers, the lack of efficient support for closed-loop channel condition feedback represents the main challenge for opportunistic scheduling. The Medium Access Diversity (MAD) protocol [13] applies opportunistic scheduling on an IEEE 802.11 WLAN downlink and is backward compatible with the legacy 802.11 DCF [1]. However, since the wireless channel is shared between the uplink and downlink traffic, opportunistic scheduling on the downlink alone is not enough as the WLAN traffic becomes more and more symmetric.

The opportunistic scheduling proposed in the literature [11] [12] [13] and in this paper are all based on signal-to-noise ratio (SNR) measurements to determine the appropriate data rate. Recent measurements [25] using off-the-shelf 802.11 devices showed that the SNR, *measured as an average over many packets*, may not be a good predictive tool for the successful delivery of a packet. Note that the measurement results do not contradict or invalidate our approach in that our rate adaptation is based on *instantaneous* SNR feedback. In fact, one contribution of this work is the design of efficient mechanisms for timely channel condition feedback, based on which the opportunistic scheduling runs.

3. OVERVIEW

In this section we overview JUDS operation in a WLAN. We first elaborate the challenges and issues, and then present the basic ideas and mechanisms that JUDS employs to address those challenges.

Table 1 shows the stages for downlink and uplink opportunistic scheduling, if conducted separately. For downlink opportunistic scheduling, the access point first broadcasts a probe message. The probe message also specifies a list of candidate clients. Each candidate client then measure the signal-to-noise ratio of the probe broadcast as its downlink channel quality. They then feed back the measurement to the access point one after the other, in the order that specified in the access point's probe broadcast. Once the access point collects the downlink channel quality of all candidate clients, the access point runs the scheduling algorithm (e.g., proportional fairness scheduling), and transmits to the scheduled client. Opportunistic uplink scheduling is more complex. First, all clients that are competing for the

Stage	Downlink Scheduling	Uplink Scheduling
Candidate Selection	AP specifies	Contention or AP specifies
Channel Probing	AP broadcasts Probe	***Clients probe AP***
Probing Feedback	***Clients feedback to AP***	AP broadcasts to clients
Scheduling & Tx	AP tx to client	AP selects client

Table 1: Opportunistic downlink/uplink scheduling

uplink must probe the access point. This process can be scheduled by distributed contention or by the access point. The access point then measures uplink channel quality, runs the scheduling algorithm, and notifies the scheduled client.

The major challenge of WLAN opportunistic scheduling is the overhead for the senders to collect from the receivers the updated channel condition at the receivers' sides. For downlink scheduling all candidate clients have to feed back their measured downlink channel quality to the access point. The communication overhead is therefore O(k), given k downlink candidate clients chosen by the access point. For uplink scheduling all candidate clients have to probe the access point individually for the access point to assess the uplink channel quality. The complexity is again O(k) given k uplink candidates.

Note that the O(k) probing or feedback overhead is severe in an asynchronous, packet-switched WLAN, where a constant physical and MAC layer overhead is incurred for every frame regardless of the frame size or channel rate. For example, in the context of IEEE 802.11b with 2Mbps basic rate and 11Mbps data rate, the physical and MAC layer overhead accounts for 31% for a frame of 1500 bytes. For higher speed IEEE 802.11a/g with 6Mbps basic rate and 54Mbps data rate, the overhead increases to 68%. To limit the overhead *it is therefore more effective to decrease the number of frames, as opposed to reducing the frame size.* Although a small number of candidates k lowers the feedback or probing overhead, it also limits the multi-user diversity or the throughput gain of opportunistic scheduling.

JUDS addresses this challenge by the following three mechanisms. First, since JUDS schedules both downlink and uplink traffic simultaneously, it is able to combine the clients' feedback of its downlink channel condition with client's probing for its uplink channel condition. Therefore, the feedback and probing overhead is cut by half, without compromising multi-user diversity. Second, JUDS exploits the cyclic scheduling between the uplink and downlink traffic, and piggybacks control signals whenever possible. In specific, because the access point and one of the clients alternate in transmitting on the shared wireless channel, the mandatory per-frame acknowledgement can be piggybacked on the next frame transmission from the other direction. This mechanism removes the 70.1% physical layer overhead and the inter-frame spacing if the acknowledgement were transmitted in a separate frame. Finally, JUDS exploits the broadcast nature of wireless transmission and enables *opportunistic probing and feedback*. For downlink opportunistic scheduling a candidate client can intentionally choose not to feedback its downlink channel and leaves it scheduled feedback slot idle, if its downlink channel quality is too low. Other clients that detect the idle slot will automatically advance the feedback schedule, therefore further saving the

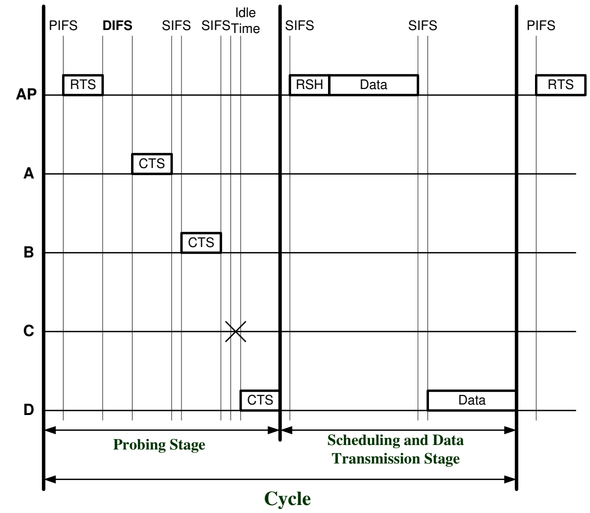


Figure 1: JUDS operation.

feedback overhead.

Putting it all together, the operation of JUDS can be illustrated by Figure 1. JUDS divides the time into repetitive *cycles*, and schedule exactly one downlink transmission and one uplink transmission, if the demand exists, in each cycle. In specific, a cycle consists of two stages: the *channel probing stage* and the *scheduling and data transmission stage*. In the probing stage, the access point and the clients coordinate to probe for channel quality measurement at *both* downlink and uplink. In the scheduling and data transmission stage, based on the channel quality measurements and the scheduling priority, the access point selects the client that transmits on the uplink and the client that receives on the downlink. Once the schedule is determined, the access point sends a downlink data frame to the scheduled downlink client. The scheduled uplink client then transmits afterwards. We present the details of those two stages in the next two sections respectively.

4. PROBING STAGE

The access point and selected candidate clients exchange uplink/downlink channel conditions in the probing stage. In this section, we first describe the two basic steps: candidate selection and downlink probing in Section 4.1 and the downlink condition feedback and uplink probing in Section 4.2. We then describe in Section 4.3 how the access point is made aware of the clients that are competing for the uplink.

4.1 Candidate selection and downlink probing

The first step that the access point takes in the *probing stage*, prior to sending the RTS, is to select the clients that are either to be probed for the downlink or to probe for the uplink channel condition. The access point chooses those candidates *randomly* for long-term fairness. Figure 1 shows an example of the candidate selection. The access point has full knowledge of those clients who are receiving packets from the downlink, i.e., client A, B, C and E in this example. Through mechanisms presented in Section 4.3 the access point is also made aware of client A, D and F who are competing on the uplink (Clients E and F are not shown in Figure 1.). The access point then selects up to k candidates from clients A-F. In this example, we assume that $k = 4$. Suppose the access point selects client A, B, C, and D. A, B, and C will be competing for the downlink, while A, D will be competing for the uplink.

The selected candidate nodes are listed in the *destination address (DA) list field* of the RTS frame, as shown in Fig. 2. The RTS frame is then broadcasted to all clients. A straightforward method of listing the clients in the *DA list* field in the RTS frame would be to just list the clients' unique 48 bit MAC addresses in the sequential order for CTS transmissions. Since this could result in a large overhead we utilize the *Association ID (AID)* – a 16-bit ID of a client assigned by the access point during the association procedure [1].

AID is effective to identify the list of *destination addresses (DAs)* in the RTS frame. We assume that the access point assigns the smallest available AID to a new client during the association procedure. We only use the 8 least significant bits, or the least significant byte of the AID for the addressing. As shown in Fig. 2, each destination address in the list corresponds to the the least significant byte of the AID. Therefore the default value of the *Address length* subfield in the *Destination List Information* field will be set to 8, enough for accommodating $2^8 = 256$ clients. Since a single access point is unlikely associated with more than 256 clients in realistic scenarios, this method will be generally feasible. In case there are more than 256 clients, the address length will be changed by setting the *Address length* subfield appropriately.

The total number of clients, as well as the number of clients competing for uplink and the number of clients competing for downlink will be recorded in the *Destination List Information* field. For example, if there are a total number of 10 clients and the number of uplink and downlink clients is 3 and 8 respectively, then there is 1 client that has both uplink and downlink frames. The first 2 clients in the list are therefore clients for uplink, the next 1 will the client with both uplink and downlink, and the other 7 will be the clients competing only for the downlink.

4.2 Downlink condition feedback and uplink probing

On receiving the RTS frame, the clients first check if it is included in the *DA list* field. If it is, the client prepares to send a CTS back to the access point. The format of the CTS frame is similar to that of 802.11 [1], but contains the extra *schedule information* field as illustrated in Figure 3. Note that all clients measure their downlink quality even if they are not listed in the *DA list* field for opportunistic probing and feedback explained later in this subsection.

The CTS frame serves as either a *downlink feedback* or an *uplink probe* frame. As for a *downlink feedback*, it reports the feasible data rate in the 4 bit *Rate* field of the CTS frame, based on the measured downlink channel quality when receiving the RTS frame. As for an *uplink probe*, it sets the *Rate* field 0.

The clients then send CTS in the order indicated in the *DA*

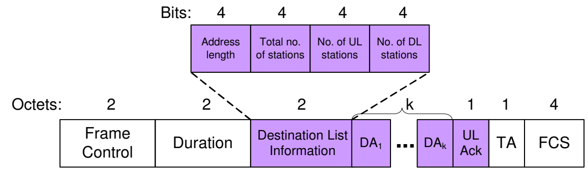


Figure 2: Modified RTS frame format for downlink probing and uplink candidate selection.

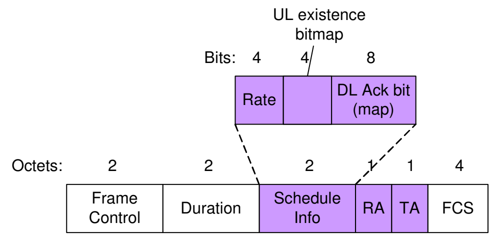


Figure 3: Modified CTS frame format for downlink feedback and uplink probing.

list field in the RTS frame. As the CTS uses the basic data rate and the clients know the order of the CTS candidates, each client can easily calculate the transmission schedule and their own slot. The first client on the list waits for DCF inter frame space (DIFS), or a period of [busy time + 1 idle slot], and sends its CTS back to the access point (See Figure 1). There are two reasons why it specifically waits for a DIFS period or a [busy time + 1 idle slot]. One is to allow clients to contend for the uplink (See Section 4.3). The other is to prevent neighboring clients that use IEEE 802.11 DCF from interfering. If the waiting period is larger that DIFS, then neighboring clients that use IEEE 802.11 DCF may dominate the channel.

In case a candidate client fails to receive the RTS frame, an idle time slot instead of an entire idle CTS duration will be wasted. When a client that is awaiting to transmit a CTS senses the channel as idle for a time slot, it realizes that another candidate has missed the turn. The client then advances its transmission starting time accordingly. For example in Figure 1, client C fails to receive the RTS. As for client D, by sensing the channel idle for a time slot after B's CTS and the SIFS duration, it will assume that client C's turn is over and transmit right after the idle time slot. This will reduce the idle time by 79% when using IEEE 802.11a [2]. Note that the above mechanism works even in the scenario of hidden/expsed terminals. A client only needs to configure its carrier sense threshold appropriately to sense the CTS transmission, as opposed to decoding the CTS frame for channel status assessment [17]. Moreover, in case where there are only downlink packets, i.e. no uplink packets, a client decides not to transmit a feedback when the current channel quality is relatively poor. By using this method, we can reduce the probing overhead when only downlink traffic occurs. We will discuss this further in the next section.

We have been exploring the broadcast nature of wireless communications to suppress the downlink feedback as described above. To further minimize the overhead of downlink feedback, a client can choose not to respond to the

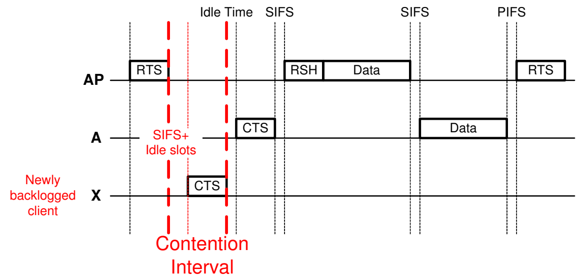


Figure 4: A *newly backlogged* client will contend in the *Contention Interval*.

access point's probe, if its downlink channel condition is well below the average and therefore the chance of being selected for transmission on the downlink by the proportional fairness scheduler (Section 5.1) is low. Note that the access point will receive no downlink condition feedback if the downlink channel quality for all probed clients is low. In this case the access point can randomly schedule a client.

4.3 Uplink contention

In order for the access point to select uplink probe candidates, a client uses the 4-bit *UL existence bitmap* in the CTS (See Figure 3) frame to inform the access point the uplink packet queue status. The first bit represents the existence of a packet in the uplink packet queue, and the next 3 bits represents the length of the queue. For example, if the first bit is set to "1" (true), and the next 3 bits are set to "3", then there are a total number of 4 packets uplink. The access point uses this information to determine the existence of the uplink packets for the corresponding client.

The access point can be informed of a contending client if the client is probed for the downlink or selected as the uplink candidate. If the client does not have a packet in its uplink buffer and there are no downlink packets destined for the client, then the client has to contend to inform the access point. The contention process is also necessary for a client first arriving. We call this client *newly backlogged* and the contention duration the *Contention Interval* as shown in Figure 4.

The newly backlogged clients contending for the uplink will send a packet after randomly selecting a backoff of contention window size CW, which is similar to the backoff procedure in IEEE 802.11 DCF [1]. In each Contention Interval, only one contenting client is allowed to transmit. After a contending client finishes transmitting, the first client listed in the *DA list* will send the CTS after an idle time slot. Since the *Contention Interval* consists of only [SIFS + 2 idle slots], a newly backlogged client may need to wait multiple cycles when there exist multiple competing clients. Our analysis in Section 6.2 show that the *expected waiting time* of a newly backlogged client is around 35 ms even with 30 newly backlogged clients contending simultaneously.

5. SCHEDULING AND DATA TRANSMISSION STAGE

In this section we describe the scheduling and data transmission of JUDS. Note that after *probing stage*, the access point collects not only the downlink channel conditions of all candidate clients, but also their uplink channel conditions. The access point then performs opportunistic scheduling for both the downlink and uplink (Section 5.1). The access point first transmits a data frame to the client scheduled on the downlink. The identifier of the client that is scheduled for uplink transmission is piggybacked in the downlink data transmission, together with the feedback of the uplink channe condition and the highest data rate supported. The selected client for uplink overhears the downlink data transmission, and decodes the piggybacked uplink feedback. It then transmits on the uplink afterwards (Section 5.2). We finally present the mechanism that removes the explicit per-frame acknowledgement (Section 5.3).

5.1 Opportunistic Scheduling and Fairness Model

After the access point receives all the downlink feedback and uplink probes, it schedules two clients for transmissions on the downlink and uplink respectively. In JUDS the access point enforces *proportional fairness* [8] [9] [10] by default. That is, the access point assigns each client a priority according to:

$$Priority_k = \frac{DR_k(t)}{T_k(t)} \tag{1}$$

The priority of client k is determined by the instantaneous data rate ($DR_k(t)$) over the average throughput ($T_k(t)$). Similar to the HDR [11] [12] system, the access point monitors the throughput of each client in a recent time window. Among the probed clients with backlogged downlink queues, the access point selects the one with the highest priority, defined in Eqn. (1), and schedules the client for downlink transmission. Similarly, the access point selects among the probed clients with backlogged uplink queues the one with the highest priority, and piggybacks the identifier and the supported data rate (according to the measured uplink channel condition at the access point) to the downlink data transmission.

Note that the fairness model advocated by the current JUDS design is composed of two parts. One is the *proportional fairness* among clients competing for the uplink and among the clients competing for the downlink, the same as that achieved when the two sets of clients are scheduled separately. The other is the throughput fairness between uplink and downlink, since the same amount of traffic is transmitted on both directions during each cycle. There are other alternate fairness models that are applicable, e.g., per-flow proportional fairness on all uplink/downlink flows [7] and temporal fairness between uplink and downlink traffic [6]. We leave it as a future work to study the impact of different fairness models on the performance of typical WLAN applications.

5.2 Data Transmission

In order for the uplink scheduling information be piggybacked in the downlink data transmission, two receiver addresses, i.e., the *downlink receiver (DL RA)* and the *uplink transmitter* address *(UL TA)*, and the supported data rate for uplink transmission (*UL Rate*) are defined in the data header, as shown in the data frame format in Figure 5. The *DL RA* identifies the destination client of the downlink data transmission, while the *UL TA* identifies the client that is

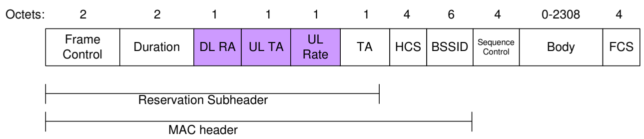


Figure 5: Data frame format

scheduled for uplink transmission in this cycle. The frame format is used in both downlink/uplink data transmissions. The *UL TA* and *UL Rate* field is always set to 0 for the uplink data frame.

Note that both *DL RA* and *UL TA* are transmitted at the basic rate regardless of the data rate at which the data is transmitted. Since the basic rate is supported by all clients, it is guaranteed that all clients scheduled for uplink transmission can decode the piggybacked scheduling information from downlink transmissions. This technique is named the *reservation subheader (RSH)* [5]. Note that due to wireless channel asymmetry, a client with good uplink channel condition (and therefore is scheduled in UL TA) may have a bad downlink channel condition. If RSH is encoded at higher rates a scheduled uplink client may not be able to decode it from the downlink.

If no downlink traffic exists the access point can simply transmit a frame with zero payload. This frame will inform the client that is chosen for uplink transmission. In case an uplink frame does not exist or the client scheduled to transmit the uplink data fails to receive the RSH, an idle duration of [SIFS + time slot] will occur. Similar to the scenario when a transmission on the uplink is missed (see Section 4.2), the access point moves on to the next cycle immediately after an idle duration is detected.

A final note is that depending on the channel coherence time, JUDS can be easily extended to pack multiple data frames in one cycle, similar to the opportunistic scheduling developed in [3,6,13,23]. We show in this paper that JUDS achieves significant throughput improvement even with one single frame transmitted over the uplink and/or downlink during each cycle and therefore impose the weakest assumption on the channel coherence time.

5.3 Implicit acknowledgement

JUDS exploits the cyclic scheduling between downlink and uplink to eliminate explicit per-frame MAC-layer acknowledgement for throughput optimization. The idea is to piggyback the acknowledgement into the downlink/uplink probing and feedback frames during the probing stage of the next cycle. In specific, we piggyback the acknowledgement at *ACK bit(map)* in the RTS and CTS (See Figure 2 and 3).

For the uplink data frame, the access point will piggyback the acknowledgment information in the *ACK bit(map)* of the RTS in the next cycle. So there will be no difference between the piggybacked acknowledgement and the ACK defined in IEEE 802.11 DCF: every uplink data frame is followed by an immediate acknowledgement piggybacked in the RTS at the beginning of the next cycle.

For timely acknowledgement of the downlink data transmission, the access point always adds the client that receives the downlink data frame in the current cycle into the list of candidate clients for the next cycle, even if the client does not have any other downlink/uplink frame or the client is not randomly chosen as a candidate for the next cycle. This way, the access point is guaranteed to receive an implicit acknowledgement from the client, piggybacked in the CTS frame in the probing stage of the next cycle. However, if the client is indeed not randomly chosen as a candidate for the next cycle, the access point will not schedule the client for transmission on either downlink or uplink to avoid skewed proportional fairness.

6. PERFORMANCE ANALYSIS

In this section, we first analyze the fundamental limit of downlink opportunistic scheduling and demonstrate JUDS's performance gain. We show that with downlink scheduling alone the throughput gain diminishes as the number of transmitting clients and the load on the uplink increase. We then appraise the scalability of JUDS's performance gain to the increasing number of active clients, i.e., those engaged in uplink and/or downlink transmissions. Finally, we characterize the relationship between throughput gain and the number of probed candidates (Section 6.1). We also show that even with 30 newly backlogged clients[1] the expected waiting time is around 35ms, falling within the tolerance threshold of most real-time applications such as voice (Section 6.2).

6.1 System Throughput

In this subsection, we compare JUDS with RBAR [5] and MAD [13] in throughput. RBAR is a rate adaptation protocol that does not perform scheduling, while MAD applies opportunistic scheduling on the WLAN downlink. To obtain the throughput of RBAR, MAD and JUDS, we first derive the achievable raw data rate through opportunistic scheduling and then take the wireless physical layer and MAC overheads into account.

We assume clients are randomly distributed and the access point is located in the center of a cell of radius D. Data payload size is fixed to L_{Data}. We also assume the wireless transceiver supports up to M discrete data rates: $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_M$. A data rate $\mathcal{R}_i$ is feasible when the current SNR is above the threshold γ_i. γ_{M+1} is assumed to be ∞ [13]. The instantaneous SNR for a receiver at time t_d can be determined by [6],

$$\gamma(t_d) = \mathcal{P}_t \cdot d(t_d)^{-\beta} \cdot \frac{\rho(t_d)}{\eta} \qquad (2)$$

where $\mathcal{P}_t$ is the transmit power, $d(t_d)$ is the distance between sender/receiver at time t_d, β is the path loss exponent, η is the noise level and $\rho(t_d)$ is the average channel gain at time t_d. We assume that the transmit power $\mathcal{P}_t$ is fixed and the noise level η is constant over time for all clients. We also assume that the fast fading is defined by Rayleigh distribution [24] where the envelope of the signal follows the

[1]Note that a newly arrived client first associates with the access point. It then contends in the contention interval (Section 4.3) after it becomes backlogged.

probability distribution function (*pdf*) given by,

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \cdot e^{-\frac{r^2}{2\sigma^2}} & (0 \leq r \leq \infty) \\ 0 & (r < 0). \end{cases}$$

where σ^2 is time-average power of the received voltage signal before the envelope detection.

Assume that a number of k clients are probed for the channel for opportunistic scheduling. By Eqn. 2, the probability that the current SNR of the selected station at time t_d is smaller than a certain threshold γ_i when k clients are probed is given by,

$$\Pr(\gamma(t_d) < \gamma_i \mid k) = \Pr(\mathcal{P}_t \cdot d(t_d)^\beta \cdot \frac{\rho(t_d)}{\eta} < \gamma_i \mid k)$$
$$= \int_0^D \Pr(\rho(t_d) < \frac{\eta \cdot \gamma_i \cdot x^\beta}{\mathcal{P}_t} \mid k) \cdot \Pr(d(t_d) = x) dx \quad (3)$$

The *pdf* for $\Pr(d(t_d) = x)$ is $2x/D^2$ [26]. We assume that the gain factors unrelated to fading (such as antenna gain) are constant c. Since the channel is Rayleigh, Eqn. 3 yields,

$$\int_0^D (1 - e^{-\frac{\eta \cdot \gamma_i \cdot x^\beta}{c\mathcal{P}_t}})^k \cdot \frac{2x}{D^2} dx \equiv \pi(\gamma_i, k) \quad (4)$$

To successfully receive at a data rate $\mathcal{R}_i$, the current SNR should be $\gamma_i \leq \gamma(t_d) < \gamma_{i+1}$. So, the average achievable data rate can be obtained as follows,

$$\overline{\mathcal{R}}(k) = \sum_{i=1}^{M} \mathcal{R}_i \cdot \{\pi(\gamma_{i+1}, k) - \pi(\gamma_i, k)\} \quad (5)$$

Using Eqn. 5, we derive the system throughput for JUDS. Assume that there are k_{UL} clients probed for the uplink and k_{DL} for the downlink and k_{TOT} is the total number of probed clients. The throughput ($\mathcal{T}$) of JUDS is determined as follows,

$$\mathcal{T} = p_{ps} \cdot \frac{L_{DL} + L_{UL}}{t_{probing\_stage} + t_{DL} + t_{SIFS} + t_{UL}} \quad (6)$$

L_{DL} and L_{UL} are the size of the MAC data payload (L_{Data}) for the downlink and uplink, respectively. p_{ps} is the probability that the RTS packet will successfully be received by *at least* 1 client. As the control packets such as RTS and CTS are sent at the basic rate ($\mathcal{R}_1$), the RTS reception will succeed only if the SNR is larger than γ_1. Therefore, $p_{ps} = 1 - \pi(\gamma_1, k_{TOT})$ (see Eqn. 4). t_{SIFS} is the duration of short inter-frame space, SIFS [1], t_{DL} and t_{UL} are the time duration to send a downlink packet and an uplink packet respectively. $t_{probing\_stage}$ is the time duration for the probing stage where,

$$t_{probing\_stage} = t_{PIFS} + t_{RTS} + t_{DIFS} + k_{TOT} \times$$
$$\{t_{SIFS} + p_{fs} \cdot (t_{CTS}) + (1 - p_{fs}) \cdot (t_{slottime})\} \quad (7)$$

t_{PIFS} and t_{DIFS} are inter-frame space time for PIFS and DIFS respectively [1]. t_{RTS} and t_{CTS} are the time to send a RTS, and CTS packet respectively and k_{TOT} is the total number of probed clients. p_{fs} is the probability that a client will successfully receive a probe packet where $p_{fs} = 1 - \pi(\gamma_1, 1)$ (see Eqn. 4). Note that the CTS transmission time

N	CW_{opt}	P_{suc}	$E[\tau]$(ms)
2	6	0.326	1.30
5	14	0.113	4.93
10	27	0.054	10.97
15	40	0.036	17.00
20	54	0.027	23.03
25	67	0.021	29.06
30	80	0.018	35.09

Table 2: Expected waiting time for newly backlogged clients

is substituted by a time slot when a client fails to receive the RTS (See Section 4.2).

The total time duration spent on downlink (uplink) transmission is,

$$t_{DL(UL)} = \frac{L_{PHY}}{\mathcal{R}_1} + \frac{L_{MH} + L_{Data}}{\overline{\mathcal{R}}(k_{DL(UL)})} \quad (8)$$

where L_{PHY} is the physical header size and L_{MH} is the MAC header size of the data packet. $\overline{\mathcal{R}}(k)$ is the average data rate obtained from Eqn. 5. Combining Eqn. 5 to 8 we obtain the throughput for JUDS.

By Eqn. 5 and 6 we obtain the exemplary numerical results of throughput via the MATLAB software [21]. We use the 802.11a physical layer [2] specifications and the cell radius is set to $D = 200$m. Figure 6 shows the average throughput as a function of the number of uplink clients (N) served by the access point. We assume that the downlink is always backlogged at the access point, and k is set to 3 for both MAD and JUDS for fair comparison. As expected, the throughput gain of downlink opportunistic scheduling in MAD steadily declines as N increases, since bandwidth allocation at the downlink decreases, and therefore the effect of opportunistic scheduling diminishes. In contrast, JUDS consistently benefits from the increased level of multi-user diversity at the uplink. The slight throughput drop in JUDS, when $N = 1$, occurs because the opportunistic uplink scheduling will not take effect when there is only one client with uplink traffic.

Figure 7 presents the average achievable raw data rate of opportunistic scheduling and expected throughput for RBAR, MAD and JUDS as a function of the number of probed candidates (k). The raw data rate serves as an upper bound for the achievable throughput. The difference between the raw data rate and throughput of the protocols is due to the overhead at the physical layer, MAC layer, and scheduling. Note that the achievable raw data rate increases with k (or the multi-user diversity). The expected throughput of MAD decreases when more than 4 clients are probed due to the excessive probing and feedback overhead, consistent with the simulation results presented in [13]. In contrast, JUDS's increasing trend in throughput sustains up to 8 probed candidates. It performs the best due to reduced overhead and the increased level of multi-user diversity, as a result of the joint uplink and downlink opportunistic scheduling.

6.2 Waiting time for contention

In this subsection we derive the expected *waiting time* of a *newly backlogged* client, i.e., the period between the time

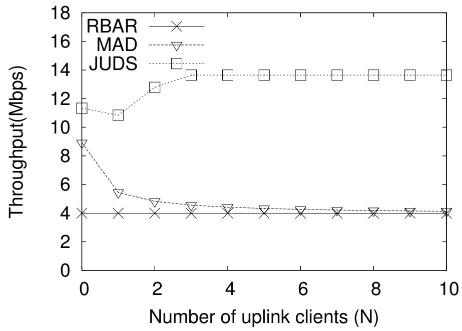


Figure 6: Average throughput vs. Number of uplink clients (N) while downlink is always backlogged

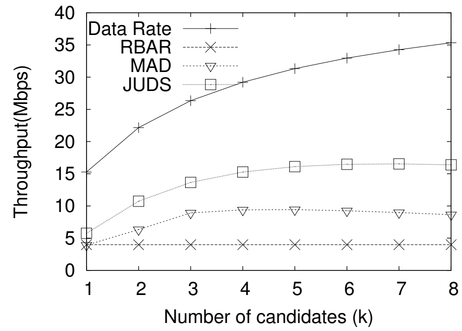


Figure 7: Average achievable raw data rate and the expected throughput vs. the number of probed candidates (k).

the client becomes backlogged and the time it succeeds in transmitting a CTS within the Contention Interval (see Section 4.3). Assume there are N newly backlogged clients and the backoff counter is set randomly in [0, CW-1], similar to similar to DCF's inter-frame random backoff. Therefore, the probability p of that a client will contend in any timeslot is $2/(CW-1)$.

As the length of the *Contention Interval* is DIFS [= SIFS + 2 time slots], a newly backlogged client can contend on one of the two time slots. For a newly backlogged client to successfully transmit a CTS within the Contention Interval, the client should either be the only one contending on the first time slot, or the only one contending on the second time slot in case the first time slot is idle. Therefore, the probability for a client to succeed in one Contention Interval is:

$$P_s = p(1-p)^{N-1}(1+(1-p)^N) \quad (9)$$

The expected waiting time $E[\tau]$ of a newly arriving client is,

$$E[\tau] = \sum_{i=0}^{\infty}(1-P_s)^i \cdot P_s \cdot i \cdot E[t_c] = \frac{1-P_s}{P_s} \cdot E[t_c] \quad (10)$$

$E[t_c]$ is the expected length of a cycle which is derived from the denominator of Eqn. 6. The expected waiting time $E[\tau]$ depends on the channel attempt probability p, and therefore is a function of the contention window size CW. So, we need to apply an optimal CW size to minimize $E[\tau]$. To obtain the optimal CW size, we apply the mechanisms proposed in [27] [28] by counting the number of idle slots. Table 2 shows the optimal CW size, success probability, and the expected waiting time as a function of the number of newly backlogged clients (N). Even with 30 newly backlogged clients, the expected waiting time is limited around 35.09ms, which is tolerable for most real time applications such as VoIP which requires a delay less than 100ms.

7. PERFORMANCE EVALUATION

We have implemented JUDS in the Qualnet Simulator [20]. We compare JUDS with ARF [4], RBAR [5] and MAD [13]. We use the IEEE 802.11a physical layer [2], which supports 8 variable data rates, ranging from 6 to 54Mbps. The physical layer parameters such as the transmission power and the receive sensitivity are adopted from the commercial Cisco Aironet 802.11a/b/g Wireless CardBus Adapter [19]. We model the wireless channel with free space Rayleigh Fading and Ricean Fading [24]. Most of the simulations are performed on Rayleigh Fading except when the channel stability was measured by varying the Ricean factor (Figure 10).

We use constant bit rate (CBR) traffic with 1kbyte payload size to control the traffic load intensity. The topology used was an infrastructure based WLAN, where all the clients communicate only with the access point. The access point maintains per client queue for opportunistic scheduling. There are one access point and 10 stationary clients in the simulations.

We conduct 5 sets of simulations. We first investigate the effect of the number of candidates in the probing stage with variable parameters. We then show the enhanced performance of JUDS compared to the other protocols for variable traffic loads. We study the performance as the number of active users increases and also vary the channel coherence time to study JUDS' performance at different channel dynamics. We finally demonstrate that JUDS achieve both close-to-perfect fairness between uplink and downlink traffic, and proportional fairness among clients competing for the uplink and downlink respectively.

Our evaluations show that JUDS is most efficient when the the clients have both downlink and uplink traffic, since a single control frame can carry both downlink and uplink control information. Even in the worst cases when the uplink and downlink clients are completely different or the traffic is extremely asymmetric (e.g., with downlink traffic only), JUDS still significantly outperforms separate uplink/downlink scheduling because of JUDS' protocol overhead reduction through piggybacked and implicit signaling.

7.1 Number of candidate clients

In this set of simulations we study the effect of the number of candidate clients (k) selected in the probing stage, when the packet size, distance between sender/receiver (d), and Ricean factor (K) [24] are varied. Overloaded traffic is given so that both uplink and downlink are always backlogged.

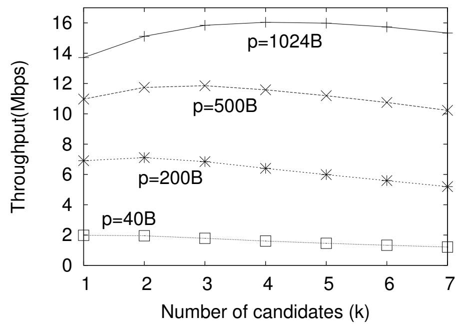


Figure 8: Network throughput vs. number of candidates with variable packet size

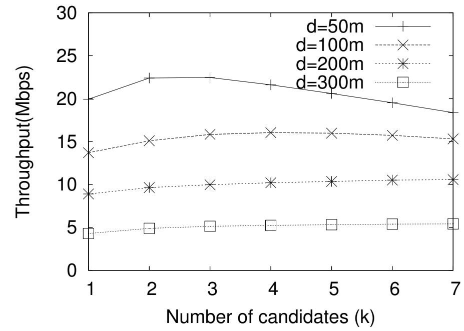


Figure 9: Network throughput vs. number of candidates with variable distance(d)

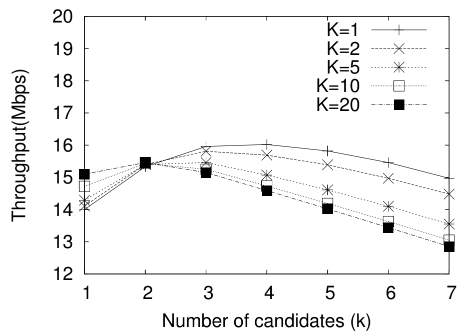


Figure 10: Network throughput vs. number of candidates with variable Ricean factor (K)

Figure 8 shows the throughput of JUDS as a function of k as the packet size p is varied. When p is large (e.g., p=1024 bytes), the throughput increases until more than k=4 candidate clients. Meanwhile, the throughput decreases monotonically with k when the packet is small (e.g., p=40 bytes). Note that the optimal throughput is quite robust to k given a certain packet size. As a result, an empirical choice of k=2 will lead to a throughput within 5.8% of the optimum for packet size ranging from 40 to 1024 bytes.

The distance between the access point and all the clients is set to d, where a small d means good average channel quality. Figure 9 shows the throughput as a function of k as d is varied. We can see that when d is small (50m), the throughput decreases as the number of candidates is more than 2. On the other hand, when d is relatively large, the throughput steadily increases with the number of candidates. The reason is that the control frames are transmitted at the basic rate while the data frames typically use a higher rate. The proportion of the scheduling overhead will therefore increase as the data rate for the data frames increases. Consequently, it is better to probe less clients when the average channel condition becomes better.

A smaller Ricean factor, K, represents a larger variance of the channel condition [24]. Figure 10 shows that as K increases it is better to probe less clients since the channel becomes more stable in terms of variance. This is due to the fact that the multiuser diversity increases with high channel dynamics [9] [10].

7.2 Throughput vs. traffic asymmetry

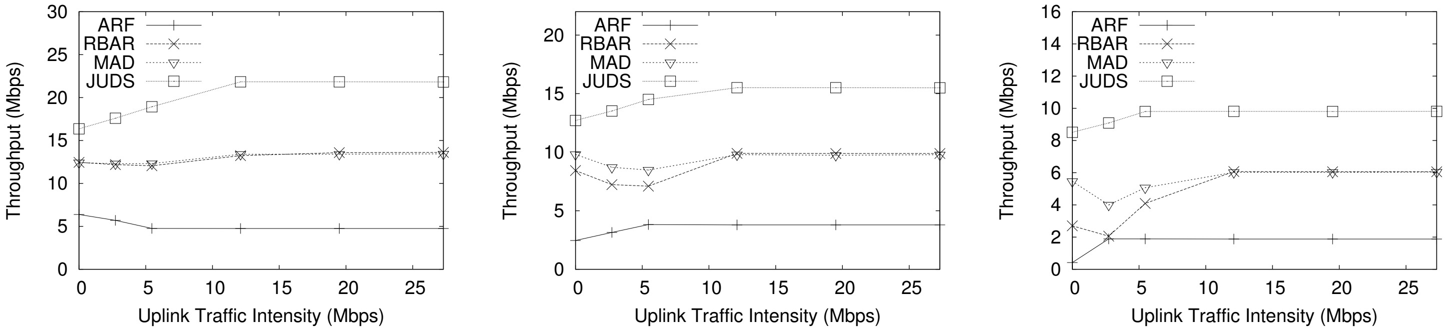
Figure 11 depicts the throughput of JUDS under different uplink/downlink traffic balances, and compare with those of ARF [4], RBAR [5] and MAD [13]. We fix the aggregate downlink traffic load to 54Mbps, and gradually increase the aggregate uplink traffic load from zero to 30Mbps (x-axis). Since MAD performs best when k is set to 3 [13], we set k to 3 for both MAD and JUDS for fair comparison. The distance between access point and all the clients d is varied from 50m to 200m.

RBAR and MAD clearly outperform ARF in every scenario due to the better rate adaptation to the fast fading channel. RBAR and MAD perform almost the same when the distance (d) is generally small, i.e., the average channel quality is relatively high due to the same reason explained in Section 7.1. MAD outperforms RBAR when the average channel quality gets lower, since the multiuser diversity on the downlink finally improves. But as the uplink traffic intensity gets higher, the number of clients competing for the shared channel increases. Therefore, the downlink is dominated by the uplink traffic and the performance gain of MAD's downlink opportunistic scheduling is lost.

As we expect, JUDS consistently outperforms MAD in throughput by up to 127% and 62% in average, especially when the uplink/downlink traffic is balanced. As the uplink traffic increases, JUDS takes advantage of the higher channel diversity due to uplink scheduling for the increased number of clients competing for the channel. Even in the worst case when there is only downlink traffic, JUDS achieves 30∼56% throughput gain over MAD due to its effectiveness in reducing the protocol overhead through implicit acknowledgement and downlink feedback suppression.

7.3 Throughput vs. number of active users

Figure 12 plots the network throughput as the number of active users increases. Each active user has both overloaded uplink and downlink traffic. When there is only one active user, opportunistic scheduling does not take effect. Therefore, MAD and RBAR performs about the same. As the number of active users N increases the bandwidth allocation at the downlink decreases. Consequently, the gain in channel utilization due to MAD's downlink opportunistic scheduling



(a) d =50m (b) d =100m (c) d =200m

Figure 11: The aggregate network throughput vs. the aggregate uplink traffic load. The distances between the access point and the all clients (d) is varied: (a) d=50m, (b) d=100m, (c) d=200m.

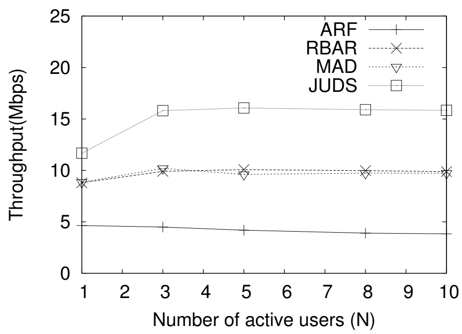


Figure 12: Average throughput vs. number of active users

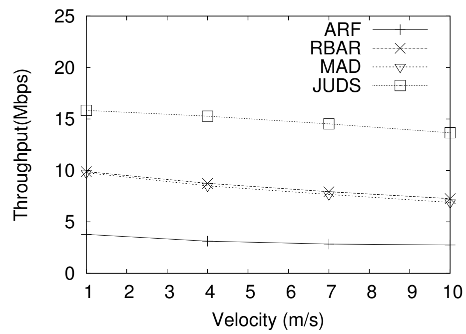


Figure 13: Average throughput vs. channel coherence time

decreases. In comparison, JUDS consistently benefits from the increased user diversity and achieves ∼50% higher utilization of the shared wireless channel.

7.4 Throughput vs. channel coherence time

In this section we vary the channel coherence time and study the impact on throughput. The channel coherence time is represented in terms of the velocity (m/s) of the user. Traffic was overloaded at both the uplink and downlink. Figure 13 shows that the throughput of ARF, RBAR, MAD and JUDS all decrease as the coherence time decreases due to the increasing probability of inaccurate channel condition estimates. JUDS again consistently outperforms all the other three due to its efficiency in the probing and maintenance of the channel condition.

7.5 Fairness

We finally evaluate JUDS' fairness, as defined in Section 5.1. Figure 14 shows normalized downlink throughput as the aggregate load on the uplink increases. In this set of simulations d is fixed at 100m. As we can see from the figure, because IEEE 802.11 MAC achieves long-term per-station fairness, the downlink throughput of ARF, RBAR, or MAD decreases quickly as the uplink load increases. Eventually, the shared wireless channel is dominated by the uplink traffic. As a result, the throughput gain of downlink opportunistic scheduling diminishes (Figure 11). In contrast, JUDS achieves a close-to-perfect uplink/downlink fairness through its cyclic scheduling. Fig. 15 shows the normalized throughput and channel time, shared among all 9 clients at both uplink and downlink. Note that nodes 1 to 9 are 20m to 180m away from the access point, at 20m increment. As a result of the proportional fairness scheduler (Eq. 1), user 1 with the best average channel condition achieves the highest throughput while user 9 with the worst channel condition achieves the lowest throughput. Meanwhile, the normalized channel time allocated to all clients are very close to each other, varying between 0.097 and 0.127.

8. CONCLUSION

The increasing rate and user diversity in a modern WLAN has made the adoption of opportunistic scheduling highly desirable. However, opportunistic scheduling on the downlink alone is fundamentally insufficient, especially when the amount of uplink traffic is high and the number of active users competing on the uplink is large. In this paper we presented JUDS, the design and evaluation of a joint uplink/downlink opportunistic scheduling algorithm for WLANs. Through synergistic integration of both the uplink and the downlink opportunistic scheduling, JUDS benefits from maximized channel diversity at significantly reduced scheduling overhead. Through analysis and extensive QualNet simulations, we show that JUDS improves the utilization of the shared wireless channel by up to 127% and achieves close-to-perfect fairness between uplink and downlink traffic.

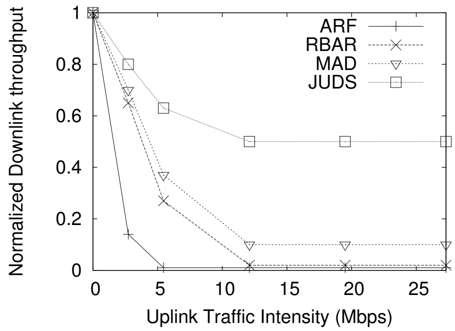


Figure 14: Uplink/downlink fairness

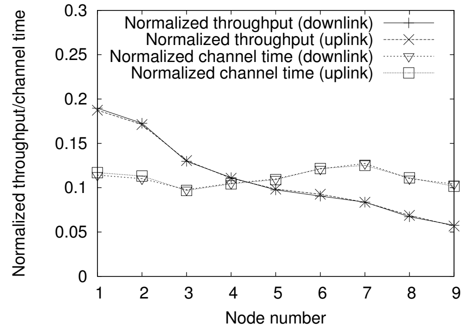


Figure 15: The throughput share and temporal share of each client.

9. REFERENCES

[1] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-1999, Aug. 1999.

[2] IEEE Std. 802.11a, 802.11a-1999, 1999.

[3] IEEE P802.11 - TG N - Standard for Enhancements for Higher Throughput, "http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm"

[4] A. Kamerman and L. Monteba. "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell Labs Technical Journal, Summer 1997, pp. 118-133.

[5] G. Holland, N. Vaidya and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," Proc. of ACM Mobicom'01, 2001.

[6] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks," Proc. of ACM Mobicom'02, 2002.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. "MACAW: A media access protocol for wireless LANs," Proc. of ACM SIGCOMM'94, 1994.

[8] F. Kelly, "Charging and rate control for elastic traffic", European Transactions on Telecommunications '97, 1997.

[9] J. Holtzman, "CDMA forward link waterfilling power control," Proc. of IEEE VTC2000-spring, May 2000.

[10] J. Holtzman, "Asymptotic analysis of proportional fair algorithm," Proc. of IEEE PIMRC'01, vol. 2, pp. F33-F37, Sept. 2001.

[11] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, S. Viterbi, "CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users," IEEE Communications Magazine, vol. 38, pp. 70-77, July 2000.

[12] A. Jalali, R. Padovani and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency high data rate personal communication wireless system," Proc. of IEEE VTC'2000, pp. 1854-1858, May 2000.

[13] Z. Ji, Y. Yang, J. Zhou, M. Takai and R. Bagorodia, "Exploiting Medium Access Diversity in Rate Adaptive Wireless LANs," Proc. of ACM Mobicom'04, 2004.

[14] D. Qiu and R. Srikant, "Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks," Proc. of ACM Sigcomm'04, 2004.

[15] T. Henderson and D. Kotz and I. Abyzov. "The Changing Usage of a Mature Campus-wide Wireless Network," Proc. of ACM Mobicom'04.

[16] "cachelogic research: The True Picture of peer-to-peer file sharing," http://www.cachelogic.com/research/p2p2004.php

[17] C. Chen and H. Luo. "The Case for Heterogeneous Wireless MACs," Proc. of ACM HotNets'05, 2005.

[18] D. Couto, D. Aguayo, J. Bicket, R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," Proc. of ACM Mobicom'03.

[19] Cisco Aironet 802.11a/b/g Wireless CardBus Adapter, "http:// www.cisco.com/en/US/products/hw/wireless/ps4555/products_data_sheet 09186a00801ebc29.html"

[20] QualNet User Manual v3.9. Scalable Network Technologies, Culver City, CA. http://www.qualnet.com

[21] The MathWorks Inc., MATLAB User's Guide, Version 6, 2000.

[22] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, P. Sinha, "Understanding TCP fairness over Wireless LAN," Proc. of IEEE Infocom'03, 2003.

[23] S. Kim, Y. Kim, S. Choi, K. Jang and J. Chang, "A High-Throughput MAC Strategy for Next-Generation WLANs," Proc. of IEEE WoWMoM'05, 2005

[24] T. Rappaport, "Wireless Communications Principle and Practice Second Edition," Prentice Hall 2002.

[25] D. Aguayo, J. Bicket, S. Biswas, G. Judd and R. Morris, "Link-level Measurements from an 802.11b Mesh Network," Proc. of ACM Sigcomm'04, 2004.

[26] J. Yoo, C. Kim, "On the Hidden Terminal Problem in Multi-Rate Ad Hoc Wireless Networks", in Lecture Notes in Computer Science (LNCS), Number 3391, Springer-Verlag, January 2005, pp 479-488.

[27] F. Cali, M. Conti, and E.Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," IEEE/ACM Transactions on Networking, 8(6), December 2000

[28] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," In Proc. of ACM Sigcomm 2005.