FlashLinQ: A Synchronous Distributed Scheduler for Peer-to-Peer Ad Hoc Networks

Xinzhou Wu Saurabh Tavildar Sanjay Shakkottai Tom Richardson Junyi Li Rajiv Laroia Aleksandar Jovicic
Qualcomm Qualcomm ECE Dept, UT Austin Qualcomm Qualcomm Qualcomm Qualcomm

***Abstract*—This paper proposes FlashLinQ - a synchronous peer-to-peer wireless PHY/MAC network architecture for distributed channel allocation. By leveraging the fine-grained parallel channel access of OFDM, FlashLinQ develops an analog energy-level based signaling scheme that enables SIR (Signal to Interference Ratio) based distributed scheduling. This new signaling mechanism and the corresponding allocation algorithms permit efficient channel-aware spatial resource allocation, leading to significant gains over a CSMA/CA system with RTS/CTS.**

FlashLinQ is a complete system architecture including (i) timing and frequency synchronization derived from cellular spectrum, (ii) peer discovery, (iii) link management, and (iv) channel-aware distributed power, data-rate and link scheduling. We implement FlashLinQ over licensed spectrum on a DSP/FPGA platform. In this paper, we present performance results for FlashLinQ using both implementation and simulations.

I. Introduction

With the proliferation of data services and smartphones (e.g., the iPhone), there has been a renewed interest in ad hoc wireless networks. Such networks promise scalability and great performance improvements in utilizing scarce spectrum resources. In the network modeling and algorithms community, this motivation has led to new research in cross-layer synchronous resource allocation mechanisms [1] which promise great theoretical gains. However, wireless ad hoc network implementations and deployments have predominantly focussed on asynchronous CSMA/CA mechanisms and modifications thereof, due to the belief that both messaging (for channel-state aware spatial coordination) and synchronization overheads will render synchronous cross-layer schemes impractical.

This paper demonstrates that we can design and implement a practical, synchronous cross-layer MAC and PHY architecture, which supports any of the cross-layer mechanisms (e.g., back-pressure [1], MaxWeight [2]) that have been proposed for network resource allocation. We describe FlashLinQ: a new OFDM-based synchronous architecture for MAC/PHY, that *(i)* builds new *analog* signaling mechanisms for multi-node distributed coordination, *(ii)* enables distributed channel-aware spatial resource allocation and packing, and *(iii)* supports QoS and fairness at multiple time-scales. In coordination with cellular providers to derive fine-grained timing for network synchronization, this paper implements FlashLinQ over a *licensed* spectrum on a DSP and FPGA-based platform to demonstrate its feasibility and the significant performance benefits accrued.

Technical Overview: FlashLinQ is an OFDM-based system that enables node discovery, channel allocation, and link scheduling with power control. The key technical innovation is to leverage the physics of propagation to develop analog signaling where information is implicitly encoded in both the presence of a signal as well as the signal strength. This mechanism, combined with the robust orthogonality of OFDM [3], is used in a tone matrix structure to provide a spatio-temporal template to support an agile distributed control layer for coordination and resource allocation. Our new mechanism addresses key problems in spatial resource allocation: *(a)* orthogonality vs. reuse (i.e., which links are allowed to simultaneously transmit and at what power-levels and data-rates); *(b)* channel-aware distributed scheduling (to account for fast/slow fading based channel gain variations); and *(c)* hidden/exposed nodes.

Hardware Overview: The FlashLinQ modem prototype is based on a general FPGA and DSP based platform which operates at a carrier frequency frequency of 2.586 GHz . The time domain sample level processing and LDPC decoder are implemented in FPGA (Xilinx Virtex-4). The frequency domain symbol level processing is implemented in a TI TMSC64x DSP chip. The L2 functionalities, including packet dissembling and resembling, fast ARQ, etc., are also implemented in DSP. The DSP communicates with a Linux based host machine via Ethernet interface.

Deployment Overview: Ad-hoc peer-to-peer communication systems have traditionally operated in unlicensed spectrum. This project is very different – we are deploying an ad-hoc network to extend managed services by cellular providers in licensed spectrum. This has great benefits for users as well as network providers. Users get more predictable performance because the interference is managed and an extended battery life due to access to fine-grained timing that enables synchronous operation (FlashLinQ is designed to leverage any of CDMA/GSM cellular timing [4], DVBH timing [5], GPS timing [6], along with in-band timing). Simultaneously, network providers get increased spectral and power efficiency. As an aside, we note that FlashLinQ can be deployed in mixed licensed-unlicensed spectrum; indeed given that the licensed spectrum communication is inherently more reliable, it can be used as a control layer, for example, for peers to discover each other and, furthermore, to negotiate the use of unlicensed spectrum for the bulk of data traffic. Our current deployment setting (licensed spectrum) makes the system design tradeoffs very different from that in unlicensed spectrum. In particular, the spectrum cost is a big factor and usually the available bandwidth is much smaller. Therefore, a crucial design objective is to maximize the spectral efficiency. Here the spectral efficiency is defined not just on a link level but, more importantly, on a system level where multiple links are to share the bandwidth efficiently.

A. Motivation for FlashLinQ

The goal of FlashLinQ is to schedule a channel-state aware *maximal* independent set at any given time slot based on the current traffic and channel conditions. The independent set is defined based on the link SIRs[1], i.e., all links in the chosen independent set simultaneously have a "large enough" SIR. As can be easily seen, determining such a set is difficult because of the strong coupling: each link's SIR depends on all the other links that are also part of the independent set.

To illustrate the importance of the SIR in link scheduling, consider for the moment a fixed transmitter-receiver pair (denoted by Tx-A, Rx-A) over which data transfer is to take place. From a spatial spectral efficiency perspective, the key problem in resource allocation is to determine which other links (transmitter-receiver pairs) are allowed to *simultaneously* transmit without creating too much interference at Rx-A. With a pure CSMA/CA mechanism, the transmitter simply senses the carrier and transmits if it hears no interference. This provides no protection at the receiver. To alleviate this problem, an RTS/CTS mechanism is used in conjunction with CSMA/CA: effectively, a potential transmitter, say Tx-B, that can hear either an RTS from Tx-A or a CTS from Rx-A, and hence can cause interference either at Tx-A (for ACK protection) or Rx-A (for data protection) to exceed -91 dBm is not permitted to transmit[2]. As a first approximation, this is equivalent to both Tx-A and Rx-A drawing a "protection circle" of a fixed radius (see Figure 1) around them – any transmitter within these circles is not permitted to transmit simultaneously (note that "distance" here corresponds to RF-distance that depends on both physical distance as well as channel fading). This mechanism ensures that as long as the intended transmitter (i.e. Tx-A) is close enough to the intended receiver (i.e. Rx-A), any single transmitter will not cause the data-transfer and the data acknowledgement on the link Tx-A – Rx-A to fail. A similar restriction is also placed on a potential receiver Rx-B.

However, from communications theory, we know that this type of protection is in general, neither necessary nor sufficient[3]: *successful decoding occurs at Rx-A as long as the SIR (Signal to Interference Ratio) is sufficiently large to permit message decoding*. This is very different from a fixed interference level at Tx-A or Rx-A – what we really need to ensure is the ratio of signal power to interference to be large enough. This implies that the protection circle drawn by Rx-A should be of a variable radius that is proportional to the RF-distance between (Tx-A – Rx-A). This is illustrated in Figure 1. As we will show in Section III-A, this condition ensures a fixed SIR protection at Rx-A from Tx-B. With this mechanism, much more efficient channel-aware spatial packing occurs, i.e., our SIR based mechanism leads to a channel-state aware maximal matching (see Section III-A) that can lead to spatial throughput gains over an 802.11g system (see Section IV).

[1]Signal to Interference Ratio at a receiver, where the term "signal" corresponds to the received power due to the intended transmitter, and the term "interference" corresponds to the power received due to all other nodes that are simultaneously transmitting.

[2]Note that the actual energy threshold depends on the decoding algorithm as well as on transmit power; see Section IV for additional discussion.

[3]These problems with CSMA/CA and RTS/CTS have been noted in literature. In particular, the insufficient protection provided by RTS/CTS has been observed in [7]. Also, the unnecessary excess protection provided by RTS/CTS has been studied in [8].

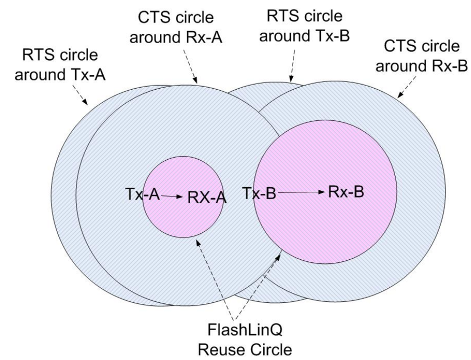

Fig. 1. Reuse radius in FlashLinQ vs. 802.11 (CSMA/CA with RTS/CTS).

II. Related Work

Distributed scheduling in wireless networks has attracted attention of many researchers in the field over the last several years. Interesting results and intuitions were obtained in these studies regarding the potential throughput loss caused by a class of *maximal matching* distributed scheduling algorithms as compared to a genie-aided centralized algorithm [9][10][11], along with various ways to improve maximal matching [12][13]. In particular, recent results in this field [14][15] show that queue length based distributed scheduling can be *throughput optimal*. Many of these schemes are based on combinatorial interference models at the physical layer and focus how to schedule links given the *feasible* independent sets, i.e., links are allowed to transmit simultaneously based on the combinatorial interference model. However, the issues of defining these feasible independent sets based on actual SIRs with fading channels (channel coefficients could change on a per-time-slot basis), and then incorporating multiple power-levels and rates are usually not addressed.

In parallel, there has been a growing interest in integrating advanced physical layer techniques, including network coding, interference alignment and cancelation in existing wireless networks [16][17][18][19]. The emphasis of these works is to show the practicality of these techniques in a real network rather than theoretically characterizing the potential gain. Most of these analyses and prototyping efforts, though, are based on the WiFi physical layer, where OFDM is used only as a point-to-point physical layer technology, i.e. both control signaling and data transmissions use full bandwidth to transmit rather than trying to multiplex users in the frequency domain.

In this paper, we study distributed maximal-matching-type scheduling protocols based on a SIR model on top of a fully implemented OFDMA-based PHY layer. As compared to CSMA/CA with RTS/CTS based protocols proposed for 802.11 [20], the key differences here are threefold: (1) No CSMA is needed after introducing a new synchronous PHY; (2) the signaling equivalent to RTS and CTS are very different in FlashLinQ based on the OFDMA-based PHY; (3) the yielding decisions are SIR based rather than SNR based as in 802.11. We show that a significant gain in spatial reuse can be achieved in FlashLinQ.

It should be noted that similar ideas as listed above have been proposed in the context of 802.11. For example, removing CSMA and having a RTS-CTS only MAC is proposed and analyzed in [21] and [22]. Also, introducing SIR based media access protocol was proposed in [23] via the use of out-of-band control channels. However, these protocols have various implementation or robustness issues and are not adopted into the main 802.11 standard body. In FlashLinQ, the synchronous nature of the PHY and the light weight design of control signals make it much easier to incorporate these salient features.

III. PHY/MAC ARCHITECTURE

FlashLinQ is designed to be a synchronous 5 MHz peer-to-peer system that enables distributed, channel-aware spatial scheduling. The basic physical layer technology used for FlashLinQ is OFDM/OFDMA. OFDM is a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier modulation method. OFDMA is a multi-user version of OFDM. OFDM/OFDMA has been the underlying technology for most of the new generation wireless systems such as 802.11g [20], LTE, and Wi-Max. We refer to [3] for an excellent tutorial on OFDM and OFDMA. There are two key aspects that enable FlashLinQ's distributed resource allocation:

1) **Signaling Mechanism**: We utilize the flexibility of parallel, single-tone OFDM channels to architect an energy-level based (analog) signaling mechanism that provides a miniaturized template of data transmissions, but *without collisions*. This mechanism enables all links to observe and infer (both from interference and rate perspectives) what would happen if they did transmit data, but without actually going through the data transmissions and the resulting contentions.
2) **Spatial Packing**: By cleverly choosing the energy level at which single-tone signals are transmitted, this analog and parallel signaling mechanism enables each link (Tx-Rx pair) to determine the SIR at each receiver. This ensures that the schedule is determined based on the link qualities that would result from choosing this schedule. As we discussed earlier (see Section I-A), this is critical to ensure efficient spatial packing.

The scheduling operation occurs every 2.08 msec in FlashLinQ (see Figure 2 for a timeline). FlashLinQ, however, requires several other functionalities for its operation:

1) **Timing synchronization**: this functionality underpins the whole system by providing timing synchronization to all nodes. Having a common notion of time optimizes system performance by allowing for dedicated channels such as peer discovery and connection management.
2) **Peer discovery**: This enables nodes to transmit presence information and detect the presence of other nodes in the neighborhood.
3) **Link management**: This allows nodes to operate in power saving mode and to page and be paged as needed for the purpose of establishing links (assign link IDs).

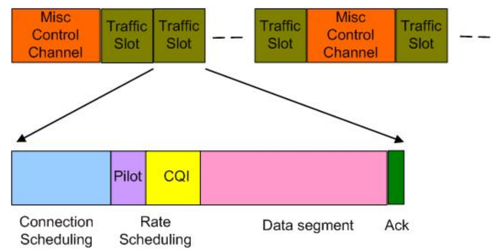

Fig. 2. The FlashLinQ operation timeline: data transmissions occur in slotted time of around 2 msec each. Within each slot, link and rate scheduling is followed by the actual data transmission. In addition, every 1 second, resources are allocated for other channels such as peer discovery and link management.

In the rest of the section, we begin with a detailed description of scheduling and resource allocation, followed by a more abridged description of link management and peer discovery.

A. Scheduling and Data Transmission

We now describe the main technical contributions of the paper: a low overhead distributed scheduling algorithm. In this section, we first describe the key ideas that form the basis of our algorithm, and then describe the signaling mechanisms that enable this approach.

As discussed in Section I-A, the goal of FlashLinQ is to schedule a channel-state aware *maximal* independent set of links for any given time slot based on the current traffic and channel conditions. The independent set is defined based on the link SIRs, i.e., all links in the chosen independent set simultaneously have a "large enough" SIR.

1) Key design ideas: To illustrate the main elements in our algorithm to achieve this goal, we first look at a simple two-link example, links $A \to B$ and $C \to D$ as shown in Figure 3: here, two links[4] have direct gains $\{|h_{AB}|^2, |h_{CD}|^2\}$, and also cross-link gains, $\{|h_{AD}|^2, |h_{BC}|^2\}$. In this setting, it is clear that if the cross-link gains are small, the links AB and CD will not significantly interfere with each other, and thus can be simultaneously scheduled. On the other hand, if the cross-link gains are large, only one of the links can be scheduled at any instant of time.

[4] h_{AB} is the path-loss between nodes A and B; its magnitude-squared corresponds to the fraction of the transmitted power from A that is received at B.

A simple way of determining if both links can be simultaneously scheduled or only one link can be scheduled, is to assign *priorities* to links. The understanding is that the higher priority link always get scheduled. However, for the low priority link to also be scheduled, it has to check if its transmission is not going to cause excessive *SIR damage* to the high priority link. This is done by comparing the *would-be SIR* of the high priority link with a *SIR threshold* if the low priority link does go ahead with its data transmission. This mechanism ensures that the high priority link is protected, and both links get scheduled only if the cross-link gains are "weak enough". By randomizing the priority of links over time, fairness across links can be maintained. Below, we elaborate on this discussion.

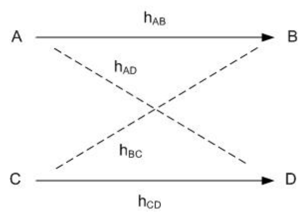

Fig. 3. Scheduling for two links

Specifically, in the two-link example in Figure 3, assuming link $A \rightarrow B$ has higher priority in the current slot, link $C \rightarrow D$ can potentially be scheduled simultaneously if C *doesn't cause too much interference to* B: we define this to be that the SIR of link $A \rightarrow B$ due to C's transmission should be at least γ_{TX} dB[5]. Then, the protection condition can be written as:

$$\frac{P_A|h_{AB}|^2}{P_C|h_{BC}|^2} \quad > \quad \gamma_{TX}, \tag{1}$$

where P_A denotes power used by node A and P_C denotes power used by node C. In our system the power control for each link is done on a slow time scale, and the power does not change dynamically from slot to slot. We do not discuss the power control algorithm here; however the scheduling algorithm presented here works for arbitrary power levels.

The condition (1) ensures protection for a higher priority link. Next, we address the second question: is this condition alone enough to allow one link to get scheduled? In the simple two link example we considered above, the answer is yes. But in a more general network with multiple links, the answer is no. Consider the setup in Figure 4. The link $C \rightarrow D$ ensures enough protection for link $A \rightarrow B$, but since D is close to A, the SIR seen by D will still be low. So, in such a scenario, we say that the link $C \rightarrow D$ should NOT transmit. This will potentially allow for a much better packing where another link $E \rightarrow F$ can be scheduled allowing for higher system throughput. Thus, we impose an additional condition for a

[5]In our implementation, we use a value of 9 dB for γ_{TX}. This value was chosen for optimizing system spectral efficiency and was determined based on both simulation and implementation results. Further, our implementation supports an adaptive threshold which changes over a slower time-scale which can be further used for providing QoS or fairness for links

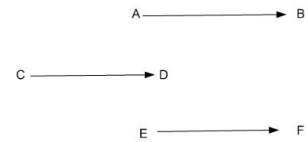

Fig. 4. Scheduling for three links

link $C \rightarrow D$ to be scheduled: D *should see a reasonable SIR if scheduled*. We define this to be that the SIR of link $C \rightarrow D$ due to A's transmission at be least γ_{RX} dB: [6]

$$\frac{P_C|h_{CD}|^2}{P_A|h_{AD}|^2} \quad > \quad \gamma_{RX} \tag{2}$$

So far, we have illustrated the three main elements in our algorithm in the simple network examples as shown in Figure 3 and Figure 4: (i) a fair priority assignment mechanism; (ii) a transmit yielding criterion to protect the receiver in higher priority links; and (iii) a receive yielding criterion to further improve network spatial packing in a multi-link scenario. In order to check these criteria in a distributed way, C needs to determine the LHS of equation (1) which involves not only the cross channel gain h_{BC}, but also P_A and h_{AB}. Similarly, D needs to determine the LHS of equation (2) which involves not only the channel gain h_{CD}, but also P_A and h_{AD}. Our main contribution is a protocol by way of which C and D infer the relevant information based on minimal signaling from A and B. In particular, the algorithm does not require any dedicated signaling between links AB and CD. The main idea used for estimating the above two criteria in a distributed way is two analog tone signals *inverse power echo* and a *direct power signal* that are used to estimate various SIRs using the geometry of the problem. The two signals will be described in the context of Figure 5.

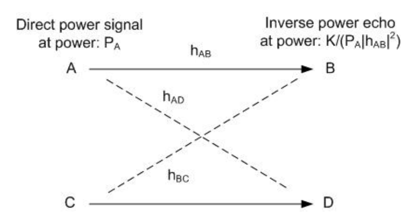

Fig. 5. Direct power signal and inverse power echo

The purpose of the *direct power signal* that is sent by a transmitter is to allow other receivers to estimate the SIR that they will see because of the transmitter. *The direct power signal sent by* A *is a signal sent at power* P_A. The signal will be received by D at power $P_A \times |h_{AD}|^2$. Similarly, the direct power signal sent by C will be of power P_C. This signal will be received by D at power $P_C \times |h_{CD}|^2$. From these two signals, D can estimate $\frac{P_C \times |h_{CD}|^2}{P_A \times |h_{AD}|^2}$ thereby determining if Equation (2) is satisfied or not.

[6]In our implementation, we use a value of 9 dB for γ_{RX}. As before, this value was chosen for optimizing spectral efficiency, but also maintaining fairness across links with different link lengths.

The purpose of the *inverse power echo* that is sent by a receiver is to allow other transmitters to estimate the SIR that the receiver will see due to the transmitters' transmission. *The inverse power echo is a signal sent by B at power* $\frac{K}{P_A|h_{AB}|^2}$ *for a well-defined system constant* K. The signal will be received by C at power $r_p = \frac{K|h_{BC}|^2}{P_A|h_{AB}|^2}$. From this, C can determine the SIR estimate:

$$\frac{P_A \times \widehat{|h_{AB}|^2}}{P_C \times |h_{BC}|^2} = \frac{K}{r_p \times P_C}. \tag{3}$$

This estimate is then used to determine if Equation (1) is satisfied or not. We note that a similar idea of inverse power CTS has been proposed in an asynchronous 802.11-like setting in [23], via the use of out-of-band *busy* tones.

2) Algorithm description: In the network setting, we consider a cascaded scheduling algorithm where links are arranged in a pseudo-random order, and scheduled in a sequential manner. In other words, the links are strictly ordered according to a random priority list. A link at priority level L is scheduled if and only if *both* the transmitter and the receiver of link L decide to allow data transfer over this link. The conditions for this to happen are:

- *The link L doesn't cause too much interference to an already scheduled link (i.e., of priority* $\{1, 2, \ldots, L-1\}$*:* we define this to be satisfied if the SIR of an already scheduled link due to interference from link L to be at least γ_{TX} dB. If this is not satisfied, Tx-yielding (transmitter yielding) occurs, where the transmitter node of link L decides not to transmit in order to satisfy SIR constraints at higher priority receivers.
- *The link L will see a reasonable SIR if scheduled:* we define this to mean that the SIR of link L is at least γ_{RX} dB. The interference here is calculated to be the *sum* of interference from all higher priority links. If this is not satisfied, Rx-yielding (receiver yielding) occurs, where the receiver node of link L decides not to allow data transfer over this link. As we discussed earlier, this allows for more efficient spatial packing.

The priority ordering, along with the cascaded Tx and Rx-yielding ensures that the resulting schedule is a channel-aware matching. By iterating, it can be easily shown that this algorithm leads to a maximal matching. Finally, re-randomizing the priority order at each timeslot ensures fairness across links.

However, we point out that this algorithm does not guarantee a certain minimum SIR for a scheduled link. This is because of the fact that multiple interfering transmitters, each of which guarantees a certain SIR to a receiver, together can cause the SIR due to the total interference to fall below the yielding threshold. This is addressed by using a rate scheduling mechanism that subsequently does an SIR estimation based on wide-band pilots from *all* the scheduled transmitters (see Section III-A3). This estimated SIR (based on total interference) is used to decide the code rate and modulation to be used for each of the links for a given time slot. This is further discussed in Section IV-B.

3) Signaling design: As discussed earlier, the basic data transfer unit is a time slot, with each slot of 2.08 milliseconds duration. Each slot occupies the entire 5 MHz, and a scheduling decision is made on per slot basis and independently of other slots. Further, each slot is divided into four physically separate subchannels (see Figure 6): link scheduling, rate scheduling, data segment and ACK, each described below.

Link scheduling: Each link has a unique link-ID (or CID) between 1 and 112 assigned to it. These link CIDs each correspond to a pair of single tones (one each for the transmitter and the receiver) within a tone matrix having 112 tone pairs (28 parallel tones, over two blocks – the Tx-block and Rx-block – of four OFDM-symbols, see Figure 6). The mapping from the CIDs to the actual tone pair within the matrix is randomized (every time-slot, a new mapping is used). Figure 6 shows a realization for two CIDs, say 3 and 7.

Recall that the algorithm described earlier in this section needed *(i)* a mechanism for sending analog signals between Tx-Rx pairs, *(ii)* that these signals should not interfere with each other, and *(iii)* a priority ordering among links. The tone matrix of single-tone pairs satisfies all these requirements: these tones are orthogonal and analog, and a natural priority ordering is assigning the highest priority to the top-left tone-pair and the lowest to the bottom-right, and with lexicographical ordering (the 'x' coordinate has higher priority than the 'y' coordinate) – thus in Figure 6, CID 7 has higher priority than CID 3. The random remapping of CIDs to tone pairs for each time-slot ensures fairness across links.

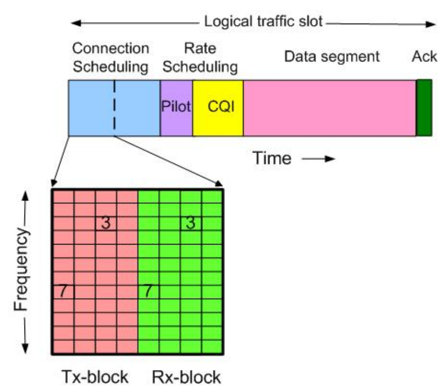

Fig. 6. Structure of link scheduling

In the language of WiFi, it will help to think of the Tx-block as the RTS block and the Rx-block as the CTS block. The purpose of these two blocks is as follows:

- **The Tx-block** is used by potential transmitters to make a request to be scheduled. The transmitter node "lights up" (i.e., transmits power) on the symbol allocated to it in the Tx-block. This request is a *direct power signal*, that is it is sent at power that would be used for the traffic channel. All the potential receivers listen to the Tx-blocks and determine if they need to perform Rx-yielding.
- **The Rx-block** If a receiver chooses to Rx-yield, it does not respond (i.e., sends no power in the symbol allocated

to the link in the Rx-block); other-wise it "lights up" its symbol at the *inverse echo power* level described earlier in this section. All the transmitters listen to the tones in the Rx-block to determine whether to Tx-yield.

A link is scheduled if neither the transmitter nor the receiver yields. This operation is repeated over multiple rounds (see Figure 6, where the tone matrix is repeated multiple times) to optimize the packing – in subsequent rounds, additional links can be added, but already scheduled links will not yield. This, we can show, leads to a maximal channel-aware matching; we skip the details for brevity.

Another issue we have not discussed is that of bi-directional traffic and half-duplex constraints. Observe that the notion of a link in this paper is directional – thus, there are two links between a pair of nodes: forward and reverse. Scheduling contentions between these two links in FlashLinQ is resolved by having *two* Tx-blocks - one for the forward and one for the reverse link transmitter, and a single Rx-block (thus, each bi-directional link has three tone symbols associated with it). Each (directional) link transmitter lights up its symbol to request to be scheduled. If both directions request, the conflict is resolved by a simple alternating priority – in even time-slots, the forward link ins the conflict and in odd time-slots, the reverse link wins the conflict. The corresponding receiver for the winning link responds by inverse power echo as before in the Rx-block.

Rate scheduling: All transmitters that were scheduled to transmit in the link scheduling slot will use the rate scheduling channel to determine the code rate and modulation that they should use for the data segment. This channel is composed of a wide-band PILOT sent by transmitters and a CQI (channel quality indicator) sent by the receivers. This slot by slot rate estimation achieves a more accurate estimation of the SIR (based on total interference) than in link scheduling, and each link's code rate and modulation is chosen based on the actual SIR due to all the other scheduled links.

Data segment: All scheduled links transmit over all tones. Note that single-tone signals are used only during scheduling, and our mechanism ensures that simultaneous transmissions over the various links do not significantly interfere with each other.

Acknowledgement: Acknowledgement uses orthogonal channels based on the CID to signal successful reception of the packet. There is a dedicated slot used for acknowledgment so that the acknowledgement signals do not interfere with other signals.

B. Link management

Link management deals with a slow time scale optimization of allocating *locally unique* identifers for each link (i.e., assignment of link IDs that are used in scheduling). We assume that links communicate over relatively longer periods of time (few seconds) as compared to one communication burst. This assumption is used to allocate different link identifiers (CID) to each link which are used for the scheduling algorithm. Note that even though we assume that the links communicate over relatively long periods of time, the traffic itself may be bursty in nature. We choose CIDs for links based on a SIR criterion that is analogous to that used in link scheduling; details available in [24].

C. Peer discovery

The peer discovery mechanism uses a small fraction of dedicated time-slots to discover the presence of other nodes in the neighborhood (up to 1 km range). This long range discovery is done by using rateless codes to broadcast peer IDs that enable discovery at extremely low SNR values over moderate timescales (about 10 seconds to discover around 1000 devices over a 1 km range).

One of the main concerns about peer discovery in an ad hoc wireless network is the energy efficiency of the algorithm. We dedicate a small number of time slots (timing for synchronization is derived from cellular CDMA in the US and potentially DVB-H in Europe; our protoype described in Section IV uses in-band timing) for the purpose of peer discovery (we use ~ 20 ms every 1 second). This time slot is used for transmitting as well as receiving presence information of nearby nodes. All devices are required to participate in peer discovery even if they are not actively communicating with other devices. In our system, this leads to a 2% duty cycle for a node to stay awake,which leads to an acceptable standby time of the device. The details of the discovery mechanism are not discussed in this paper as the focus of this paper is on the scheduling algorithm.

IV. Implementation and Simulations

In this section, we evaluate the efficiency of FlashLinQ link scheduling by experiments using the FlashLinQ prototype devices and also simulations.

A. Measurement Setup and Results

The FlashLinQ prototype modem (as shown in Figure 7) is based on a general FPGA/DSP based platform which operates at 2.586GHz carrier frequency. We chose TI DSP chipset TMS-C6482 and XiLinx Virtex-4 FPGA to build the OFDMA based FlashLinQ physical layer modules. Specifically, the time-domain sampling processing and FFT are done in the FPGA and frequency domain symbol level processing are performed mainly in the DSP. The link scheduling algorithms reside in the DSP as a result of this separation in the current prototype. More details on the implementation is available in [24].

Fig. 7. Prototype modem.

Our experiments are conducted with four devices named AMC Theaters (AT), Movie Buff (MB), Teen Shopper (TS) and Pub Patron (PP). The first set of results shows how FlashLinQ devices make transmitting or yielding decisions (spatial packing) at different channel conditions. In this experiment, we have four devices forming two links, one between AT and MB and the other between TS and PP. We let the four of them sit on a straight line (as shown in Figure 8) within a room. In the beginning of the experiment, we let the two transmitters, TS and AT, stay at the far sides of the picture (about 3 meters away from each other) and the two receivers, PP and MB close to their interferers. We then move PP and MB closer to their transmitters and thus create strong signal and weaker interference for both links. In FlashLinQ, since the yielding decisions are made upon SIRs instead of SNRs, we would expect the two links to orthogonalize the channel use in the beginning of the experiment and switch to full reuse when the interferers are not strong enough. Note that if a 802.11 type of protocol is used, fully orthogonalization is the only possible result since, in our experiment, each node is well within the carrier sensing range of any other node in the system, including its intended transmitter or receiver and also its interferers.

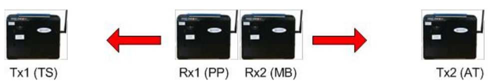

Fig. 8. Reuse versus orthogonalization experiment setup

To collect the data, the modem reports its current link scheduling status to the Linux based host every second. Figure 9 shows plots for the sum rate (the top window) and individual rates for TS, PP, MB, and AT (the bottom 4 windows in that order) along Y-axis versus time along the X-axis. Since the traffic was unidirectional, two of the nodes (TS and AT) report zero rate throughout. From Figure 9, we can see that both links are yielding to the other transmitter in the beginning of the experiment and both links achieve half of the full capacity. Since FlashLinQ guarantees fair sharing of the air interface between the contending links, both links get 50% of the resource. When the two receivers move closer to their intended transmitter, both users get scheduled in all traffic slots by reusing the available bandwidth. Note here in these experiments, we only turn on the low rate options and the peak rate of a link is limited by $1.5Mbits/s$.

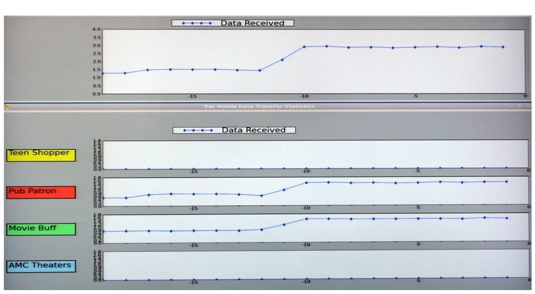

Fig. 9. Measurement data as a time series.

The second set of measurements are collected by repeating the previous experiments with different yielding thresholds. The emphasis here is to show the importance of SIR based yielding as compared to SNR based yielding, e.g., carrier sensing protocol in 802.11, and also no yielding at all. In particular, we compare the system behavior under the following three different choices of yielding thresholds: *(1)* Both transmit and receive yielding threshold to be around 9dB: this is the normal operation of FlashLinQ; *(2)* Both transmit and receive yielding threshold to be around 18dB: this choice of yielding threshold has a strong bias towards orthogonalization. The purpose of this scenario is to mimic the behavior of 802.11; and *(3)* Both transmit and receive yielding are disabled, i.e. the yielding thresholds are chosen to be $-\infty$dB: this forces all devices to reuse the full spectrum, regardless the surrounding environment. The measured results are shown in Figure 10. In the low interference scenario, where two links are geographically separated, yielding threshold choices at $-\infty$ and 9 dB lead to similar result since both links see little interference from the other link. Yet the aggressive threshold of 18 dB suffers since it orthogonalizes unnecessarily and thus gets only 50% of the full capacity. In the strong interference case where the receivers get closer to their interfering transmitters, the results obtained with 9dB and 18dB yielding threshold are roughly the same since both force the two links to orthogonalize which is better than completely reusing the spectrum. However, the performance with full reuse is much worse since both links transmit simultaneously but with bad SIRs. It can be easily seen here that a proper choice of the yielding threshold can achieve a good throughput in both strong and weak interference scenarios.

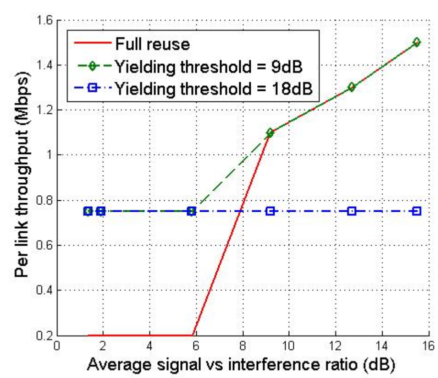

Fig. 10. Lab measurement data with different yielding thresholds.

B. Simulation Results

In this section, we present simulation results comparing FlashLinQ with a 802.11g protocol. Our simulations are based on a detailed software implementation of both FlashLinQ and 802.11g, and all signaling overheads are fully accounted. The FlashLinQ system operates over a 5 MHz spectrum (for which this system is designed), whereas the 802.11g protocol operates over a 20 MHz spectrum. Our results are hence normalized to bits/sec/Hz to account for the excess bandwidth for 802.11g. For the WiFi protocol, as per 802.11g specifications, an energy sensing threshold of -76 dBm and PLCP header decoding (at 0.5 dB SINR) is used for yielding.

We simulate both outdoor and indoor settings. For the outdoor deployment, links are dropped randomly in a $1000m \times 1000m$ square. Link lengths are chosen to be 20m. To remove the boundary effect, we introduce a wrap-around model in the signal strength calculation between any two nodes. The pathloss between any two nodes is modeled is calculated based on ITU-1411 LOS model with antenna height of 1.5 meters (see [25] for a summary of channel models). For the indoor deployment, links are dropped randomly in a $50m \times 100m \times 20m$ building with 5 floors each of height 4 meters. Link lengths are chosen to be 20m. For 40% of the links, the two devices are one floor apart, while for the remaining links, the devices are on the same floor. The pathloss between any two nodes is based on Keenan-Motley model with the following parameters: floor penetration loss = 15 dB; wall penetration loss = 5 dB; 1 wall every 22 meters. In addition, for both scenarios, carrier frequency is assumed to be 2.4GHz. Slow fading is modeled as independent shadowing for each channel gain with standard deviation of 10 dB. A transmit power of 20dBm is used for both systems along with a noise figure of 7 dB and an antenna gain of -2.5 dB per device.

The Figure 11 shows the capacity gain we have with the FlashLinQ protocol as compared to WiFi protocol in above two deployments. We vary the number of links from 1 to 256. From the curve, we see that WiFi sum rate saturates much earlier and at a much lower value as compared to FlashLinQ, and that FlashLinQ results in an increase of 450% sum throughput with 256 links.

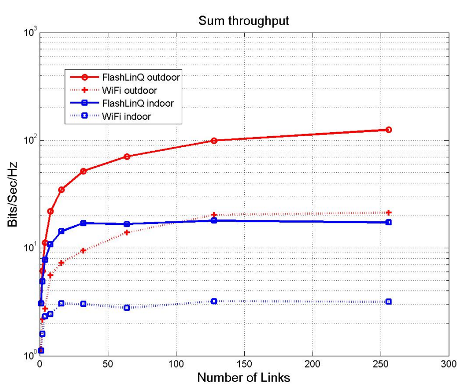

Fig. 11. Sum throughput comparison

V. Conclusions

This paper has proposed FlashLinQ – a synchronous peer-to-peer wireless PHY/MAC network architecture for distributed channel allocation. The key scheduling objective has been to develop a channel aware maximal independent set scheduling algorithm in a distributed manner. Our performance study has indicated that significant spectral efficiency gains can be obtained over 802.11 – this is key for our licensed spectrum deployment scenario. Our effort is currently focused on developing a chipset solution (ASIC) for FlashlinQ for use in a large-scale field trial, and can potentially be incorporated into commercial wireless devices.

References

[1] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1), 2006.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. Scheduling in a queuing system with asynchronously varying service rates. *Probab. Eng. Inf. Sci.*, 18(2):191–217, 2004.

[3] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, Cambridge, UK, 2005.

[4] 3rd Generation Partnership Project 2 (3GPP2). Cdma2000 high rate packet data air interface specification c.s20024-a v2.0. September 2005.

[5] Digital video broadcasting (dvb); transmission system for handheld terminals (dvb-h), etsi en 302 304 v1.1.1 (2004-11).

[6] B. W. Parkinson and J. J. Spilker. *The global positioning system: theory and applications*. AIAA (American Institute of Aeronautics & Ast); 1st edition, January 1996.

[7] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *GLOBECOM*, 2002.

[8] F. Ye, S. Yi, and B. Sikdar. Improving spatial reuse of IEEE 802.11 based ad hoc networks. In *GLOBECOM*, 2003.

[9] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. Netw.*, 14(2):302–315, 2006.

[10] S. Sarkar, P. Chaporkar, and K. Kar. Fairness and throughput guarantees with maximal scheduling in multi-hop wireless networks. In *WiOpt*, pages 286–298, 2006.

[11] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu. Asynchronous congestion control in multi-hop wireless networks with maximal matching-based scheduling. *IEEE/ACM Trans. Netw.*, 16(4):826–839, 2008.

[12] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks. *IEEE/ACM Trans. Netw.*, 17(4):1132–1145, 2009.

[13] S. Sanghavi, L. Bui, and R. Srikant. Distributed link scheduling with constant overhead. In *Proceedings of ACM SIGMETRICS*, pages 313–324, 2007.

[14] Libin Jiang and Jean C. Walrand. Convergence and stability of a distributed csma algorithm for maximal network throughput. In *CDC*, pages 4840–4845, 2009.

[15] Jian Ni and R. Srikant. Distributed csma/ca algorithms for achieving maximum throughput in wireless networks. *CoRR*, abs/0901.2333, 2009.

[16] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: analog network coding. In *SIGCOMM*, pages 397–408, 2007.

[17] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *SIGCOMM*, pages 159–170, 2008.

[18] S. Gollakota, S. Perli, and D. Katabi. Interference alignment and cancellation. In *SIGCOMM*, pages 159–170, 2009.

[19] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with wi-fi like connectivity. In *ACM SIGCOMM*, pages 27–38, 2009.

[20] *IEEE 802.11g-2003: http://standards.ieee.org/getieee802/download/802.11g-2003.pdf*.

[21] P. Karn. MACA - a new channel access method for packe radio. In *ARRI /CRRI, Amateur Radio 9th Computer Networking Conference*, September 1990.

[22] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LAN's. In *ACM SIGCOMM*, pages 212–225, London, United Kingdom, 1994.

[23] J. Monks, V. Bharghavan, and W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of IEEE Infocom*, 2001.

[24] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic. Flashlinq: A synchronous distributed scheduler for peer-to-peer ad hoc networks. Technical Report, 2010.

[25] Dieter J. Cichon and Thomas Krner. *Propagation prediction models*. COST 231 Final Rep, 1995.