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

***Abstract*—This paper proposes FlashLinQ - a synchronous peer-to-peer wireless PHY/MAC network architecture. FlashLinQ leverages the fine-grained parallel channel access offered by OFDM and incorporates an analog energy-level-based signaling scheme that enables SIR- (Signal to Interference Ratio) based distributed scheduling. This new signaling mechanism, and the concomitant scheduling algorithm, enables efficient channel-aware spatial resource allocation, leading to significant gains over a CSMA/CA system using 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. FlashLinQ has been implemented for operation over licensed spectrum on a DSP/FPGA platform. In this paper, we present FlashLinQ performance results derived from both field data and simulations.

I. Introduction

With the proliferation of data services and smart-phones (e.g., the iPhone), there has been a renewed interest in ad hoc wireless networks. Such networks promise scalability and other improvements in the utilization of scarce spectrum resources. This motivated the network modeling and algorithms community to develop cross-layer synchronous resource allocation mechanisms [2] which, in theory, promise significant gains. Wireless ad hoc network implementations and deployments, however, have focused predominantly on asynchronous CSMA/CA mechanisms and modifications thereof. This is partially 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 describes a wireless system called FlashLinQ that, by example, demonstrates that we can design and implement a practical, synchronous MAC and PHY architecture that can support cross-layer mechanisms (e.g., backpressure [2], MaxWeight [3]) that have been proposed for network resource allocation. FlashLinQ is a new OFDM-based synchronous architecture for MAC/PHY, that *(i)* incorporates 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, from which FlashLinQ extracts fine-grained timing for network synchronization, we have implemented FlashLinQ over a *licensed* spectrum on a DSP and FPGA-based platform to demonstrate its feasibility and the significant performance benefits that accrue.

Technical Overview: FlashLinQ is a *synchronous* (time slotted) OFDM-based system that enables node discovery, channel allocation, and link scheduling with power control. This system allows for an opportunistic *fading-state aware* schedule to be re-computed each time-slot (roughly 2 msec). The key technical innovation is to leverage the physics of propagation and parallel signaling enabled by OFDM to develop a tone-matrix-based *analog* signaling scheme. This mechanism encodes, within each tone, both the presence of a signal as well as the signal strength. Leveraging this, the OFDM matrix enables the transmitters and receivers to sample the *potential* interfering links, thus enabling calculation of estimates of SIR[1] at each link (transmitter/receiver). This, in turn, enables explicit link and rate scheduling, where each link that is scheduled for transmission optimizes its data-rate to obtain an efficient spatial packing of links.

Thus, the new signaling mechanism addresses several 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); *(c)* large dynamic range in signal strengths (enabling very long links, e.g. 250 meters, to coexist in a “sea” of short links, e.g. 10 meters); and *(d)* 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 of 2.586 GHz using a bandwidth of 5 MHz. 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 reassembling, fast ARQ, etc., are also implemented in the 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

A shorter version of this paper appeared in the Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing, October 2010, [1]. X. Wu, S. Tavildar, T. Richardson, J. Li, R. Laroia and A. Jovicic are with Qualcomm New Jersey Research Center, Bridgewater, NJ, Email: {xinzhouw, tavildar, tomr, rlaroia, junyil, ajovicic}@qualcomm.com. S. Shakkottai is with the ECE Department at The University of Texas at Austin, Email: shakkott@austin.utexas.edu. Shakkottai’s research was conducted at Qualcomm New Jersey Research Center, Bridgewater, NJ.

[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.

network to extend managed services by cellular providers in licensed spectrum. This has important benefits for users as well as network providers. Users get more predictable performance, because the interference is managed, and an extended battery life, because fine-grained synchronization enables low duty-cycle operation. (FlashLinQ is designed to leverage any of CDMA/GSM cellular timing [4], DVBH timing [5], GPS timing [6], along with in-band timing.) In addition, network providers get increased spectral and power efficiency. As an aside, we note that FlashLinQ can be deployed in a mixed licensed-unlicensed spectrum configuration; indeed, licensed spectrum communication, since it is inherently more reliable, can be used as a control layer in which, for example, peers discover each other and negotiate the use of unlicensed spectrum for bulk data traffic. System design trade-offs are very different in a licensed spectrum setting (our current deployment) than in unlicensed spectrum. In particular, licensed spectrum has high capital cost and, correspondingly, the available bandwidth is usually smaller than in unlicensed bands. Therefore, maximizing spectral efficiency is a crucial design objective. Here "spectral efficiency" should be maximized not just on a link level but on a system level, where multiple links share the spectrum.

A. Motivation for FlashLinQ Scheduling

The (idealized) goal of FlashLinQ scheduling is to find, for every time slot, a *maximal* feasible subset of links, i.e., a set of transmissions that can simultaneously coexist with each other while maintaining a sufficiently large SIR, among all the (directed) links that have data to transmit. Note that the notion of "sufficiently large" depends on the desire rate. Under a simplistic binary interference model in which either a given link interferes with another link, B say, to such an extent that if it transmits then B cannot maintain adequate SIR, or the link does not interfere with B at all, a maximal subset corresponds to an independent set in a directed graph whose vertices correspond to directed links and whose edges indicate interference. Roughly, the vertices $\{A, B, ...\}$ are connected by (oriented) edges $\{(A, B), ...\}$ that indicate an exclusion condition, e.g., (A, B) indicates that link B cannot transmit if link A transmits. More generally and more realistically, a link A contributes some interference to B depending on transmit power and channel conditions. Thus, the determination of a maximal feasible subset is a complex time-varying problem since each link's SIR potentially depends on all the other links and on the desired rate. The FlashLinQ scheduler is designed to provide distributed determination of a feasible subset, ideally a maximal one.

To illustrate the importance of the SIR in link scheduling, consider 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 acknowledgment on the link Tx-A – Rx-A to fail. A similar restriction is also placed on a potential receiver Rx-B.

From communications theory we know, however, that this type of protection is, in general, neither necessary nor sufficient for optimal performance[3]: *successful decoding occurs at Rx-A as long as the SIR (Signal to Interference Ratio) is sufficiently large to permit message decoding*. Ensuring successful decoding (at adequate rate) is very different from requiring a maximum 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 and 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. Consequently, our SIR based mechanism leads to a channel-state aware maximal matching (see Section III-A) that can achieve spatial throughput gains over an 802.11g system (see Section IV).

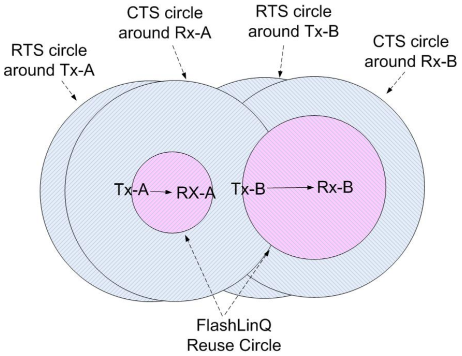

Fig. 1. Reuse radius in FlashLinQ vs. 802.11 (CSMA/CA with RTS/CTS).

[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].

II. Related Work

Distributed scheduling in wireless networks has attracted attention of many researchers over the last several years. Interesting results and insights have been obtained concerning the potential throughput loss suffered by a class of *maximal matching* distributed scheduling algorithms, as compared to a genie-aided centralized algorithm [9][10][11]. Various ways to improve the maximal matching have been proposed [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 assume a combinatorial interference model at the physical layer and focus on scheduling links given the *feasible* independent sets, i.e., subsets of links are allowed to transmit simultaneously according to the combinatorial interference model. The possibilities raised by defining feasible independent sets using actual SIRs under fading channels (channel coefficients that can change on a per-time-slot basis) and then incorporating multiple power-levels and transmission 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 cancellation into 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 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 an 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, exploiting 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 ideas similar to those 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 via the use of out-of-band control channels was proposed in [23]. 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

The FlashLinQ peer-to-peer system has been designed to operate synchronously in 5 MHz of bandwidth, with protocols enabling distributed, channel-aware spatial scheduling. The underlying physical layer technology used for FlashLinQ is OFDM/OFDMA. OFDM is a well-known frequency-division multiplexing (FDM) scheme. OFDM uses a digital multi-carrier modulation method that effectively reduces the channel to a set of orthogonal elementary complex degrees of freedom organized into a set of distinct carriers (tones). OFDMA is a multi-user version of OFDM that exploits the physical layer orthogonality to easily orthogonalize users. OFDM/OFDMA has been the underlying technology for many advanced wireless systems such as 802.11g [20], LTE, and WiMAX. We refer to [24] for an excellent tutorial on OFDM and OFDMA.

OFDMA requires *timing synchronization* among multiple devices, whose application is extended in FlashLinQ to provide a foundation for efficient operation of all of its key aspects. Synchronization underpins the entire system by providing the basis for OFDM orthogonal signalling and the ability for all devices to operate simultaneously with low duty cycle.

There are two key aspects of FlashLinQ's distributed resource allocation protocol that are founded on the OFDMA signal structure.

1) **Signaling Mechanism**: We exploit the flexibility of parallel, single-tone channels afforded by OFDM to construct 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 were to transmit data, but without actually spending the resources needed to perform the data transmissions and without realizing the resulting contentions.
2) **Spatial Packing**: By carefully 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 degradation to the SIR that it causes at each receiver. This enables a feasible set of transmissions to be determined taking into consideration the link qualities that result from choosing this feasible set. As we discussed earlier (see Section I-A), this is critical to ensure efficient spatial packing.

The scheduling operation (feasible set selection) occurs every 2.08 msec in FlashLinQ (see Figure 2 for a timeline).

On a slightly longer time scale, there occur the following two processes fundamental to the FlashLinQ system.

1) **Peer discovery**: This enables nodes to transmit presence information and detect the presence of other nodes in the neighborhood.
2) **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).

In the remainder of the section, we present a detailed description of scheduling and resource allocation, followed by an abridged description of timing synchronization, peer discvoery and link management.

A. Scheduling and Data Transmission

This section contains the main technical contribution of the paper: a low-overhead distributed scheduling algorithm. We

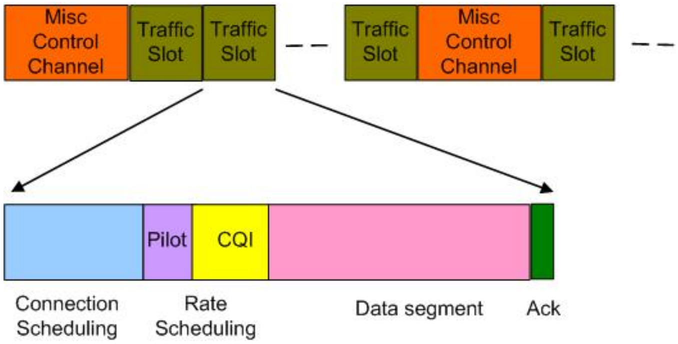

Fig. 2. The FlashLinQ operation timeline: data transmissions occur in slotted time of around 2 msec each. Within each slot, link and rate scheduling is followed by the actual data transmission. In addition, every 1 second, resources are allocated for other channels such as peer discovery and link management.

first describe the key ideas underlying our algorithm and then describe the signaling mechanisms that enable the approach.

As discussed in Section I-A, the goal of FlashLinQ is to schedule a channel-state aware *maximal* feasible set of links for any given time slot based on the current traffic and channel conditions. The feasible 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 present the key elements of the algorithm, we first consider a simple two-link example. We consider links $A \to B$ and $C \to D$ as shown in Figure 3. Here, the two links[4] have direct-link gains $\{|h_{AB}|^2, |h_{CD}|^2\}$, and cross-link gains, $\{|h_{AD}|^2, |h_{BC}|^2\}$. If the cross-link gains are small compared to the direct-link gains, then the links AB and CD will not significantly interfere with each other and can therefore be simultaneously scheduled. On the other hand, if the cross-link gains are relatively large, then only one of the links can be scheduled at any instant of time.

If only one link can be scheduled, there arises the problem of selecting that link. A simple way of resolving the ambiguity while also providing a basis for determining the interference scenario is to assign *priorities* to links. The assumption is that, in this setting, the higher priority link will be scheduled. The low priority link, however, will not be scheduled only if its transmission will cause excessive *SIR damage* to the high priority link. This is determined by comparing the *would-be SIR* of the high priority link with an *SIR threshold* assuming the low priority link does in fact proceed with its data transmission.

To be more precise, in the two-link example in Figure 3, assume link $A \to B$ has higher priority in the current slot. Link $C \to D$ can potentially be scheduled simultaneously if *C doesn't cause too much interference to B*. To define this we require the SIR of link $A \to B$, assuming C is transmitting (and ignoring other potential interferers), to be at least γ_{TX} dB[5]. Thus, the protection condition can be written as:

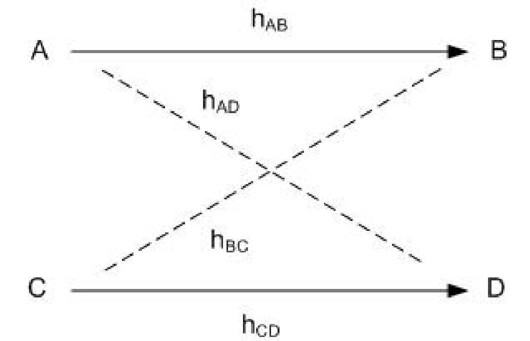

Fig. 3. Scheduling for two links

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

where P_A denotes transmit power used by node A and P_C denotes transmit power used by node C. In our system, power control of the links is performed on a time scale slower than scheduling so transmit 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. We remark that a final optimization of link usage prior to data transmission is effected through rate selection, which occurs after connection scheduling.

Condition (1) provides SIR protection for the higher priority link from the lower priority link. But even if the condition is satisfied it does not mean that the lower priority link should transmit. In the two link setting there is no reason for the low priority link not to transmit, except perhaps to conserve power. In a more general network with multiple links, however, there may be system throughput gains that result from if, under certain conditions, the link does not transmit. Consider the setup in Figure 4. The transmission $A \to B$ is adequately protected from link $C \to D$ but, since D is close to A, the SIR seen by D will be low. In such a scenario, we say that the link $C \to D$ should NOT transmit: This will potentially allow for a much better spatial link packing since another link, such as $E \to F$, can alternatively be scheduled, allowing for

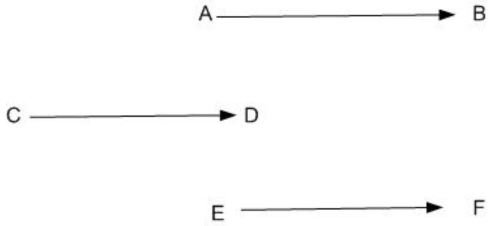

Fig. 4. Scheduling for three links

[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.

[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

higher system throughput. Therefore, we impose an additional condition for a link $C \rightarrow D$ to be scheduled: D *should expect adequate SIR if scheduled.* We define the condition by requiring that the SIR of link $C \rightarrow D$ due to A's transmission at be least γ_{RX} dB: [6]

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

These two mechanisms ensure, in the two link scenario, 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, a basic level of "fairness" across links can be maintained in the sense that all links will have access to the spectrum with a scheduling probability of at least $\frac{1}{|\mathcal{N}_l|}$, or the inverse of the neighbor size of link l. Here two links are said to be neighbors of each other if they cannot be scheduled simultaneously under the constraints of (1) and (2).

So far, we have illustrated the following three key elements of our algorithm in the simple network examples 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. The system design needs to provide the relevant devices a means to check these criteria. Device 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 signalling protocol by which C and D infer the relevant information based on minimal transmissions from A and B. In particular, the protocol does not require any dedicated signaling between links AB and CD. The main mechanism used to enable distributed determination of the above two criteria by providing the information needed to estimate various SIRs is a two analog-tone-signal exchange consisting of a *inverse power echo* and a *direct power signal*. The two signals will be described in the context of Figure 5.

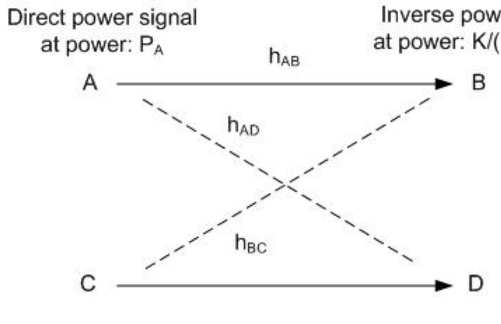

Fig. 5. Direct power signal and inverse power echo

The *direct power signal* is sent by a transmitter to allow all receivers to estimate the power received from that 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.

The *inverse power echo* is sent by a receiver to allow all other transmitters to estimate the ratio of the received power of their transmission to that of the receiver's intended transmitter. *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 now form the SIR estimate:

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

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

2) Algorithm description: In the network setting, we consider a cascaded scheduling algorithm where the priorities of the links are arranged in a pseudo-random order, and links scheduled in a sequential manner. (We will discuss the acquisition process of such a pseudo-random priority in a later section III-A3.) 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. It will decide to transmit under the following conditions.

- *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 where the interference is taken 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 time slot ensures a basic level of fairness across links as we discussed before.

Note that this algorithm does not guarantee a precise minimum SIR for a scheduled link. There are potentially multiple interfering transmitters, each of which individually guarantees a certain SIR to a receiver, that combined together can cause the actual SIR to fall below the yielding threshold. (In addition, we have so far ignored thermal noise at the

[6] In our implementation, we use a value of 9 dB for γ_{RX}. As before, this value was chosen for optimizing spectral efficiency.

receiver.) This is addressed by using a rate scheduling mechanism that subsequent to the scheduling operation performs 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 determine 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 being 2.08 milliseconds in 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): connection scheduling, rate scheduling, data segment and ACK, each described below.

Connection scheduling: Each link has an associated locally unique connection ID (or CID) which is an index between 1 and 112, which is acquired as part of the link management process. When there are more than 112 links present in a collision domain, new devices cannot find a free CID in the "slower" link management, and thus cannot participate in the FlashLinQ schedulign phase (see Section III-D for more details on CID acquisition). They however periodically probe for a free CID; when available, they can capture it and then participate in connection scheduling and data transmission as described below.

For every slot, connection scheduling signaling consists of two signaling blocks: Tx-block and Rx-block, as shown in Figure 6. Both Tx-block and Rx-block have four OFDM-symbols with FFT size 32, i.e, each symbol has 32 different tones (or sub-carriers), out of which, 28 are usable for signal transmission. A link with an assigned CID corresponds to a pair of single tones, one in Tx-block, to be used for transmitter, and the other in Rx-block, to be used for receiver. The mapping from the CIDs to the actual tone pair within the connection scheduling blocks 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 scheduling algorithm described earlier in this section requires *(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. To elaborate, each link has a Connection ID (CID) that is associated with it in the link management outer control (see Section III-D). The priority of links is driven by this Connection ID. This connection ID is mapped to a priority level (that is time-varying). Since two links in the same collision domain have different Connection IDs, this ensures that tones are not in conflict if the mapping is unique (i.e., two links do not take the same priority). There are two implementations of the mapping between the CID and priority – one, via a deterministic table mapping CID to priority-level that is hardwired into each device, and another via a global pseudo-random number generator in the FlashLinQ system (i.e., same algorithm and seed at each device).

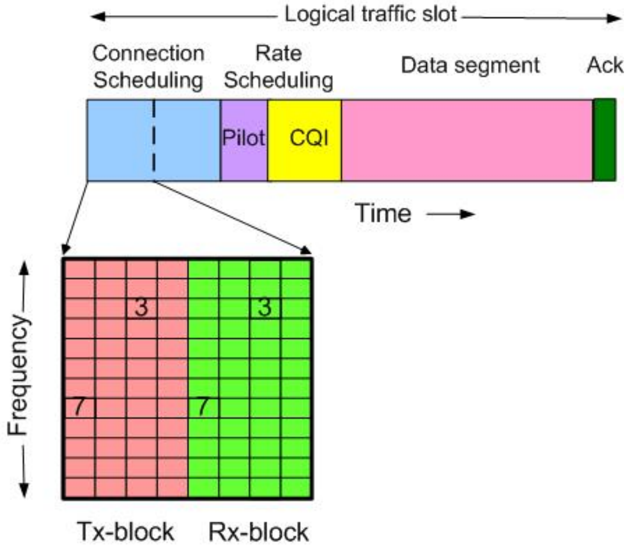

Fig. 6. Structure of connection scheduling

To use the language of WiFi, the Tx-block is analogous to the RTS block and the Rx-block is analogous to 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 and tone 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 and tone using its *inverse echo power* level described earlier in this section. All the transmitters listen to all the tones and symbols 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 and tone 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 wins 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 connection 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 *simultaneously* 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 connection scheduling, because each link's code rate and modulation is chosen based on the actual SIR corresponding to the final outcome of the scheduling mechanism.

Data segment: All scheduled links transmit over all tones. Note that single-tone signals are used only during connection 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.

4) Advanced features of the design: In the previous sections, we described the baseline design of FlashLinQ connection scheduling where multiple links can share the bandwidth in a *fair* way. This baseline design can be easily enhanced to support many advanced features, including support for QoS, MIMO scheduling, frequency band splitting and multicast/broadcast messages. Here we describe the main ideas to support these features in FlashLinQ, showing that they do not require significant additional overhead.

QoS Design: The baseline FlashLinQ connection scheduling protocol enforces a random priority allocation among links. Over time, this mechanism makes sure all links obtain a similar share of the channel use. However, it is desirable for the system to be able to give higher priority to certain links over others and support hierarchical QoS levels. Towards this end, the tone matrices that we have described are split into multiple sub-blocks representing different priority levels (priority ordering across blocks of tones). A link is assigned multiple tone-pairs – at any timeslot, the choice of which tone-pair to use dynamically depends on the queue-length/backlog or packet delay. We have used this mechanism to study a simple version of the back-pressure algorithm for tone-pair selection, and have observed the anticipated QoS performance gains.

MIMO Design: Proper use of multiple antennas can bring tremendous gains to ad hoc networks and, moreover, these gains are *much easier* to obtain in such networks as compared to the traditional cellular communications. First, all antennas are at low ground level and the *angular spread* between two users is typically larger than the cellular case, where the base station antenna is usually placed 30m above the ground. Large angular spread is critical to enable spatial multiplexing between a Tx-Rx pair. [24]. Second, the channel matrix information is easier to obtain due to TDD nature of the channel in ad hoc networks. Third, the restricted association nature between the transmitter and receiver creates a rich interference environment. Even simple beamforming schemes can reduce the *protection circle*, i.e. the area where no other links can co-exist, of any link and thereby bring a gain in *spatial reuse*, especially in a congested environment. The parallel analog signaling in the current FlashLinQ connection scheduling can be naturally used to enable MIMO transmissions (including spatial multiplexing and beamforming) as well. In particular, we modulate the direct and echo powers by the beamforming vectors, e.g. let multiple transmit antennas of a MIMO-enabled transmitter transmit at the assigned tone simultaneously using a phase vector determined by the transmit beamforming vector. Similarly, receivers modulate the echo signal using the receive beamforming vector so that surrounding neighbors can make correct measurement of the would-be SINR damage of their transmissions at one receiver. Further, to enable spatial multiplexing between one transmitter-receiver pair, they can acquire multiple tone-pairs where each tone-pair contends for *one* of the data streams with other users in the system, using corresponding transmit and receive beamforming vectors.

FDM Design: In the baseline design, all scheduled links transmit simultaneously over all tones. Instead, one can potentially split the data segment into multiple parallel blocks (frequency band splitting into four blocks of 1.25 MHz each) – we have observed that this is useful (can potentially double the measured throughput) in settings with a few very long links (e.g., 500 meters), requiring a low rate, and many short links, requiring higher rate. This band-splitting modification enables simultaneous scheduling of long links with the short links by using a simple modification of the scheduling algorithms described above.

Multi-tone Design for Mitigating Frequency Selective Fading: As mentiond earlier, single tone signaling has low *PAPR*, which can be exploited in connection scheduling by increasing transmit power. On the other hand, connection scheduling with single tone signaling might suffer from *frequency selective fading*, e.g., the channel on the selected tone might be in a deep fade and the power measured on that tone would then poorly represent the overall channel quality. When FlashLinQ is considered for deployment in environments with extensive *multipath delay spread* a multi-tone connection scheduling should be considered. In this case, a CID corresponds to multiple tone pairs, e.g. four, that are "spread-out" across the frequencies within the tone matrix. Yielding decisions will then be based on the *average* measured energy over the multiple tone-pairs. This mitigates frequency selective fading and improves the SINR estimate made during connection scheduling. The rate scheduling signals use wide-band pilot signals and are therefore inherently robust against frequency selective fading.

Broadcast Design: Another important feature of the FlashLinQ design is its support of reliable broadcast and multicast. A characteristic capability of the FlashLinQ system is the *peer discovery*, see Section III-C. During the peer discovery phase, mobiles exchange identity and interests with each other; the peer discovery channel is essentially a light-weight broadcast channel. The data rate for the peer discovery channels is intentionally quite low (72 bits per user every 8 seconds) to ensure low system overhead, low power consumption, and relatively large range. In addition to peer discovery, to support higher rate broadcast messages, FlashLinQ can provision additional periodic traffic resources for dedicated broadcast. For example, one out of 25 traffic slots can be designated broadcast slots. During these traffic slots, the data segment is further partitioned into multiple, e.g. 9, transmission blocks. Multiple transmitters compete separately in corresponding connection scheduling blocks for the use of these transmission blocks. The connection scheduling block is also tailored for the broadcast nature of the transmissions, in which traffic from a single transmitter is supposed to reach multiple receivers. Overall, in addition to the peer discovery capability, FlashLinQ can support up to 1800 bits/s of broadcast messages for each transmitter.

B. Timing Synchronization

Ad-hoc networks are usually conceived as operating largely asynchronously in unlicensed spectrum. Synchronization brings many advantages but achieving synchronization in a distributed wireless system is a resource and energy intensive problem. In the FlashLinQ system, which operates in licensed spectrum, we achieve synchronization using existing infrastructure. In particular, we use CDMA systems deployed in the US that send *unscrambled* signals from which fine-grained timing and frequency information can be derived. In Europe, DVB-H is a potential source for such timing and frequency information. In addition, one can envision deploying an in-band timing source explicitly for this purpose (we have done this for our prototype system, see Section IV).

However, base stations may not be completely synchronized to each other, which may lead to timing offsets between nodes on cell boundaries, causing FLQ devices to fail to detect timing signals in the presence of channel fading. Thus, residual timing errors can remain even with infrastructure based synchronization. We address this problem using a two step approach:

- *Secondary timing synchronization:* this protocol works in the FlashLinQ spectrum and is used to correct the timing errors between different nodes after synchronization to the base station. Note that the secondary timing synchronization protocol solves the problem of distributed synchronization for the case when all nodes have already achieved rough synchronization. This is a much simpler problem than a completely distributed synchronization problem.
- *Larger cyclic prefix (CP)*: The residual errors after the secondary timing synchronization protocols are accommodated by using a larger CP for the OFDM system, one that covers for propagation delay as well as timing errors.

For our system, the timing synchronization algorithm achieves timing synchronization to within the propagation delay (maximum of 5 microseconds) between different nodes.

C. Peer discovery

The peer discovery mechanism enables nodes to discover the presence of other nodes in their neighborhood, which is about a 1 km range. This long range is facilitated by using rateless codes to broadcast peer IDs, enabling discovery at extremely low SNR over moderate timescales. Here, we briefly summarize two important aspects of peer discovery. The details are not discussed in this paper since the focus of this paper is on the scheduling algorithm.

Energy efficiency: One of the chronic concerns about peer discovery in ad hoc wireless networks is the energy efficiency of the process. We use synchronization (Section III-B) to dedicate small time slots for the purpose of peer discovery (roughly 20 ms per second). This time slot is used for transmitting as well as receiving presence information to and from nearby nodes. In our system, this amounts to a 2% active duty cycle, which leaves an acceptable device standby time. All devices participate in peer discovery even if they are not actively communicating with other devices.

Discovery performance: The peer discovery time slot is partitioned into orthogonal resources using the robust orthogonalization inherent in OFDM: each resource uses a subset of the tone-time matrix determined by OFDM for the peer discovery slot. A node monitors the peer discovery channel and picks a locally unused resource to transmit on. The node then sends coded information using codes similar to fountain codes. In our study, we have seen that the protocol can successfully discover up to a few thousand devices over a 1 km radius and within, roughly, a 10 - 15 second time interval.

D. Link management

After a device detects another device of interest in peer discovery, it may page that device to establish a *link* between the two devices. The paging channels are a set of time recurring resources (every 1 second) which support multiple simultaneous pages with low failure probability. As part of the paging process, the transmitter and receiver exchange user identity and other authentication information to establish a link. Another important component of the link establishment process is the acquisition and determination of a *locally unique* connection identifier (CID) that is used in connection scheduling to identify the link. For managing this resource we provision a special channel, the CID channel, which operates in conjunction with paging, This allows devices to monitor CID usage and identify a viable resource using the protocol described below.

We assume that links remain in the system for long periods of time (a few seconds) relative to a communication burst (2.08 ms). This assumption is used to allocate distinct link identifiers (CID) to each (local) link, that are used during the scheduling process. Note that, even though we assume that the links communicate over relatively long periods of time, the traffic itself may be bursty in nature. As described below, CIDs are selected based on a SIR criterion that is analogous to that used in link scheduling.

Before we describe the CID selection algorithm, we need to first address the meaning of local uniqueness in this context.

We define local uniqueness in terms of SIR rather than interference. The motivation for this is similar to that for the link scheduling algorithm: for a wireless system, the criterion of importance is the SIR, and not merely the absolute signal and interference levels.

To be specific, consider two links $A \leftrightarrow B$ and $C \leftrightarrow D$. The various channel gains associated to them are shown in Figure 7. The transmit powers used by nodes are denoted as P_A, P_B, etc.

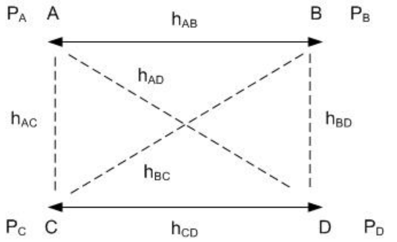

Fig. 7. CID Reuse

We say that a link $A \leftrightarrow B$ can reuse a CID with a link $C \leftrightarrow D$ ($A \leftrightarrow B$ is not local to $C \leftrightarrow D$) if the following conditions are satisfied:

$$\min(\frac{P_B|h_{AB}|^2}{P_C|h_{AC}|^2}, \frac{P_B|h_{AB}|^2}{P_D|h_{AD}|^2}) > SIR_{threshold}, \quad (4)$$

$$\min(\frac{P_A|h_{AB}|^2}{P_C|h_{BC}|^2}, \frac{P_A|h_{AB}|^2}{P_D|h_{BD}|^2}) > SIR_{threshold}, \quad (5)$$

$$\min(\frac{P_D|h_{CD}|^2}{P_A|h_{AC}|^2}, \frac{P_D|h_{CD}|^2}{P_B|h_{BC}|^2}) > SIR_{threshold}, \quad (6)$$

$$\min(\frac{P_C|h_{CD}|^2}{P_A|h_{AD}|^2}, \frac{P_C|h_{CD}|^2}{P_B|h_{BD}|^2}) > SIR_{threshold}, \quad (7)$$

where $SIR_{threshold}$ is a constant, set to be 20 dB for our system. The motivation for this definition is as follows: equation (4) says that the SIR seen by A when B is transmitting to A, and simultaneously C is transmitting to D, or D is transmitting to C is at least $SIR_{threshold}$. Similarly, equations (5), (6), and (7) ensure that the SIRs seen by B, C, and D should be at least $SIR_{threshold}$. If all these equations are satisfied it means link $A \leftrightarrow B$ does not interfere significantly with link $C \leftrightarrow D$, and hence they can use the same CID. Note the $SIR_{threhold}$ (20dB) here is chosen to be much larger than the scheduling yielding threshold γ_{TX} and γ_{RX} (9dB) as defined in (1) and (2). Roughly speaking, the CID selection protocol ensures that the CID is not reused by another link within the *two-hop* neighborhood of any link.

This raises the question of how to design a distributed system that allocates CIDs so that equations (4) to (7) are satisfied for every pair of links. The main idea here is to use analog signals: *inverse power echo* and a *direct power signal*, described in Section III-A. We assume that for each link $A \leftrightarrow B$, both A and B know the powers P_A and P_B as well as the channel gain h_{AB} (similarly for $C \leftrightarrow D$; powers are known due to FlashLinQ design parameters, and channel gains are learned analogous to connection scheduling).

The protocol uses a dedicated signaling resource for CID selection. The resource occupies roughly 1 millisecond every second. The resource is partitioned into orthogonal components, using OFDM tone-time matrix, one for each CID. Each component contains four sub-resources (elementary complex degrees of freedom). The protocol followed by an active link (say $A \leftrightarrow B$) that has a CID is as follows:

1) A uses two sub-resources to send out two signals that indicate that the CID is currently being used: a direct power signal and an inverse power echo.
2) B uses the other sub-resources to send out a direct power signal and an inverse power echo.

The protocol followed by a link (say $C \leftrightarrow D$) that wants to grab a CID is as follows. C monitors all the sub-resources for each CID. For each CID, based on the power received, it decides whether that CID can be used while still satisfying equations (4) and (5). This is done as follows:

1) Using A's direct power signal, C estimates $\frac{P_D|h_{CD}|^2}{P_A|h_{AC}|^2}$.
2) Using A's inverse power echo, C estimates $\frac{P_C|h_{AC}|^2}{P_B|h_{AB}|^2}$.
3) C similarly uses the two signals sent by B.

D follows a protocol similar to C's protocol described above. Based on this, C and D both jointly determine through a message exchange for each CID if the equations (4) to (7) are met for a given CID and then agree upon a CID.

E. Comparison with 802.11

In this section, we summarize some of the key differences with the 802.11g (CSMA/CA with RTS/CTS) protocol. These benefits will be quantified through measurements and simulations in Section IV.

1) **Synchronization**: FlashLinQ is a time slotted system. This allows FlashLinQ to have dedicated slots for connection scheduling as well as rate scheduling. The direct impact of this is that it reduces system overhead. A more indirect benefit for FlashLinQ is that it enables many algorithms that are hard to implement in an asynchronous system. Arguably, one can incorporate many of the ideas in this paper into 802.11, but the asynchronous nature of 802.11 makes the implementation significantly more difficult, and the resulting gains much lower.
2) **Tx-Rx yielding**: In FlashLinQ, transmitters yield only based on receiver echoes and do not yield to other transmitters. Similarly, receivers yield only to other transmitters. This is in contrast with 802.11 where transmitters and receivers both yield to transmitters (CSMA/CA) or transmitters and receivers both yield to transmitters as well as receivers (RTS/CTS). The FlashLinQ approach enables more spatial reuse, and solves the hidden node problem without a spatial reuse penalty.
3) **Spatial reuse**: In 802.11 the reuse decisions are made based on sensing, meaning that the reuse radius is fixed and independent of the length of the primary link. Further, the reuse region is drawn around the transmitter and excludes both transmitters and receivers[7]. This makes the 802.11 reuse decision highly suboptimal, particularly

[7] This is true in the case of CSMA/CA. If RTS/CTS is incorporated, the reuse region is around *both* transmitter and receiver, and excludes other transmitters and receivers

for short links. On the other hand, in FlashLinQ the reuse radius depends on the primary link length; the shorter the primary link, shorter the reuse radius. This is illustrated in Figure 1.

4) **Power control**: In FlashLinQ, short links can use lower transmit power and hence can coexist with long links that use higher transmit power. This is akin to being able to whisper (short range communication) in a large lecture hall without interrupting the lecture for other audience members (long range communication). In Figure 8, C can transmit to D at lower power since D is close to C thereby allowing A to transmit to B simultaneously. We discussed the power control issue in much greater details in a separate paper [25].

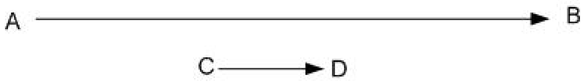

Fig. 8. Power control facilitates reuse

5) **Rate scheduling**: In FlashLinQ, we have dedicated per slot rate scheduling in which interference estimation is done before every transmission. This provides much more robust rate scheduling than 802.11 rate scheduling which is typically based on ack/nak. It is particularly useful in a dynamic interference environment.
6) **Range**: From a link budget point of view, FlashLinQ's traffic link is supported at 14 dB lower power than 802.11, thus inherently supporting longer links. This corresponds to 2X-3X range improvement depending on the propagation environment. From a scheduling perspective, the FlashLinQ mechanism enables longer links to periodically "win" a chance to get scheduled – in 802.11, it is hard to sustain long links due to a lack of MAC-level coordination.

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 9) 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 sample rate processing and FFT are performed in the FPGA and frequency domain symbol level processing is performed mainly in the DSP. As a result of this separation, the link scheduling algorithms reside in the DSP. Further implementation details are available in [26].

Fig. 9. 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 10) 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 measured energy levels of the interference, 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.

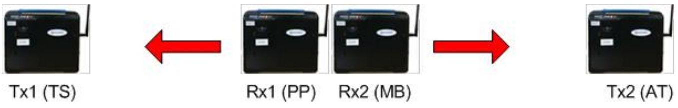

Fig. 10. Reuse versus orthogonalization experiment setup

To collect the data, the modem reports its current link scheduling status to the Linux based host every second. Figure 11 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 11, 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. We remark that in these experiments, we only enabled low rate options with at peak rate limit of $1.5 Mbits/s$.

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 energy-measurement based yielding, e.g., the carrier sensing protocol in 802.11, and also full

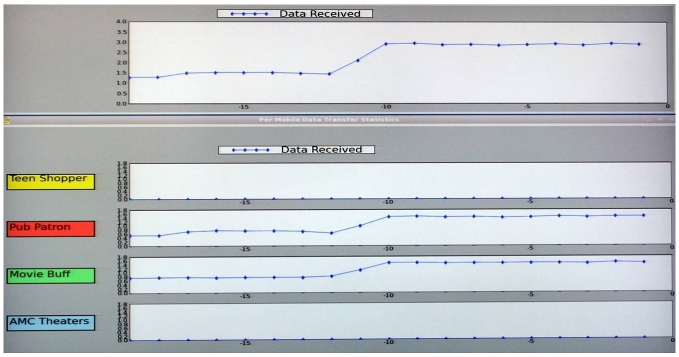

Fig. 11. Measurement data as a time series.

reuse. 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 12. 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.

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.

Before discussing the simulation settings, it is important to note that there are several design modifications that can be made to 802.11g, such as out-of-band SIR based signaling [23], power and rate control, etc., that can lead to improved performance of the 802.11g system. However, as our simulations are based on a detailed software implementation, this would require us to fully spec-out such a system, which is beyond the scope of our work here. In any case, our main objective here is to demonstrate that we can indeed design a synchronous and distributed opportunistic scheduling system that is competes well with traditional 802.11g systems, even after accounting for all signaling overheads.

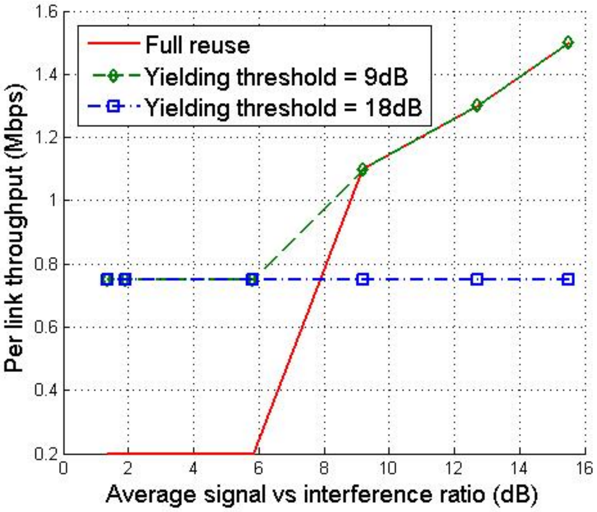

Fig. 12. Lab measurement data with different yielding thresholds.

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 use a wrap-around model (torus) in the signal strength calculation between any two nodes. The pathloss between any two nodes is modeled using an ITU-1411 LOS model with antenna height of 1.5 meters (see [27] 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. Fast fading is not modeled in the simulations. 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 13 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. The sum throughput of the network under these two protocols are shown in Table I, where an increase of 450% can be seen in sum spectrum efficiency with 256 links by using FlashLinQ for both indoor and outdoor deployments.

	Indoor	Outdoor
FlashLinQ	125	17.3
802.11g	21.2	3.2

TABLE I
SUM RATE (BITS/SEC/HZ) FOR 256 LINKS.

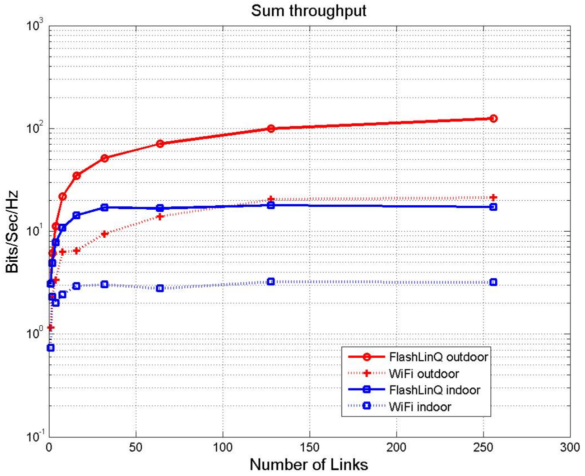

Fig. 13. Sum throughput comparison

Next, we present comparison of the CDF of the individual links (see Figure 14). We see that there is a uniform improvement of link rates. Moreover, since FlashLinQ does not suffer from the hidden node problem, we see that the tail performance has improved significantly for FlashLinQ. This effect is particularly pronounced for the indoor scenario. This is because a number of links are dropped with receiver and transmitter across different floors thereby increasing the probability of hidden nodes. This impacts the link performance significantly since WiFi, even with RTS-CTS, cannot effectively deal with the hidden node problem due to asynchronous nature of WiFi.

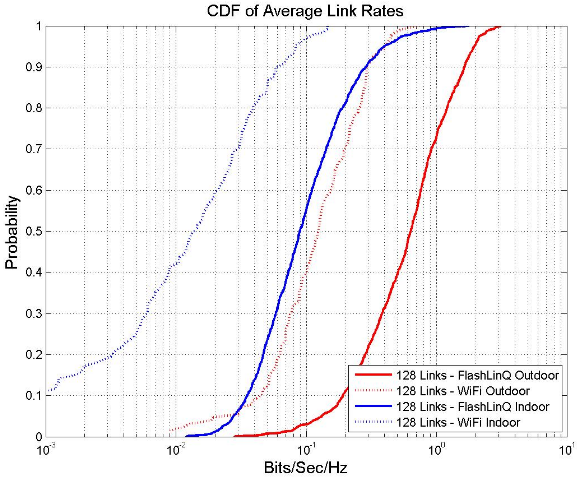

Fig. 14. Rate CDF comparison between WiFi and FlashLinQ for indoor and outdoor scenarios

Finally, we compare the CDF of average scheduling latency (scheduling latency = inter-schedule delay, see Figure 15) among all links. Here we see that for the outdoor scenario, WiFi has similar performance as FlashLinQ for most of the links, in terms average inter-schedule delay, despite the four-fold bandwidth advantage of WiFi. WiFi has a slight advantage over FlashLinQ for users with few interfering neighbors. This is because in FlashLinQ, the inter-schedule delay is lower bounded by the slot size 2ms, while in WiFi, the inter-schedule delay can be much smaller due to larger bandwidth and limited maximum packet size. On the other hand, for the indoor scenario, we observe a significant improvement with FlashLinQ on the tail performance. This is due to the insufficient link protection under the WiFi CSMA protocol, which leads to starvation for some links with a large amount of interfering neighbors.

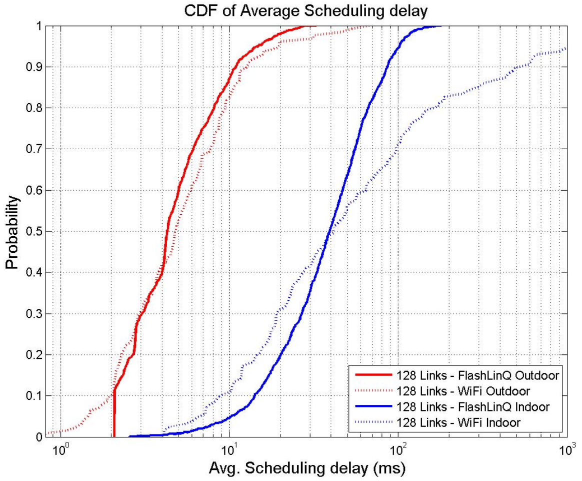

Fig. 15. Latency (inter-schedule delay) CDF comparison between WiFi and FlashLinQ for indoor and outdoor scenarios

V. CONCLUSIONS

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

Finally, we comment that FlashLinQ is by no means optimal, and that there are several other design optimizations that can be made in 802.11 systems to improve performance as the vast literature in this research area indicates. However, FlashLinQ demonstrates that we can indeed architect, design and implement a fast (timeslot-by-timeslot) *channel-aware opportunistic* synchronous system, that accounts for all signaling overheads and results in gains over a conventional 802.11 system. This is of interest, given the considerable interest in slotted-time opportunistic scheduling that is an active area of research today, and indicates that such systems may be a viable alternative to 802.11-based systems.

References

[1] 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. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, October 2010.

[2] 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.

[3] 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.

[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 *Proceedings of IEEE GLOBECOM*, 2002.

[8] F. Ye, S. Yi, and B. Sikdar. Improving spatial reuse of IEEE 802.11 based ad hoc networks. In *Proceedings of IEEE 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 *Proceedings of 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 *Proceedings of IEEE 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 *Proceedings of ACM SIGCOMM*, pages 397–408, 2007.

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

[18] S. Gollakota, S. Perli, and D. Katabi. Interference alignment and cancellation. In *Proceedings of ACM 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 *Proceedings of 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 packet 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 *Proceedings of 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] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge University Press, Cambridge, UK, 2005.

[25] François Baccelli, Junyi Li, Thomas Richardson, Sundar Subramanian, Xinzhou Wu, and Sanjay Shakkottai. On optimizing csma for wide area ad-hoc networks. In *WiOpt*, pages 354–359, 2011.

[26] 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.

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