A Concurrent Acknowledgement Scheme for Broadcast Messages in Wireless Networks

Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker

CU-CS-1046-08 September 2008

University of Colorado at Boulder

Technical Report CU-CS-1046-08
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

A Concurrent Acknowledgement Scheme for Broadcast Messages in Wireless Networks

Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker

September 2008

Abstract

Network protocol designers, both at the physical and network level, have long considered interference and simultaneous transmission in wireless protocols as a problem to be avoided. This, coupled with a tendency to emulate wired network protocols in the wireless domain, has led to artificial limitations in wireless networks.

In this paper, we argue that wireless protocols can exploit *simultaneous transmission* to reduce the cost of reliable multicast by orders of magnitude. With an appropriate application interface, simultaneous transmission can also greatly speed up common group communication primitives.

The proposed method only functions on a *network* of directly reachable nodes in a wireless network rather than an internet of connected networks; however, this is precisely the domain where many such group communication mechanisms are needed since they are commonly used in routing protocols and other physical-layer mechanisms such as leader-election protocols.

We demonstrate how simultaneous transmission can be used to implement a *reliable broadcast* for an infrastructure and peer-to-peer network using a prototype implemented using a software defined radio. We also illustrate how to over come problems with the technique such as the "near-far nodes" problem as well as handling an arbitrary number of nodes. We then describe how the mechanism can be extended to group communication problems and the challenges inherent in that design.

1 Introduction

Noise and interference are fundamental aspects of communications, and are exceptionally important for wireless communications because it's more difficult to contain propagation without waveguides such as wires and fibers. Typically, protocol designers view interference or noise as properties to overcome and avoid. This fundamental design objective limits the scope of simultaneous multi-user communication. Conventional single carrier communication focuses on decoding the strongest signal while discarding anything else as noise or interference.

Multi-user communication requires some form of *orthogonal channel* for modulation that allows multiple parties to communicate simultaneously. There are a number of ways to implementing orthogonal channels - code division multiple-access (CDMA) has been adopted as a very reliable multiple access techniques by using specially designed *codes* with strong auto-correlation properties. With spatial frequency reuse, frequencies are allocated in a way such that signals from far away communicating pairs originate will be so strongly attenuated that they won't interfere in local communication. Time division multiplexing, or taking turns using a channel, is another method.

In this paper, we focus on using *orthogonal frequency division modulation* (OFDM) to provide distinct orthogonal signals. While CDMA uses orthogonal codes to allow multiple transmission at the same radio frequency, OFDM is a mechanism that splits the available spectrum into a number of orthogonal non-interfering *subchannels*. Being orthogonal, each of the subcarriers can be treated as information carrying medium without significant interference with another subcarrier. Variants of the OFDM waveform are used in a number of current wireless (and wired) physical layers, including the 802.11g, 802.11a.

Each of these orthogonal communication mechanisms can allow us to receive data from multiple distinct parties using additional processing. For example, in a time division network, different nodes can transmit in different assigned time slots. In a frequency division method, different nodes can transmit on different frequencies at the same time. CDMA cell phones all transmit simultaneously at the same frequency and time, but because they use different codes, the cellular receiver can decode the different communication streams. Under OFDM, different nodes can also communicate on different *subcarriers*, as uses in WiMax, which employs "scalable OFDMA" where users use different subcarriers or set of subcarriers to transmit data over the same medium and at same carrier frequency.

The ability to be able to distinguish multiple simultaneously transmitted signals is empowering in communication protocols, but such signals typically need to be fairly simple.

For example, think of asking a room full of people if they ate breakfast that day. Individuals could respond using voice, but humans have a hard time distinguishing all the streams of information. However, if people raise their hands instead, it's immediately clear who has and has not eaten breakfast. However, it's hard to get complex information, such as *what* someone had for breakfast, since the set of possible responses are so large.

Simultaneous transmissions can be an advantage in a number of network applications that call for multiple nodes to participate and also use simple information, like route requests, leader election, network management and other operations involving broadcast or multicast messages. Not only does simultaneous transmissions make the message exchange faster, its also allows such exchanges to be reliable.

In this paper we demonstrate the various application of simultaneous transmissions using multicarrier orthogonal modulation techniques and also investigate the feasibility of such a system by implementing the protocol in a prototype hardware platform.

The rest of the paper is organized as follows. Section 2 provides some of the key uses of simultaneous transmission along with a description of the proposed protocol. The next section deals with the implementation details using a custom hardware. Future works and idea extrapolation is done in section 4 and we conclude in section 5.

2 Using Simultanous Transmissions

In this paper, we focus on speeding *group communication* using *simultaneous transmission and reception*. There are many types of group communications, the most common of which is broadcast or multicast Conventional infrastructure wireless networks (*e.g.* a standard WiFi network) usually only use broadcast packets to translated wired broadcasts into wireless packets. The standard 802.11 physical layer doesn't provide a method for determining if a broadcast was delivered; thus such broadcasts are typically transmitted at the lowest modulation rate (in an effort to increase the reliability of reception). Since broadcast messages are exchanged without acknowledgement control frames, there is a limited scope for the source or the access point(AP) to reliably ensure the reception of the message at the host nodes.

In “ad hoc” networks, broadcast messages are used in for many purposes. Typical application include host discovery, network maintenance, route discovery, etc. For example, wireless protocols such as AODV [5] periodically broadcast a routing table to “neighboring nodes” (meaning those that can hear the message). Node also periodically transmit “hello” messages to determine if nodes are still reachable. These messages are typically “unicast” messages, because there is no way to safely determine if they’ve been received.

Reliable broadcast messages, “hello” link maintaince messages and many other communications share a common pattern: a message is sent and one or more nodes should “vote” on the transmitted message. For reliable broadcasts, the vote is an acknowledgement that “I have received and can decode the message”. If a node has not received the message, the sender would retransmit it. Link maintanance messages are almost identical, except that if a formerly “adjacent” node doesn’t receive the message, it is removed from the node neighbors table (with no retransmission). Many other protocols, such as voting protocols, can map to a similar query followed by a yes/no decision from other nodes.

Some of these protocols concerning a single network “link” have an analgous extension to a “network” counterpart. For example, there is considerable work on providing reliable network-wide support for broadcast packets in wireless networks, as well as distributed leader election.

2.1 SMACK - Reliable Link Layer Broadcasts

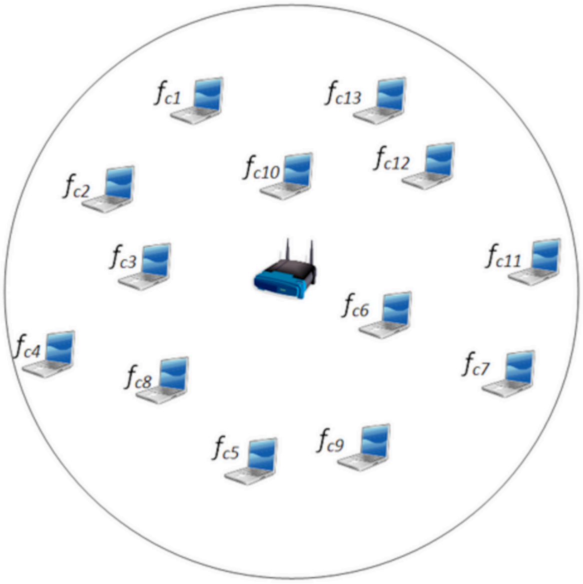


Figure 1: ACKs using Multicarrier Modulation

For any reliable broadcast mechanism to be reliable, there must be a clearly defined set of nodes in the network; Figure 1 shows a single access point and multiple clients. Each client is assigned a unique “membership number”.

For this implementation we have chosen the OFDM based physical layer for 802.11a/g as the underlying signaling. Figure 2 shows a schematic illustration of the properties of the OFDM waveform that are needed. A given bandwidth, such as the 2.4Ghz band used by 802.11g, is subdivided in a number of *subcarriers* around a center frequency; that center frequency is the "channel" to which an 802.11 radio is set. Some subcarriers are lost to guard bands (to prevent interference with adjacent channels) and some are lost for other puposes such as pilot tones, used to improve reception.

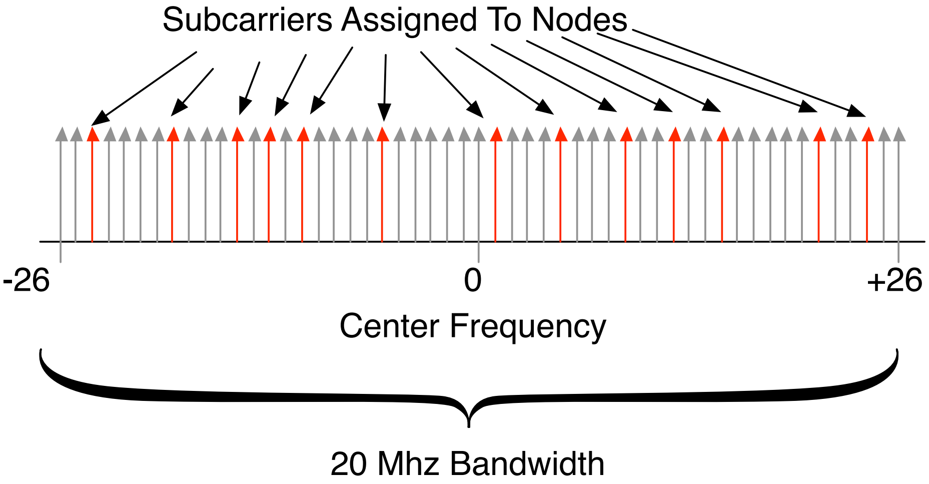


Figure 2: Schematic Illustration of OFDM Waveform properties

In 802.11g, 53 subcarriers remain for data modulation. Normally, a single transmitter modulates all subcarriers to send high bandwidth data. Since we only need to transmit a "yes" or "no", we assign subcarriers to individual nodes, as illustrated in Figure 2; different clients are assigned subcarrier bins labeled as $f_{c1}, f_{c2}, \ldots, f_{cn}$ where n depends on the number of users and the number of subcarriers available. The orthogonality of individual subcarriers allows us to use each of them as separate data carriers for different hosts. Using multicarrier modulation techniques allows the AP the receive ACKs from a greater number of clients in the shortest possible time, dramatically reducing the time to gather reliable acknowledgements for broadcasts. We use the physical layer to combine the responses from the different nodes. Upon receiving a successful broadcast message from the AP the clients use their pre-defined subcarriers to encode a $'1'$ for an ACK and $'0'$ for a negative ACK, using BPSK modulation.

To summarize, the protocol has the following steps:

1. When nodes join the network, the AP assigns each node a unique "membership id", which is a small integer.

2. An AP sends the broadcast message using conventional PHY specifications for 802.11a/g.

3. On receiving the broadcast message all clients decode the message (if possible).

4. If a client successfully decodes the message, the client then uses the single orthogonal subcarrier specified by the membership identifier to indicate it has received and decoded the message

5. The AP receives the composite time domain signal of *all* OFDM subcarriers and performs an FFT to obtain the frequency domain representation of the signal. After performing demodulation the individual acknowledgements can be recovered. A one in the n^{th} bit position can be mapped as an ACK from one of the N (number of subcarriers) clients.

Due to the conversion between the time domain and frequency domain, relatively tight timing synchronization is needed for the composite additive signal to be decoded at the AP; however, that time synchronization is provided by the broadcast message itself.

If the total number of user is less than 53 then there will be unused frequency bins as shown by the gray arrows in Figure 2. If however the number of users is more than the available frequency bins then more clients can be accommodated by assigning the same frequency to different user over multiple OFDM symbols which makes the protocol highly to an arbitrary number of clients. We'll also show that despite having the acknowledgements take twice as long for 54 *vs.* 53 clients, the time spent transmitting the acknowledgements is minimal compared to the actual broadcast message.

There are some important challenges in implementing such a protocol,

- The near-far effect - the duration of the ACK transmission by any client will depend on the longest propagation delay in the topology.
- Timing Requirements - the AP must have the correct samples to be able to correctly decode the time domain signal to extract the ACKs.
- Subcarrier assignment - we require *a priori* assignment of subcarriers to individual clients is required.
- Orthogonality - the individual subcarriers must not interfere be able to be decoding at the AP.

2.2 Extending Link Layer Broadcasts

As Figure 1 makes clear, we have mainly worried about providing a reliable broadcast for a "single hop" wireless network. We'd still like to have reliable broadcasts in multi-hop wireless networks. Such protocols usually use link layer flooding which often requires re-broadcast and leads to a common phenomenon called a *broadcast storm* [4]. This problem is especially elevated by that lack of ACKs – without an acknowledgement, it's unclear what nodes have received messages. ACK-based broadcast schemes degenerate the broadcast mechanism into multiple unicast communication increases network overhead and latency. Tang and Gerla [7] propose an ACK based broadcast scheme by changing the 802.11 MAC layer. Sheu et. al in [6] provided an algorithm for exchanging ACKs for broadcast messages by allotting pre-defined time slots for the hosts to transmit the ACKs back to the AP. In both the cases the fundamental assumption is that the ACKs are transmitted one after the other.

Given a reliable ACK as a basic operation, we can obviously improve on these algorithms. More importantly, we can use the *time of arrival* information available at the physical layer to further improve the performacne of reliable network broadcasts. We show how to do this in §4.

3 Hardware Implementation

To demonstrate that the challenges to using simultanous reception to implement reliable broadcasts that are tractable, we implemented a prototype using a software defined radio platform. The basic design involves an OFDM transceiver on a Virtex-IV FPGA along with a custom front-end radio. Design and implementation has been detailed in [1] , which deals with all the signal processing algorithms that has been synthesized into fixed point hardware designs.

The platform is capable of transmitting and receiving generic 802.11g as given in physical layer specification [3].

The OFDM transceiver components are shown in Figure.3 and consist of the following of a custom radio front-end responsible for up/down conversion to/from the 2.4GHz ISM band. and a Xilinx ExtremeDSP development kit IV manufactured by Nallatech. The ExtremeDSP board includes a Virtex IV equipped with a PCI/USB interface and two sets of A/D and D/A converters. Gain control is also a part of the radio which can be controlled by software on the host computer

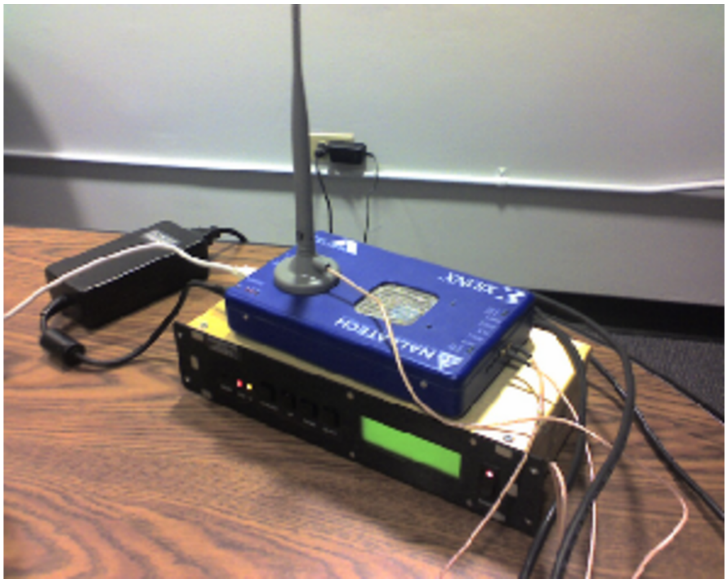


Figure 3: Hardware Platform

Implementing the protocol described in section2.1 it requires transmission of *non-contiguous OFDM symbols*, where none but one of the subcarriers is used to transmit the information. This requires some changes in the transmitter design. The transmitter design has been detailed in [2] which employs a hybrid design allowing sufficient reconfigurability to perform such non-contiguous transmissions. The protocol requires the involvement of a reconfigurable transceiver as well as a MAC layer which controls the hardware to perform the required tasks.

Figure 4 shows the basic schematic diagram of the protocol components. The receiver subsystem is responsible for broadcast packet detection, synchronization, equalization and decoding of the information bits. The FPGA is connected to the host over a PCI bus which is used to transfer the information bits, if the packet passes the PLCP parity check, to the MAC entity. If the packet was successfully decoded the MAC entity sends the modulation and subcarrier information of the same PCI interface to the hardware which interpreted the message and modulates one subcarrier, which is pre-assigned by the AP, to transmit the acknowledgment for the broadcast packet. The acknowledgment is essentially a single tone whose frequency depends upon the subcarrier index used for modulation.

The "AP" (implemented using our hardware) receives a composite additive signal from all the clients and, depending upon the number of users, the number of distinct frequency components in the signal will vary. A simple Fourier transform at the AP will reveal the tones in the signal and since the frequency indices are pre-assigned the AP can decipher how many clients have sent an acknowledgment for the broadcast packet. Observing the magnitude of the Fourier transform we can identify high energy subcarriers and thus identify the corresponding client.

However, the transmission time for different clients depends on their distance from the AP. Therefore it is important for the AP to wait for a specific time T, dependent on propogation delay, to insure all clients have decoded the broadcast message and the ACKs have propagated to the AP. In a typical infrastructure network the distance between the AP and clients can be considered to be at most 300m, which introduces a propagation delay of approximately 1μs. The transceiver turnaround time and PCI transfer time also must be accomodated – all clients must be transmitting at the state time. Therefore we can define T as

$$T \geq 2 \times T_{propagation} + T_{r_x latency} + T_{PCI\_transfer} + T_{t_x latency}.$$

Figure 5 shows a simulated environment (using Matlab) with 16 clients; four of the clients acknowledgements to a broadcast are shown. Since each of the clients uses one subcarrier to send ACKs, the ACKs are nothing but a single tone with frequency depending on the subcarrier index.

The bottom sub-figure shows the composite waveform, which is the summation of the tones from all 16 clients. The blue dotted line represents the FFT window used in OFDM demodulation. It is assumed that the propagation delay for uplink and downlink combined is less than 1 OFDM symbol = 4μs; this is well within our assumption of a 300m transmission range and allows for a 2 μs processing overhead. The receiver and transmitter latencies and the PCI transfer time will remain constant for all clients because the payload is the same for all nodes.

Figure 6 shows the frequency components of the composite waveform after performing the FFT; this is a snapshot of the response at a single point in time. In this case 8 clients have sent an ACK – only those subcarriers which are used to transmit the ACKs show a higher magnitude. The waterfall plot in Figure 7, which shows the spectral information over a longer time, shows the time lags between different ACK transmission and how the AP should adjust its FFT window to compensate for the near far effect. Not all the responses in Figure 6 have the same magnitude and some threshold must be set. That threshold will depend on the relative magnitude of the subcarriers used for transmission and the communication channel condition. This will also depend on the implementation and is a design parameter.

Other channel effects can play a significant role in this scenario. During downlink, the AP sends the broadcast packet with proper pilot tones inserted at regular intervals [3]. These are used by the client nodes to estimate the channel and equalize [1] the signal.

The uplink posses a greater challenge as there are no pilot tones to estimate the channel. At the same time channel estimation and equalization is a redundant step in this case as a single subcarrier is excited with BPSK modulate symbol. Thus unless the nodes are moving

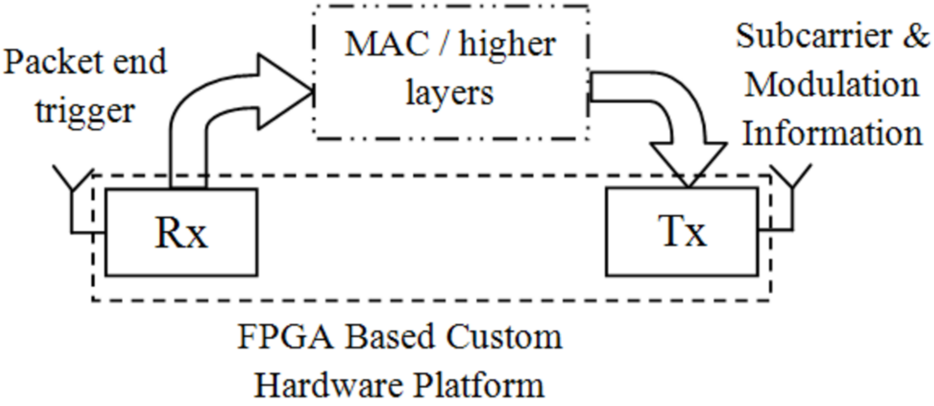

Packet end trigger
MAC / higher layers
Subcarrier & Modulation Information
Rx
Tx
FPGA Based Custom Hardware Platform

Figure 4: Hardware Platform Schematic

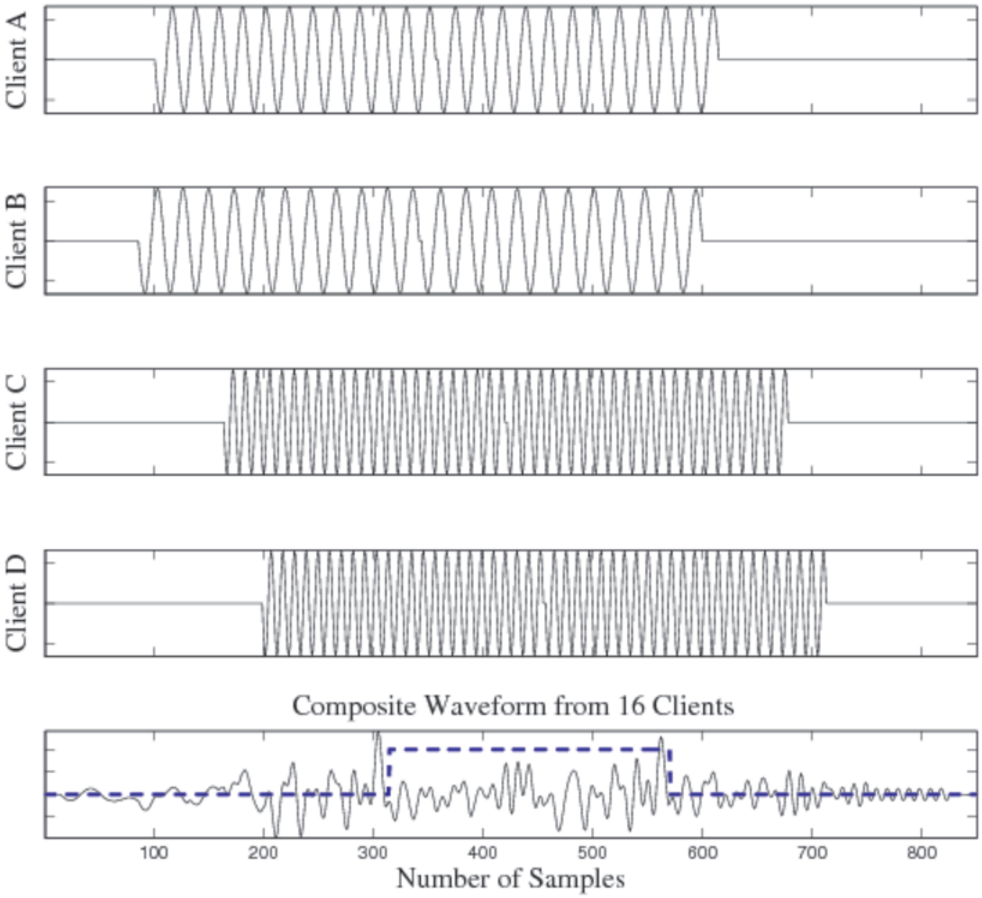


Figure 5: Timing Offsets Between ACKs and FFT Window

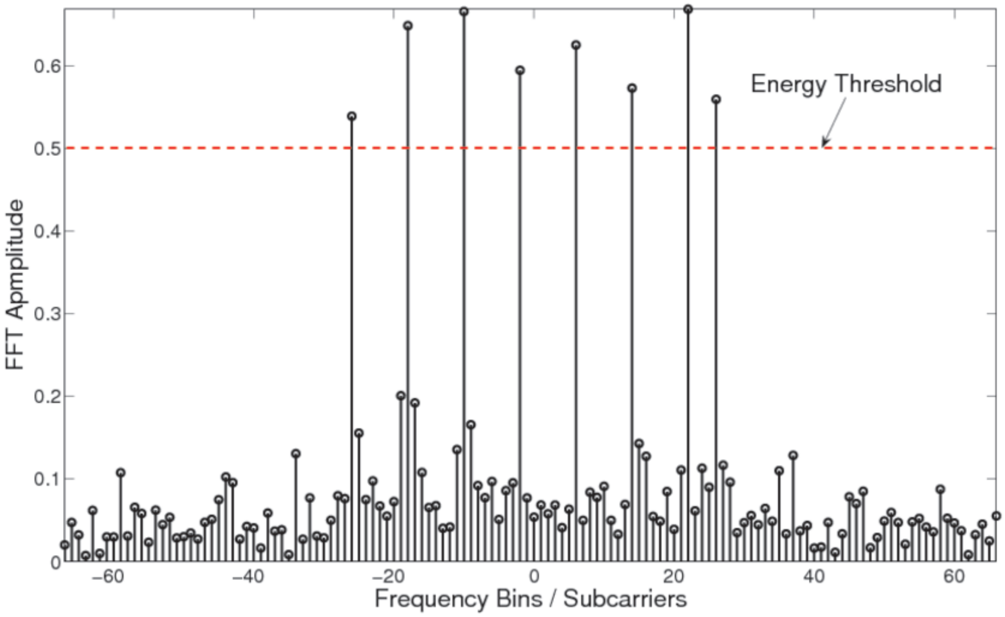


Figure 6: Fourier Transform of the Composite Waveform

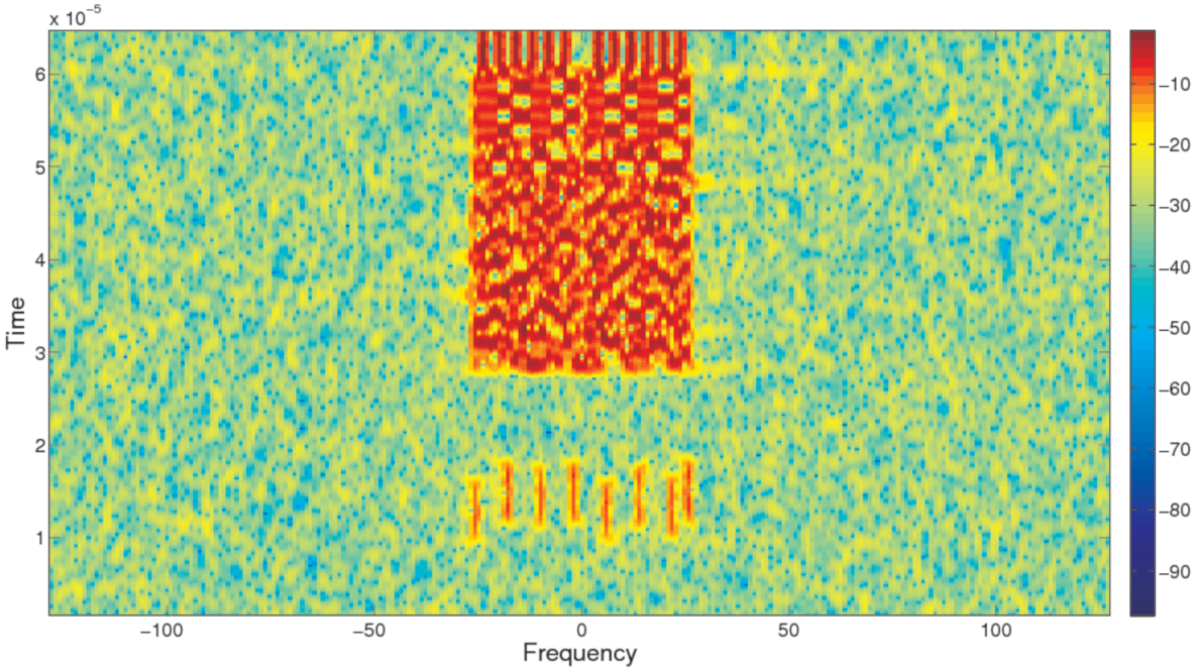


Figure 7: Waterfall Plot of the SMACK Protocol

at such speeds so as to introduce Doppler shifts there is a very little probability that the tones will shift frequency bins (which would lead to false positives).

Normally, the 802.11g standard mandates that a low data rate is used to transmit a broadcast packet, in order to increase the probability that all nodes receive the data. Reliable ACK's allow higher transmission rates to be used. Some of the nodes may not detect those packets, but the message can then be retransmitted at a lower modulation, just as is currently done with unicast messages (which use ACKs).

3.1 Results From Hardware

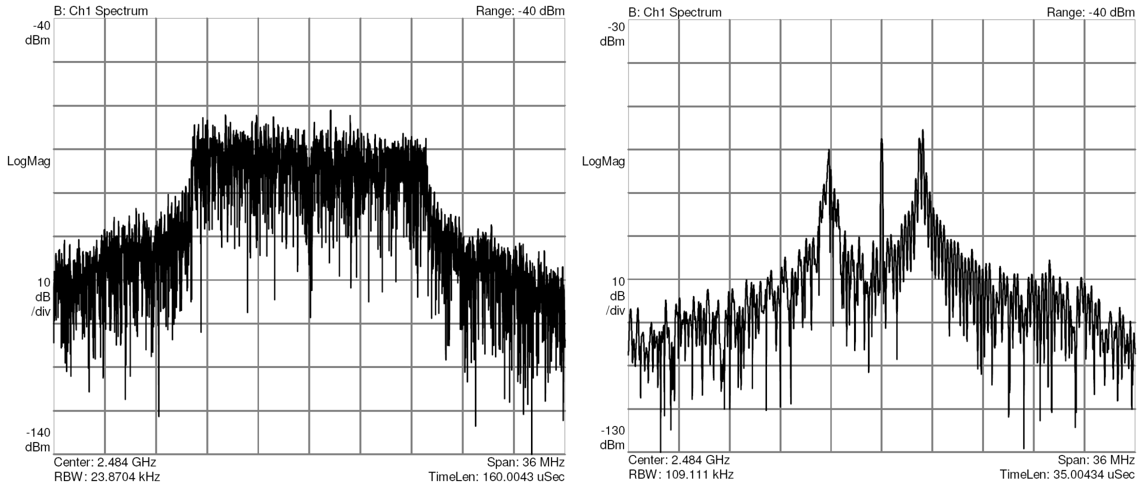
Given that you can demonstrate almost anything using a Matlab simulation, we felt it was important to demonstrate the protocol using three prototype hardware nodes. One of the radios was used to transmit one broadcast packet using the standard 802.11a/g PHY specification. The receivers decoded the broadcast packet and prepare the ACK packet with information on their pre-assigned subcarrier and transmit. The receivers were placed at two widely-varying distances from the transmitter to highlight the impact of near-far differences in clients. We used a vector-signal analyzer to capture the physical data.

Figure 8(b) shows the spectrum for the broadcast packet, which uses all the subcarriers for transmitting broadcast information, and Figure 8(a) is the spectrum for the composite ACK signal from the two client nodes. Node 1 is transmitting in subcarrier $+8$ and node 2 is transmitting the ACK using -8. Figure 9 shows that the node transmitting in subcarrier $+8$ has a higher signal power (closer to the AP) compared to the one transmitting using subcarrier -8 (further from AP). Also different time of arrival of the ACKs show the near-far effect. The waterfall plot also shows the broadcast packet at the top of the graph – that packet is transmitted first using the full spectrum available.

3.2 Efficiency and Generality

To understand how much more efficient it is to use physical signaling as we've done, consider the costs of transmitting a message using the 802.11g PHY that's the basis for our extension. A normal message requires a 20μs premable to be transmitted and then, at best, each 48×6 bits takes one OFDM symbol time (4μs) to transmit. Thus, a 64 byte message, which can't actually even contain the ethernet addresses in a standard 802.11g packet would take at least $20 + 4 \times 3$ or 32μseconds. After a 2μsecond "SIF" period, clients would normally respond using a similar message format. Thus, an ACK to a standard 802.11g packet would take another $\approx 32\mu$s. By comparison, using SMACK *all* clients can provide acknowledgement information within two OFDM symbol periods, or a total of 8μs.

This means there's plenty of time for higher level signaling and coordination. For example, rather than simply "broadcasting" some data, the AP could be asking a higher level question such as "do any of you have a router to node XYZ?". The client computers could then take varying amounts of time to analyze that query and then respond. The only constraint is that there must be some previously agreeing upon time after which all nodes should have signaled their answer. Since a normal "dumb ACK" take $\approx 32\mu$s, it may make sense to allow 10's of microseconds for the operating system to analyze the query and respond. Some nodes might signal earlier and some might signal later, but in the end, the information from all nodes could be combined and demodulated at the receiver.



(a) Spectrum of Broadcast

(b) Spectrum of ACKs

Figure 8: Spectrum of the Broadcast Packet and ACKs using three prototype radios

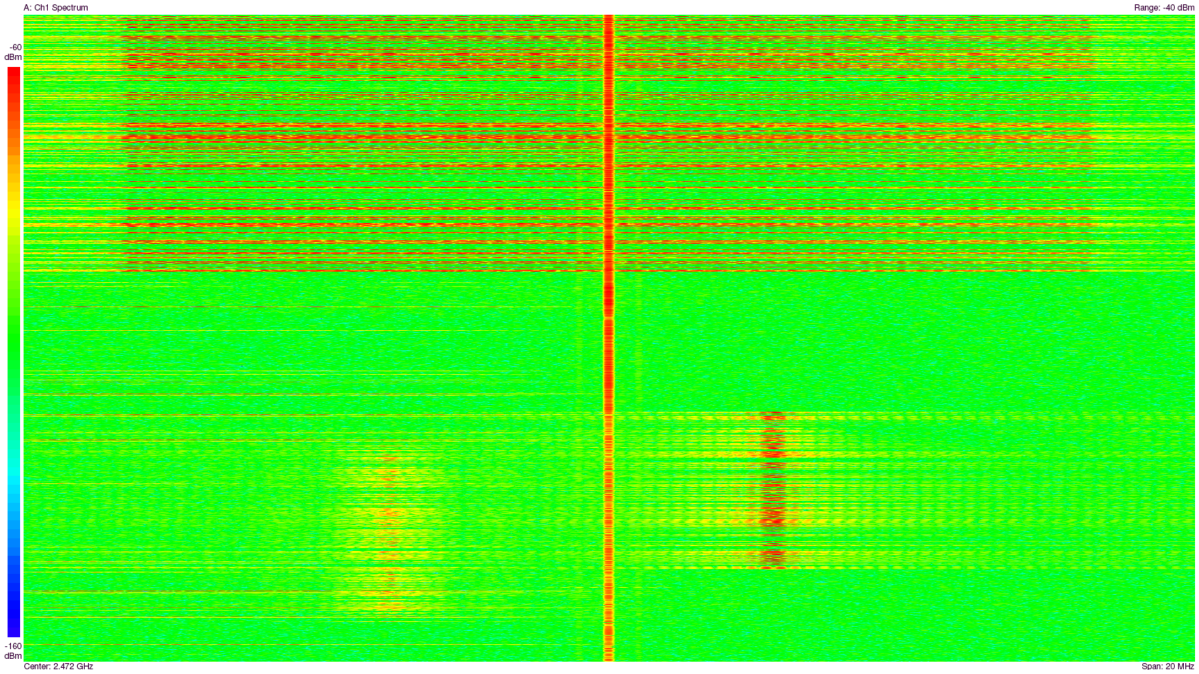


Figure 9: Waterfall Plot for SMACK Protocol Using Two Prototype Radio Platforms

4 Multihop Reliable Broadcasts

We can also use the physical layer information estimate which nodes are further away and then utilize this information to extend the smart acknowledgement protocol to choose a remote neighbor for the next broadcast in a network-wide broadcast. If you examine Figure 9, you notice that the signal from the "near" node (on the right) arrives before that of the "far" node (on the left). This can be detected at the physical layer using multiple FFT's and correlators[1], meaning that for the set of nodes that respond, we can determine which ones are further away (assuming they take the same amount of overhead time to start the ACK transmission). We can exploit that information to build a reliable broadcast protocol in multihop mesh networks with a minimum number of broadcast packets that mitigates broadcast storm.

Assume the origin node of the broadcast message transmits the packet and receives acknowledgement from its neighbors. Among the neighbors who acknowledged successful reception of the broadcast packet, the source selects one of the "most distant" nodes as the source of next broadcast. That "most distant" node is informed that it should relay the packet, either by another standard 802.11g format message or by an extension to our physical layer signaling. For example, the origin node could then transmits a 2-symbol duration tone in the same subcarrier as the remote node to indicate that the specified node should then relay the transmision. All neighbors in the vicinity receives this signal, and can determine if they are to relay the broadcast.

In the next step, the selected remote neighbor transmits only if it has not transmitted the same broadcast packet before, or if it has not already received multiple copies of the same broadcast packet. However, since a node only selects *one* of its neighbors as the most-distant next transmitter, the message may only propagate in one direction. This can be overcome by adding standard limited-degree flodding using sequence numbers and standard CSMA/CA contention for channel access. In this method, a node cannot overhear its neighbors' transmission of acknowledgement while it itself is transmitting the acknowledgement, and cannot infer any idea whether all its neighbors has received the broadcast packet. This may lead to some redundant transmissions at the edge of the network, but the use of both broadcast acknowledgements and the selection of distant nodes for limited forwarding can greatly reduce the number of messages sent.

Figure 10 shows a network of randomly placed nodes in an area of $500m \times 500m$. Each node has a transmission range of $200m$. Nodes shown in red transmitted the broadcast message, while those shown in green received the broadcast message. Node 51 is the initiator of the broadcast message. It chooses node 56 as the next transmitter. Node 36 decides to transmit since it has not received a second copy of the broadcast packet, and thus the message propagates to the north of the area. Also, node 62 decides to transmit due to similar reason, but neighbors of node 62 is a subset of neighbors of the initiator 51, and so this broadcast is redundant.

We implemented an analytical model of this protocol and figures 11 show performance of the protocol in multihop network. We choose a random scenario of nodes with a random initiator of the broadcast message in an area of $1000m \times 1000m$. The transmission range of the nodes has been considered to be $200m$ throughout the analysis. In Figure 11, we notice that total number of broadcasts in the network does not increase linearly as the number of nodes increases in the network. Even with 500 nodes in the network, the total number of broadcast packets approximately equals to only 35.

[1] We omit details for space.

The limited paper length precludes describing full details of this approach, but the the brief description above indicates that we can use physical layer information to improve higher level abstactions, such as reliable network wide broadcast.

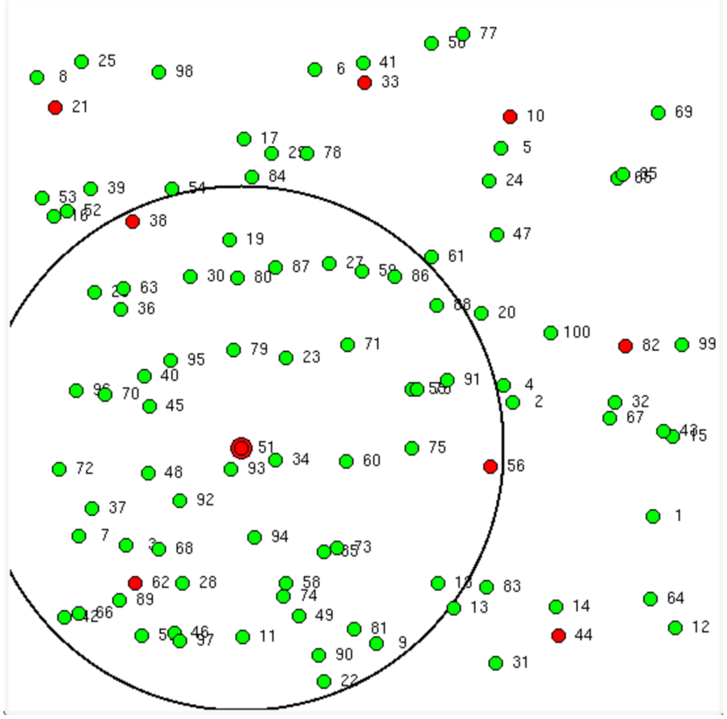


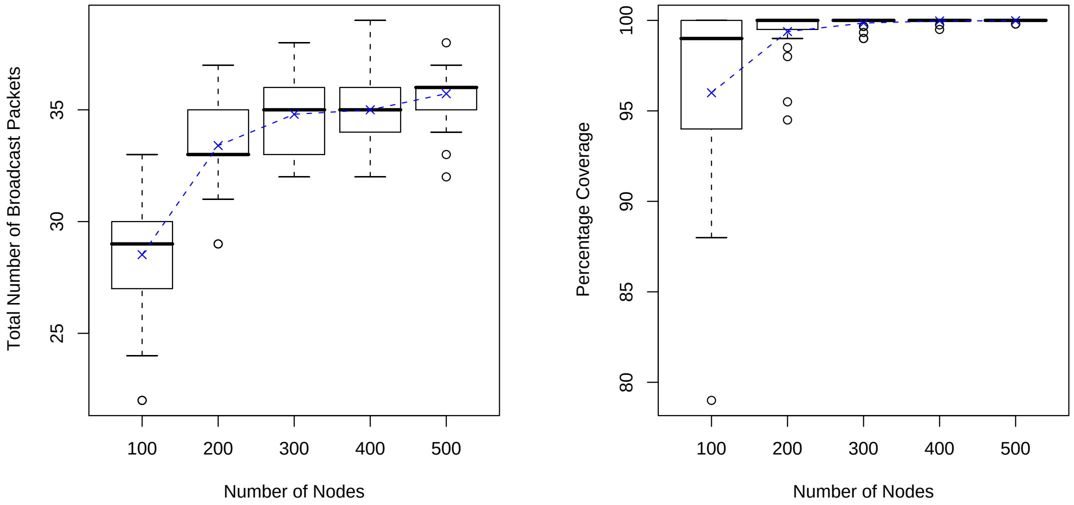
Figure 10: Percolation of Broadcast Message in the network

5 Conclusion

We've shown that by using, rather than fighting against, the properties of the wireless physical media, we can develop robust signaling primitives that are both practical and allow innovative algorithms. We used a signalling method (OFDM) that is easy to understand and visualize, but the general technique is amenable to other methods of orthogonal signaling, such as CDMA or combined methods such as coded OFDM.

The critical insight is that we can combine the results from multiple clients using simultanous reception in an efficient manner. We can use this mechanism to both make specific network functions, such as broadcasts, reliable, but can also use the primitives to implement higher level group communication and signaling protocols. As long as the queries require simple "yes/no" answers, there are a number of robust mechanisms to combine the signals.

The question remains of how such functionality could be exposed to client and host operating systems, particularly since similar techniques are difficult to implement on non-broadcast networks (*i.e.* most traditional networks).



(a) Number of Broadcast Packets Transmitted

(b) Percentage of Coverage

Figure 11: Performance with increasing Nodes

References

[1] A. Dutta. Reconfigurable OFDM Receiver for Next Generation Wireless Mesh. Master's thesis, University of Colorado, Boulder, Colorado, United States, 2008.

[2] J. Fifield, P. Kasemir, D. Grunwald, and D. Sicker. Experiences with a platform for frequency agile techniques. In *DYSPAN*, 2007.

[3] IEEE Computer Society : LAN/MAN Standards Committee. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*

[4] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.

[5] C. E. Perkins and E. M. Royer. Adhoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb. 1999.

[6] S.-T. Sheu, Y. Tsai, and J. Chen. A highly reliable broadcast scheme for ieee 802.11 multi-hop ad hoc networks. volume 1, pages 610–615, 2002.

[7] K. Tang and M. Gerla. Mac reliable broadcast in ad hoc networks. volume 2, pages 1008–1013 vol.2, 2001.