A REAL TIME 4X4 MIMO-OFDM SDR FOR WIRELESS NETWORKING RESEARCH

Jesse Chen,Weijun Zhu‡, Babak Daneshrad*, Jatin Bhatia‡, Hun-Seok Kim*, Karim Mohammed*, Sandeep Sasi‡, Anish Shah**

*Wireless Integrated Systems Research (WISR) Lab, Electrical Engineering Department,
University of California, Los Angeles, 90095, USA
‡Silvus Communication Systems, Inc. Los Angeles, CA, 90064

Abstract

A real time, 2 Mbps to 150 Mbps portable SDR unit with MIMO and sensing capability which exposes all the PHY parameters to the higher layers will help advance experimental cognitive radio (CR), and wireless networking research. The current SDR implements a slight variant of the 802.11n draft specification, with the entire baseband implemented on a single Xilinx Virtex II 8000 device using commercial boards from Nallatech. A fully self contained PHY solution, this SDR is capable of over 300 unique modes of operation all under direct control of the MAC, and features a robust MAC API through which all PHY parameters can be controlled on a per-packet basis. The same API will allow the PHY to communicate channel state information, SNR, and RSSI measurements back to the MAC. In this paper we provide an overview of the SDR and its development.

1. INTRODUCTION

1.1 Motivation

Development of wireless networks include many phases, but invariably, verification on a practical testbed or prototype is needed to validate the theoretical and simulation work. Additionally, experimental validation of MAC performance or the impact of cognitive approaches on the network throughput requires versatile broadband radios with minimal packet decode latencies. To date, network and cognitive radio researchers have been forced to adopt commercial platforms such as 802.11 based systems for their research needs. These platforms seldom provide full control of the RF, PHY, and lower MAC layer functionalities. Both research and commercial grade testbeds have been reported to address these shortcomings. Lyrtech [1] for example, offers a commercial grade platform for developers which include only RF and baseband hardware. The work at Rice University [2] is similar in nature, but goes one step further by making a repository of firmware modules available to the user. The testbed described in [3] is of a different class, in that the signal processing is performed offline using Matlab. Unlike the work reported in [1,2], the testbed described in this paper emphasizes real-time capability and firmware development, rather than custom hardware development. In this sense, it is also fundamentally different from the class of store-and-process testbeds seen in the literature. To the best of our knowledge, the testbed presented in this work is the most comprehensive and versatile reported to date. It is an "out-of-the-box" functional solution for networking researchers, and has a robust and easy to use MAC-PHY API, featuring more than 300 unique modes of operation.

1.2 SDR Highlights and Features

Our efforts produced a real time MIMO OFDM testbed that can satisfy the needs for both the cognitive radio community and the traditional wireless networking community. The testbed was implemented on a single Virtex II FPGA with real time capabilities. It was developed to support a large number of permutations of physical layer modes (see Table 1) and a slight variant of the IEEE 802.11n draft proposal [4]. The featured data rates range from 2 Mbps to 150 Mbps in an over-the-air bandwidth of 5 MHz, 10 MHz and 15 MHz. This allows for development of prototypes for intelligent spectral allocation, high throughput testbeds, and other testbeds whose main concern lies beyond raw throughput. Figure 1 shows the current form factor for the testbed described in this work.

All combinations of the parameters shown in Table 1 are supported. These modes are defined by the MAC through an API interface. The header of each MAC to PHY transmission contains the value of the configuration registers for that specific packet. In this manner, the higher layers can dictate exactly the type of packet and mode that is to be used. In the reverse direction, the PHY can provide CRC results, SNR, RSSI, and channel state information to the MAC and networking layers to enable advanced protocols.

The next version of the prototype will allow independent allocation of the RF transceiver chains. This will enable any number of antennas to be used for transmission while the others are engaged in tasks such as channel sniffing for cognitive radio applications.

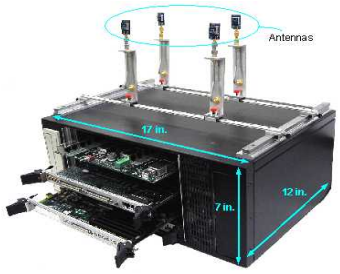

Figure 1 – The first generation testbed (both baseband and RF) is built into a cPCI chassis measuring 17"x 8"x 12"

Table 1 – Supported Modes

Bandwidth	5MHz, 10MHz, 20MHz
Antenna Configuration	Any combination up to 4x4
Modulation	OFDM with 2,4,16,64 QAM constellations
Coding	Binary Convolutional Code
Coding Rate	1/2, 2/3, 3/4, 5/6
Packet Size	0 ~ 65535 bytes
MIMO Processing	MMSE
MIMO Signalling	Spatial multiplexing, space-time block coding, cyclic delay diversity, RX beamforming

2. PACKET STRUCTURE

The frame structure of the current testbed is based on the IEEE 802.11n next generation Wireless LAN standard [4]. Figure 2 illustrates the packet structure of the transmit signal. The duration of each section assumes a bandwidth of 20MHz and is proportionally scaled for lower bandwidths.

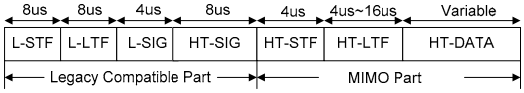

Figure 2 – Physical Layer Frame Structure

The packet is divided into two major sections, the first of which is termed the Legacy Compatible Part (LCP). Only one antenna is required to transmit the LCP and it can be decoded by a system with one or more receive antennas. When available, multiple antennas could also be used to transmit the LCP in order to enhance performance. The design of the LCP ensures backwards compatibility with legacy single antenna systems and allows for a network of nodes, each possessing a different number of antennas. The second section of the packet is MIMO specific. It supports per sub-carrier based spatial multiplexing and space-time block coding. Up to 4 spatial streams are supported. Refer to [4] for a detailed description of each of the subsections.

3. RADIO ARCHITECHTURE

3.1 PHY Architecture

Figure 3 and Figure 4 are the main functional block diagrams of the transmitter and receiver, respectively.

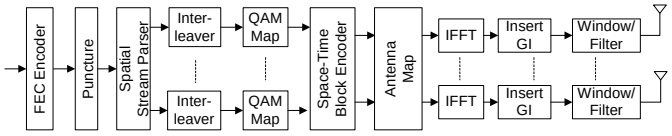

Figure 3 – Functional Block Diagram of the Transmitter

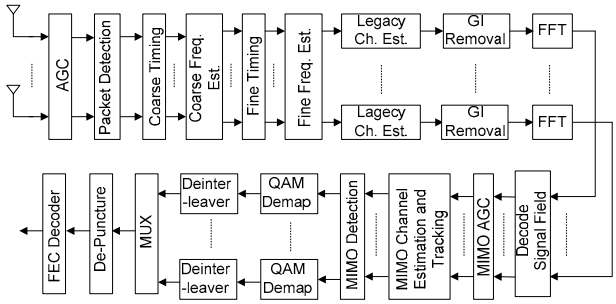

Figure 4 – Functional Block Diagram of the Receiver

3.2 PHY Algorithms

3.2.1 Synchronization

A two-stage synchronization strategy has been adopted for this work. The first stage is similar to [5] and uses the L-STF to detect an incoming packet and estimate the coarse packet start position as well as coarse frequency offset. The second stage uses the L-LTF to refine the packet start position and frequency offset estimate. Since the L-STF is periodic, packet detection is declared when the autocorrelation value rises above a certain threshold. This autocorrelation value will remain high until the signal transitions to the L-LTF. The estimated position when the autocorrelation falls below a threshold, is used as a reference for coarse packet start position. A coarse frequency offset can also be calculated using the autocorrelation value at the same time. Since the autocorrelation tends to have slow ramp-down, we correlate the received signal with a locally stored, truncated version of the L-LTF to refine the OFDM symbol timing during the second stage. Once the symbol timing is determined, the periodic property of the L-LTF enables refining of the frequency offset estimate.

3.2.2 Channel Estimation

A basic per sub-carrier channel estimation method has been implemented for simplicity. This approach is outlined in [6] and simply correlates a local copy of the training sequence with received samples. Channel estimates are obtained by averaging the entire received HT-LTF, weighted by the transmitted HT-LTF values on a per sub-carrier basis.

3.2.3 Phase Noise and Residual Frequency Tracking

A simple open-loop phase offset estimation and correction technique has been implemented. The phase offset for each OFDM symbol is estimated by comparing the equalized, received pilot sub-carrier values and expected pilot sub-carrier values (derived from the channel estimate and known transmit pilot values) [7].

3.3 MIMO Decoding Algorithm

Our literature search revealed three classes of MIMO detector solutions. These included ML based detectors [8, 9], V-BLAST based detectors [10-12], and linear MMSE based detectors [13, 14]. Based on the results of a literature search and our early work, we reached the conclusion that the linear MMSE based solutions were the best candidates for achieving the target throughput, latency and hardware resource usage requirements. All of the MMSE detector implementations reported in literature [13, 14] use a Squared MMSE formulation of the MIMO detector problem with an explicit matrix inversion of $(\mathbf{H}^*\mathbf{H}+N_o\cdot\mathbf{I})^{-1}$. In a fixed point hardware implementation however, a matrix inversion followed by matrix square operation is generally not desirable because of numerical stability issues [15]. In fact, the inverse matrix operation as well as the matrix squaring operation can be avoided in the linear MMSE detection problem by using the equivalent Square-Root reformulation [12]. The most computationally intensive part of the Square-Root MMSE algorithm is the QR decomposition on the compound channel matrix. We examined two types of low complexity QR decomposition techniques. The first is a modified Gram-Schmidt QR decomposition and the second is a Givens rotation based QR decomposition [15]. The unitary transformation based Givens rotation method is more numerically stable than the Gram-Schmidt method [15]. When one considers hardware implementation on an FPGA, multiplication-intensive methods such as the modified Gram-Schmidt QR decomposition are usually more desirable than a CORDIC-intensive Givens rotation QR algorithm which requires an excessive number of slices. A number of dedicated multipliers in an FPGA are available for use without incurring extra cost whereas a CORDIC operator consumes a significant amount of FPGA slices. Moreover, we have found that the modified Gram-Schmidt based QR in Square-Root MMSE detection can be made more numerically stable to a point where it is comparable to the unitary transform based QR technique [16]. The fast Givens rotation (also called Squared Givens rotation) was also considered, but it requires more FLOPS than the Gram-Schmidt based method when it is applied to a complex valued matrix [15]. Based on the aforementioned study, we selected the modified and scaled Gram-Schmidt QR decomposition combined with Square-Root MMSE MIMO detection for our hardware implementation.

3.4 MAC Signalling/Messaging Interface

The MAC signalling/messaging interface has been designed for flexibility and expandability. The PHY layer interacts with the MAC layer through a set of commands that allow for fast information transfer of TX and RX frames. These commands also allow the MAC to completely specify all of the parameters necessary for transmission, as well as to request a large subset of the internal signals from the receiver PHY. The MAC can specify the following transmit parameters on a per packet basis:

1. **Antenna Selection** (dictates the specific antennas to be used)
2. **Constellation Size** (BPSK,QPSK,16-QAM,64-QAM)
3. **Coding Method** (Convolutional or Low-Density Parity Check)
4. **Coding Rate** for convolutional code (1/2 , 2/3 , 3/4 , 5/6)
5. **Number of Antennas** (1 - 4)
6. **Antenna Processing** (Space-Time Block Coding, Spatial Multiplexing, Beamforming)
7. **Guard Interval Length** (0.8 μs/1.6μs in 10 MHz mode)
8. **Carrier Frequency/Channel Selection**
9. **Bandwidth** (5, 10, 15 MHz)
10. **Preamble Length** (11a, or HT Mixed/Green Mode)
11. **Power Level**
12. **Packet Length** (0-65535 bytes)
13. **Sounding Packet** (This indicates that the packet is to be used for channel sounding)
14. **MAC Payload Data Unit Aggregation** (Whether or not aggregation needs to be performed at the receiver)
15. **Smoothing** (Whether or not the PHY should perform smoothing of the channel estimates at the receiver)
16. **Scrambler initialization** (7-bit initialization sequence)

This interface is quite expandable and can allow for parameters to be added in the future. A full set of these parameters is passed to the PHY for each packet transmission. The MAC begins transmitting the data payload immediately after passing the configuration information to the PHY. This ensures that the PHY transmits the data payload as soon as possible.

On the receive side, when a valid packet is decoded, the PHY initiates a transfer to the MAC with items 2-15 of the above list. It can also provide the following additional information:

- **Received Signal Strength Information**
- **Signal to Noise Ratio**
- **Silence Level**
- **Channel Estimates**
- **QAM Constellation points**

Parameters 4-7 can be further specified for a subset of the subcarriers, transmit channels and receive channels. All of this information can either be captured for every packet received, or for only the packets which correspond to specified Modulation Coding Schemes (MCS). This captured data can then be stored for later retrieval. There is also the option of capturing this information for any sounding packet that is received.

While the current implementation of the digital processing engine does not support parallel, independent communication sessions, the interface already has built-in support for this. The transmitter antenna selection parameter allows the MAC to define which antennas should be used to transmit each packet. Similarly for the receiver, the RF chains for each antenna can be configured independently, allowing the receiver to listen for packets on a variety of frequencies.

4. SIMULATION RESULTS

Two sets of simulation curves are presented here to describe the floating point performance of the system. Although fixed point simulations are not shown here, the performance of the actual hardware in a digital loopback configuration has been matched to these curves. All curves are based on the complete receiver implementation, and include tasks such as channel estimation, synchronization, PA non-linearity, and PLL phase noise. The IEEE 802.11n Channel Model D (non-line-of-sight) was used.

Figure 5 compares the performance of spatial diversity systems with the same data rate but different antenna configurations. The 4x4 system shows a 10dB improvement over the 1x1 at a 10^{-2} PER. Figure 6 compares the performance of spatial multiplexing systems with the same underlying constellation and coding rate. Clearly a MIMO spatial multiplexing system can achieve a data rate much higher than a SISO system with very minimal energy penalty (using energy per information bit as a metric for energy efficiency).

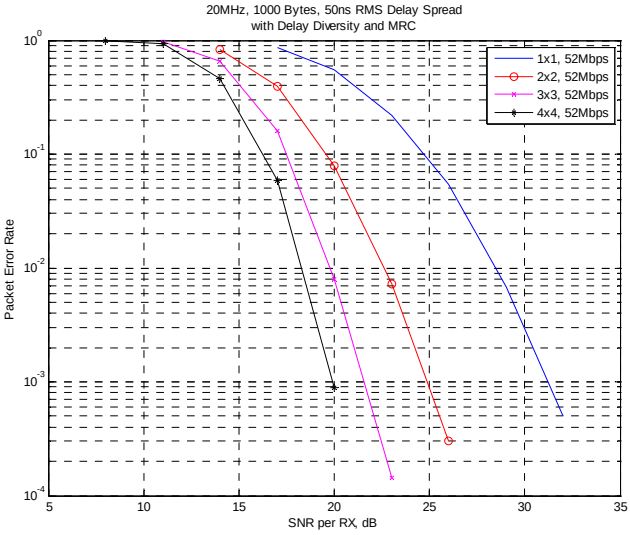

Figure 5 – Performance of Spatial Diversity Systems

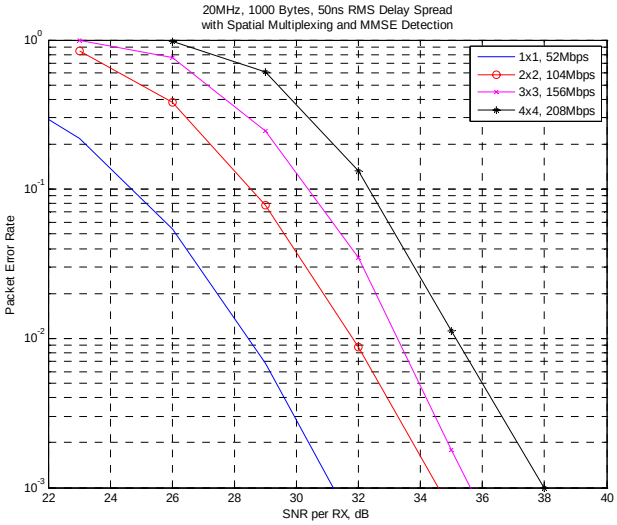

Figure 6 – Performance of Spatial Multiplexing Systems

5. HARDWARE

5.1 Top Level Architecture

The radio system presented in this paper consists of three main subsystems as depicted in Figure 7 : RF, Baseband and Host. The I and Q channels of the RF subsystem interface with the baseband processor through ADC/DAC modules which are mounted on a motherboard. The host computer uses the DimeTalk API to talk to

the motherboard through the PCI bus. RX and TX FIFO queues along with control block RAM have been implemented on the motherboard to facilitate efficient communication. Details of these subsystems will be given in the subsequent sections.

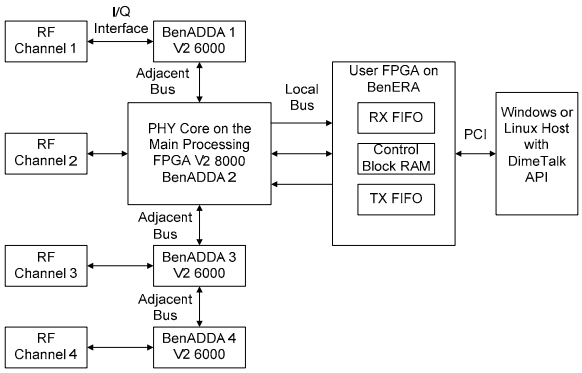

Figure 7 – Top Level Radio Architecture Diagram

5.2 RF Subsystem

The RF subsystem architecture consists of four dual-band capable transmit and receive chains. Each chain is built around the Maxim MAX2829 802.11a/b/g transceiver IC. Figure 8 shows a diagram of this architecture and clock distribution circuit driven by a <1ppm, 20MHz oven-controlled crystal oscillator (OCXO). Various tunable features such as center frequency, gain and linearity are controlled through the SPI interface on the MAX2829 which is driven by the parallel port of a single board computer (6U, cPCI form factor) housed within the same chassis. Fast changes in LNA and RX/TX VGA gains can be implemented through a series of parallel control bits. A switch network selects the desired band of operation as well as the desired RF mode (transmit/receive). Baluns, linear power amplifiers and band-pass filters complete the RF chains.

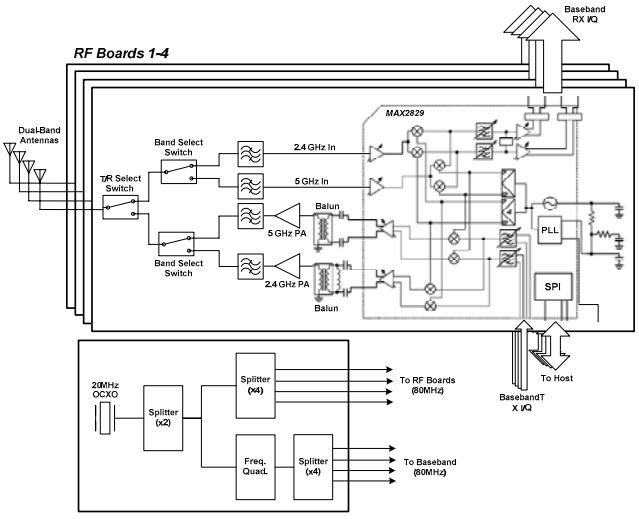

Figure 8 – RF Subsystem Architecture

Measurements of the RF subsystem's 1-dB compression point and third order intercept point (IP_3) were carried out for both the TX and RX chains. These results are presented in Table 2. A two-tone test was used to determine the IIP_3 and OIP_3 and was carried out with total TX and RX gains set at 18.5dB and 3dB, respectively. The two input tones were of equal amplitude and separated in frequency by 100 kHz. The 1-dB compression point input and output power was determined for the TX and RX chains using the same gains that were used for the IP_3 characterization.

Table 2 – RF Subsystem Characteristics

TX Chain (18.5dB total gain)	
Input 1-dB Compression Point	-3.0 dBm
Output 1-dB Compression Point	+14.5 dBm
Input IP_3 (100kHz input tone separation)	+8.9 dBm
Output IP_3 (100kHz input tone separation)	+23.9 dBm
RX Chain (3dB total gain)	
Input 1-dB Compression Point	+2.5 dBm
Output 1-dB Compression Point	+4.5 dBm
Input IP_3 (100kHz input tone separation)	+15.1 dBm
Output IP_3 (100kHz input tone separation)	+18.6 dBm

5.3 Baseband Subsystem

The baseband subsystem consists of four Nallatech BenADDA [17] modules, three of which feature a single Xilinx Virtex II 6000 FPGA. The fourth BenADDA features a Virtex II 8000 FPGA on which the entire baseband processor is implemented. The resource utilization for this FPGA is outlined in Table 3. All four BenADDA modules contain two 14-bit, high speed ADC (105MSPS) and DAC (160MSPS) channels which interface directly with the RF subsystem's RX output and TX input channels. A single Nallatech BenERA motherboard (6U cPCI form factor) carries the four BenADDA modules. The BenERA features a Xilinx Virtex E 2000 FPGA, a 64-bit (33MHz) PCI interface and a 240-way backplane GPIO bus. This motherboard handles the routing between the baseband FPGA (Virtex II 8000), the BenADDA modules and the PCI Bus. The same host computer which controls the RF modules through a parallel port interface also interfaces with the baseband processor through the BenERA on the PCI bus.

Table 3 – FPGA (Virtex II 8000) Resource Utilization

Resource	**Utilization**
Slices	46,590 (99%)
Slice Registers	57,980 (62%)
Lookup Tables (4-input)	64,270 (68%)
Block RAMs	152 (90%)
Multipliers (18x18)	127 (75%)

5.4 PHY-MAC Hardware Interface

The MAC interface can be divided into the following three groups of logical interfaces:

- **Parallel Data Interface**
- **Parallel Control/Status Interface**
- **Register Interface**

The Parallel Data Interface is primarily used to transfer payload data between the MAC and PHY. In addition, configuration and status data that change on a per packet basis are also piggybacked on to the payload data through this interface. The receiver and transmitter sections of the PHY have independent parallel data interfaces to the MAC, each comprising of a source-driven clock, a byte-wide data bus, associated data-validity and flow-control signals, as well as signals to classify the data present on the bus as payload, header, or trailer.

The Parallel Control/Status Interface is used for transfer of timing-critical control and status information. These include enable signals for each transmitter and receiver of the PHY, and CCA signals.

The Register Interface provides the MAC with access to PHY registers. In addition, several intermediate signals of importance such as channel estimates, QAM symbols, etc. are mapped onto register addresses on this interface. This enables the MAC to read out these signals in a continuous FIFO stream by initiating an

operation identical to a register read on the corresponding address. The register interface consists of a MAC-driven clock, a byte-wide command bus, a byte-wide data bus, and associated enable and flow-control signals. The flexible three-byte command word format chosen, allows for a seamless binding of register access with data stream probing, while providing a sufficiently large PHY register space of 32-bit words.

6. FIELD TRIALS

In both the indoor and outdoor trials, the ability to switch modes on a per packet basis was demonstrated by consecutively transmitting 13 different types of packets.

Indoor laboratory over-the-air trials were successfully conducted for 1x1, 1x2, 2x2 space-time coding (STC) and 2x2 spatial multiplexing (SM) modes. Each antenna configuration was exercised with a 5 MHz and 10 MHz bandwidth and QAM constellation sizes of 4, 16 and 64. The GUI shown in Figure 9 was built to facilitate the field trials and feedback valuable information such as error statistics, channel conditions and packet transmit/receive status, as well as received waveforms and constellations. Critical statistics such as channel eigenvalues, SNR and capacity, along with packet error rate (PER) and received constellations are displayed in real-time.

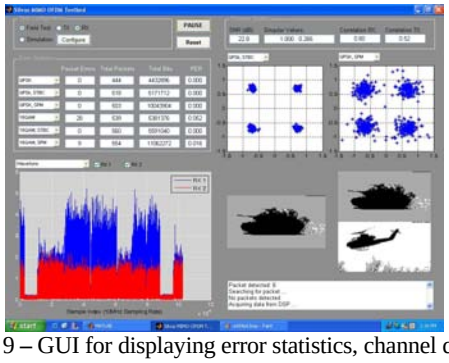

Figure 9 – GUI for displaying error statistics, channel quality, constellations and signal waveforms

A movie/file-transfer scenario with 2x2 ACK/NACK capabilities was also successfully tested. The transmitter was able to send multiple consecutive packets and wait for an ACK from the receiver. In the event that no ACK was received by the transmitter, retransmission occurred.

The 4x4 antenna configuration has been exercised in the form of an analog loopback and a 10 MHz bandwidth digital loopback (with the 4x4 FPGA bit file). The digital loopback trial was executed for 1x1, 2x2, 3x3 and 4x4 modes with QAM constellation sizes of 4, 16 and 64 and spectral efficiencies of 2-18 bps/Hz (corresponding to 20 Mbps to 180 Mbps peak over-the-air data rate). Indoor laboratory over-the-air trials were also successfully conducted for 3x3 STC and 3x3 SM modes with 10 MHz bandwidth and QAM constellation sizes of 4, 16 and 64.

Outdoor mobile trials were also conducted in a vehicle which travelled at low speeds through residential areas and high speeds on a freeway environment. Figure 10 illustrates a sample path taken by the vehicle. 10 Mbps operation was successfully validated in mobile trials at up to 70 mph. 1x2, 2x2 STC and 2x2 SM with QAM constellations of 4, 16 and 64 were all exercised.

7. CONCLUSION

A highly flexible, real-time 4x4 MIMO-OFDM SDR for wireless networking research has been presented which is capable of over 300 unique modes of operation. Bandwidth, antenna configuration, modulation coding scheme, packet size and MIMO signaling are all easily configurable via a robust, open MAC API. The baseband has been implemented on a single Xilinx Virtex II 8000 and enables a flexible data rate of 2-150 Mbps. Indoor and mobile trials reaching speeds in excess of 60mph were conducted to validate operation in all modes.

Figure 10 – Map of outdoor demo indicating path of mobile radio

8. REFERENCES

[1] Lyrtech Signal Processing, MIMO Applications Development. http://www.lyrtech .com/DSP-development/mimo_antennas

[2] P. Murphy et al. “Design of WARP: A Flexible Wireless Open-Access Research Platform”, *Proc. EUSIPCO*, 2006.

[3] S. Caban, et al. “Vienna MIMO Testbed.” *EURASIP Journal on Applied Signal Processing Volume*, 2006.

[4] IEEE P802.11n™/D1.06, November, 2006.

[5] T. M. Schmidl and D. C. Cox, “Robust Frequency and Timing Synchronization for OFDM,” *IEEE Transactions on Communications*, Vol. 45, No. 12, Dec. 1997.

[6] J. Terry and J. Heiskala, *OFDM Wireless LANs: A Theoretical and Practical Guide*. Indianapolis, IN: Sams, 2001.

[7] R. M. Rao, B. Daneshrad, “Analog Impairments in MIMO-OFDM Systems,” *Wireless Communications, IEEE Transactions on*, Vol.5, Iss.12, December 2006

[8] A. Burg, N. Felber, and W. Fichtner, “A 50 Mbps 4x4 maximum likelihood decoder for multiple-input multiple-output systems with QPSK modulation,” in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, vol. 1, 2003, pp. 332–335.

[9] A. Burg, et al. “VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm”, *IEEE Journal of Solid-State Circuits, Vol. 40, No. 7*. July 2005.

[10] P.W. Wolniansky, et. al.. “V-BLAST: an architecture for realizing very high data rates over the rich scattering wireless channel”, *International Symposium on Signals, Systems and Electronics*, 1998.

[11] B. Hassibi, “An Efficient Square-Root Algorithm for BLAST”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[12] R. Bohnke, et al., “Reduced Complexity MMSE Detection for BLAST Architectures”, *GLOBECOM* 2003

[13] M. Myllyla, et al. “Complexity Analysis of MMSE Detector Architectures for MIMO OFDM Systems”, Proceedings *of the Thirty-Ninth Asilomar Conference.*

[14] I. LaRoche, S. Roy, “An Efficient Regular Matrix Inversion Circuit Architecture for MIMO Processing”, *IEEE International Symposium on Circuits and Systems*, 2006.

[15] G. Golub, C. F. Van Loan, “Matrix Computations”, The Johns Hopkins University Press, 3rd Edition

[16] H.S. Kim, et. al. “A Hardware-Friendly Linear MMSE Detector for MIMO-OFDM Based Systems” (In Preparation)

[17] “BenADDA-Pro Reference Guide,” Nallatech Inc., July 19, 2005.

[18] Raghu Rao, et. al. “Indoor Field Measurements with a Configurable Multi Antenna Testbed,” *GLOBECOM* 2004