DESIGN OF THE TRANSIT ACCESS POINT HARDWARE PLATFORM

Patrick Murphy, J. Patrick Frantz and Behnaam Aazhang

Rice University
Department of Electrical and Computer Engineering
6100 Main St., Houston, TX 77005, USA
{murphpo, jpfrantz, aaz}@rice.edu

ABSTRACT

Our objective is to design, analyze, prototype and experimentally study the theoretical underpinnings for a wireless internet that simultaneously achieves deployability, scalability, high performance and a cost-effective economic model. A core building block will be what we call wireless Transit Access Points. A transit access point, or TAP, is a wireless base station with two major features. First, like any standard base station, it provides wireless data services to mobile users. Second, and more importantly, a TAP is capable of high speed wireless links to other TAPs. These connections utilize multiple antennas at each end to dramatically increase the spectral efficiency and throughput of the link. Such TAP-to-TAP links are designed to supplement, or even replace, the wired network infrastructure usually required when deploying wireless data systems. This paper presents the custom hardware platform designed for the TAPs project.

1. INTRODUCTION

The design of a TAP must support its use in a variety of situations in which it must provide a variety of services. This is clear in the network illustrated in Figure 1. In this example, every TAP provides connectivity to mobile users in its vicinity. Some TAPs rely on high-speed wireless links for their own connectivity (*B* & *C*, for example). Other TAPs have wired connections to a larger network infrastructure (*A* & *D*) but must share this connectivity with non-wired TAPs. Finally, some TAPs must act as wireless routers (*C*), providing network connectivity to other access points which cannot directly communicate with a wired node. The hardware design of a TAP must provide all of these capabilities.

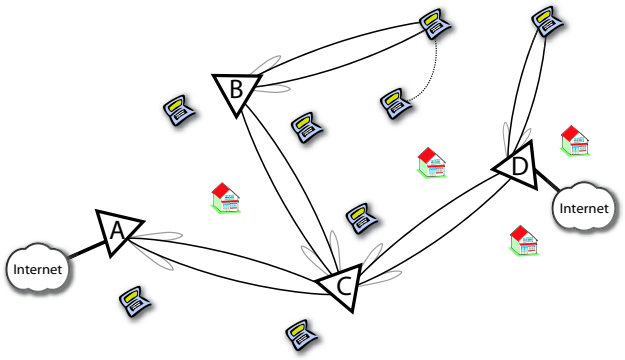

Figure 1: Example transit access point network

The need for platforms for prototyping wireless communications sytems is not new. A variety of other testbeds have been constructed in recent years for exploring multiple-antenna algorithms. Two of these testbeds are briefly described here to provide some background on our motivations for constructing our own.

One such testbed was designed at the University of Texas' Wireless Network and Communications group [1]. This testbed is built entirely from commercial off-the-shelf hardware, mostly from National Instruments. It provides resources for half-duplex operation at RF up to 2.7 GHz with two transmit and two receive antennas. Embedded PCs are used for baseband processing with algorithms implemented in LabVIEW. Processing and memory limitations prevent real-time operation; data bursts are limited to around 200ms in length with each burst requiring four seconds of processing at the receiver. This duty cycle permits fairly long transmissions (relative to packet sizes in wireless LAN standards), but clearly falls far short of real-time.

Another testbed was built at UCLA and has been used to implement and evaluate a wide variety of MIMO algorithms [2]. This testbed supports up to three transmit and four receive antennas. It operates in real-time at an RF of 220 MHz with a bandwidth of 4 kHz. The testbed hardware is a combination of custom RF hardware and commercial equipment for baseband processing. This testbed has been used to evaluate the performance of a large number of space-time algorithms in real wireless environments [3].

Each of these testbeds, and others like them, provide some of the resources we view as necessary for prototyping wideband MIMO algorithms. Neither one, nor any other testbed we could find, addresses all of the requirements we identified for highly flexible, real-time, wideband MIMO prototyping. This paper presents a hardware platform designed to address all of these needs.

Section 2 discusses in detail the various requirements and constraints involved in providing these capabilities. Section 3 describes the resulting hardware design. Finally, section 4 offers some concluding remarks.

2. HARDWARE REQUIREMENTS

2.1 Wireless Interfaces

One of the fundamental premises of the TAP design is its being equipped with multiple radios and antennas which can be used in unison for spectrally efficient links at very high data rates. This requires that the hardware design provide a means for multiple radios to be driven by a common baseband processor. Further, the physical layer design for TAP-to-TAP links will be entirely custom and will likely not be interoperable with any existing system. Thus, a TAP's radios must be capable of wideband operation in order to support the spec-

tral requirements of these high throughput links but cannot not be tied to any particular standard.

Another requirement imposed by the capabilities described above is the need for multiple air interfaces. An air interface is defined here as the logical abstraction of multiple radios and antennas which act together to communicate over a single link. The number of air interfaces required will vary depending on what services a particular TAP is expected to provide. For example, a TAP which provides connectivity to mobile users and uses a high speed wireless link for its own network connection would need at least two interfaces. A TAP at the core of a network which shares its wired network connection with many other non-wired nodes would need many more.

2.2 Processing Resources

The other fundamental requirement of the TAP hardware is the need for sufficient processing power to implement advanced multiple antenna wireless communications algorithms. This requirement poses a kind of cart-before-the-horse problem in that many of these algorithms are still being researched. This is especially true for TAP-to-TAP links, whose physical layer design has only recently been started. Consequently, the TAP hardware design should provide as much processing power in as flexible way as possible. Further, it should support some means of supplementing these processing resources should the base design prove insufficient.

2.3 Flexibility

A final requirement of the TAP platform is flexibility. This requirement is imposed by the wide variety of scenarios in which a TAP must operate. This is well illustrated in the example TAP network shown in Figure 1. In this example, only a few TAPs have wired network connections. Some TAPs act as routers for neighboring access points while others provide access only to mobile users. Finally, some TAPs utilize multiple TAP-to-TAP links while others require just one. Clearly the TAP platform must be very flexible, able to provide whichever capabilities are required at a particular node in the network.

3. HARDWARE DESIGN

3.1 Design Decisions

As mentioned above, the TAP baseband algorithms are not yet defined. Dedicated baseband processors are available for a wide variety of wireless networking standards, but such devices are not suitable for implementing the custom algorithms in a TAP. The device that provides the baseband processing must therefore be very flexible. Further, these baseband algorithms are expected to be very complex, requiring significant processing resources. The baseband processor must be tailored to the DSP-intensive operations, such as filtering and correlation, which are common in communications algorithms. Finally, the processor must be capable of highly parallel operation in order to realize the bandwidth and throughput goals for a TAP.

Next, the TAP baseband processor connects to multiple radios, each of which communicates via wideband analog interface. Once digitized, each analog interface will require a high throughput, high precision digital connection to the baseband processor. Assume that a TAP's baseband processor controls four radios and that each radio has a analog complex baseband interface (e.g. separate I/Q analog signals) with a bandwidth of 20 MHz. The resulting sustained throughput requirements for the baseband processor exceed 300 MByte/sec in each direction. This is a substantial requirement, generally exceeding the capabilities of processors whose primary off-chip interfaces are standard memory busses.

It became clear early in the design process that FPGAs were the only devices which could practically meet all of these requirements. Large FPGAs provide tremendous amounts of processing power, all of which, by definition, operates in parallel. All of an FPGAs interfaces to external devices also operate in parallel, significantly easing the aggregate throughput requirement discussed above. FPGAs are also extremely well suited for DSP-intensive operations. For example, large devices include more than 300 dedicated multiplier blocks, all of which can be used simultaneously.

Once FPGAs were chosen as the baseband processor, a radio had to be selected. Two options were considered here. First, it would be possible to design a radio from discrete components, implementing all of the necessary mixing, filtering and amplification in a custom circuit. Such a design would provide a completely generic analog interface and could be tailored to the desired bandwidth and radio frequency specifications. Unfortunately, the design of such a system is a very challenging undertaking which falls well outside the expertise of anyone involved with this project. Instead, a third-party radio transceiver was identified which meets the TAP radio requirements.

A final high-level design decision involves the partitioning of a TAP into multiple boards. In general, a TAP will have a minimum of three air interfaces, each equipped with four radios. Some TAPs will also have wired network connections in addition to their wireless interfaces. Assuming each wireless interface will require at least one FPGA, a TAP will consist of at least three large FPGAs plus 12 radios. Designing a single board with all of these components would be risky and very expensive. Instead, a TAP's functionality is divided across three boards. The first is a simple radio board, containing a single RF transceiver and the necessary analog-digital conversion. The second board hosts the large baseband FPGA and has slots for four radio boards. This board provides the TAP with a single air interface. Finally, a third board contains a smaller FPGA with a wired network interface. This division of hardware seems natural but poses the significant problem of designing some means to interconnect the boards. The wireless interfaces in a TAP will need to communicate with both the wired and other wireless interfaces. Given the very high data rates these interfaces are expected to support, this board-to-board communication must be very fast and have very low latency. Further, an average TAP will consist of four boards, each of which must be able to communicate with every other. If even more boards are added to a TAP (e.g. should additional wireless interfaces be required), this problem of interconnection only becomes more complicated.

A possible solution for the board-to-board interconnect is a traditional backplane architecture where each board connects to a common, parallel bus. There are a wide variety of standards for such busses, including the various flavors of PCI, which are easily implemented in an FPGA. The prob-

lems which plague such architectures, however, would prove especially troublesome in a TAP. For example, the maximum number of boards which can access the bus must be predefined when designing the backplane. Additionally, because boards are connected in parallel, they must contend for bus resources. In the worst case, a pair of boards could monopolize the bus, severely limiting the rate at which any other boards could communicate. Finally, the maximum communication rate in most parallel busses is simply too slow for the high-throughput connections which are required between boards in a TAP.

The solution which was selected for this design is, in many ways, the exact opposite of the bus architecture described above. Instead of having a connections to a common backplane, every board in a TAP is directly connected to every other. This point-to-point topology enables communication between any pair of boards regardless of what resources any other boards are consuming. A fully-interconnected design, however, somewhat complicates the architecture of a TAP. Each board must be equipped with a dedicated connection for every other board in a TAP. Adding a new wireless interface, for example, would consume an additional connection on every existing board. At first a glance, it seems problematic that the resource requirements increase with the number of boards in a TAP. However, if each board could be efficiently equipped with a large number of identical, high speed connections, this architecture is justified. Fortunately, this requirement is easily met in the TAP design, as detailed in the discussion of component selection below.

3.2 Low-level Design

This section discusses the low-level hardware design of the TAP platform. The actual design includes a large number of parts, including analog converters, memory and power regulators. Rather than list all the various parts and their individual functions here, readers can view schematics for the various TAP boards directly [4]. Instead, the primary components and key design features of each of the three TAP boards are discussed.

3.2.1 Wireless Processing Board

The choice of the FPGA for baseband processing is actually fairly straightforward. As described above, this design must provide sufficient resources to implement baseband algorithms which, though not yet defined, are expected to be very complex. As a result, the chosen FPGA should be as large as is reasonably possible. The most advanced FPGAs available at the time of this design are those in Xilinx's Virtex-II Pro line [5]. The part chosen for the TAP baseband processor is the XC2VP70, one of the largest FPGAs in the Virtex-II Pro family.

Table 1 summarizes the resources provided by the TAP baseband FPGA. It also lists the resources for a slightly less capable chip, the XC2VP50. This FPGA is pin-compatible with the larger chip; either can be mounted on the wireless processing board as the TAP baseband processor.

In addition to providing generous logic resources, the Virtex-II Pro baseband FPGA includes a number of additional features which will be critical in the operation of a TAP. The first is the inclusion of multiple PowerPC cores embedded in the logic of the FPGA. These cores are full 32-bit RISC processors whose external interfaces are tied to the device's programmable logic fabric. The TAP baseband FPGA has two such cores which operate independently. These cores are meant to compliment the operation of the FPGA by providing a means to execute pre-existing software code or to perform other processing which is not well suited for implementation in general logic. It is even possible to run a full operating system in one of the processor cores. Xilinx provides a version of Linux which will boot in the embedded PowerPC and can interact with the custom design implemented in the surrounding logic. This offers the interesting possibility of a TAP being, in essence, a PC with many wireless network interfaces. This abstraction could prove very useful when it comes time to implement higher-layer protocols (MAC, routing, etc) in a TAP. Figure 2 shows a block diagram of the board.

	XC2VP50	*XC2VP70*
User I/O	852	996
Gates	≈5 million	≈7 million
Integrated RAM	4.2 Mbit	5.9 Mbit
18x18 Multipliers	232	328
PowerPC Cores	2	
Rocket I/O	16	
Price Each	≈US$1600	≈US$2500

Table 1: TAP baseband FPGA resources

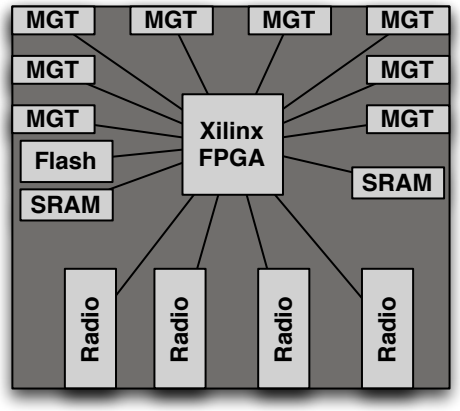

Figure 2: TAP wireless processing board

Another feature of Virtex-II Pro FPGAs key to the TAP architecture is RocketIO, Xilinx's name for its high-speed serial transceiver technology. These transceivers, generically known as MGTs (**m**ulti-**g**igabit **t**ransceivers), enable very high throughput, full duplex connections for chip-to-chip or board-to-board communication. Each transceiver is capable of communicating at 3.125 Gbit/sec over a four wire serial link. Multiple transceivers can be bonded together to achieve even faster aggregate throughput. The TAP baseband FPGA is equipped with 16 such transceivers, providing significant resources for off-chip communications. Eight MGTs are wired to off-board connectors on the wireless processing board, allowing up to eight boards to be fully interconnected when constructing a TAP.

The wireless processing board is a 16-layer, 8" (20cm) square printed circuit board. The board was manually routed, using 0.005" traces and spacing, to minimize the number

of required routing layers and maximize signal integrity between the FPGA and radio boards headers. Each radio board header has 124 digital signals routed to I/O on the FPGA. These wide, dedicated buses allow direct communication between the FPGA and the various digital interfaces of the radio boards' RF transceivers and data converters. The board has two banks of SRAM providing 18 Mbit of memory to supplement the FPGA's on-board RAM. There is also CompactFlash slot, used to store configuration bitstreams and board-specific settings like a MAC address or serial number.

3.2.2 Radio Board

A radio transceiver suitable for use in this design proved to be the most difficult part to choose. The difficulty stems from the scarcity of wideband radio chips which are not tied to a particular baseband processor. A vast majority of wireless networking designs are built to comply with one or more of the IEEE 802.11 standards and provide little flexibility beyond those specifications. A TAP's wireless interfaces, on the other hand, will not use these standards and must be designed to support a multiple antenna physical layer which is still under active development.

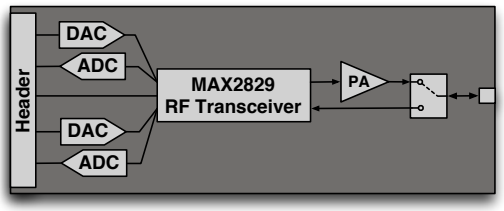

Figure 3: TAP radio board

Fortunately, a suitable radio chip was recently released which exceeds all of the requirements discussed above. This radio, the MAX2829 from Maxim Integrated Products, is a direct conversion radio transceiver which supports both the 2.4 GHz and 5 GHz ISM bands [6]. Although it is intended for use in 802.11a/b/g/n designs, this transceiver provides a flexible analog baseband interface. This interface allows the translation of any waveform, with a bandwidth up to 40 MHz, between baseband and RF, regardless of the waveform's adherence to an 802.11 standard. Further, when driven by a common reference clock, the phase coherency of the local oscillators in multiple MAX2829 transceivers is guaranteed. This feature is critical in MIMO applications as many algorithms require carefully controlled phase relationships and accurate measurements of phase among multiple antennas.

The TAP radio board's RF chain is completed by bandpass filters, a dual-band power amplifier and a Tx/Rx switch. The amplifier in the current design is capable of transmitting OFDM waveforms at 18 dBm. Future revisions of the board will likely incorporate more powerful amplifiers which will be better suited to long-range, outdoor deployment. Figure 3 shows the final structure of the board.

The TAP radio board is a 6-layer, 2x3" (4.5x7cm) PCB. The board houses all of the components necessary to provide a purely digital interface to the host board's FPGA. Dedicated linear power regulators are also used to meet the low noise requirements of the data converters and RF transceiver. The RF signals are routed to board-edge connectors to provide flexibility in the selection and arrangement of antennas.

3.2.3 Wired Network Board

The final board in the TAP platform provides a wired network connection and will generally be used by TAPs which have direct access to a internet connection. This board is based on a Virtex-II Pro FPGA smaller than the one mounted on the wireless processing board. This FPGA provides eight multi-gigabit transceivers (MGTs), six of which are dedicated to communicating directly with wireless interfaces in a TAP. The remaining two MGTs are used as gigabit Ethernet connections. The FPGA also provides two PowerPC cores, an ideal resource for implementing the various networking algorithms operating above the MAC and PHY.

The design of this board is currently underway.

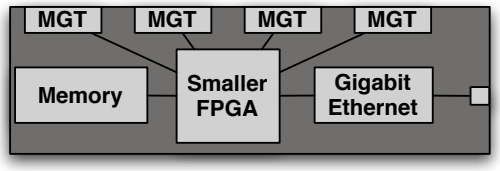

Figure 4: TAP wired network board

4. CONCLUSION

We have discussed in detail our hardware design efforts as part of the Transit Access Point project. This hardware platform has been designed from the ground up to enable our goal of a high performance, multi-hop wireless internet. Many other aspects of this wireless internet, including physical layer, MAC and routing algorithms, are under active investigation. Future publications will discuss the development of these algorithms and their implementation and deployment on this platform.

REFERENCES

[1] A. Gupta, A. Forenza, and R.W. Heath, Jr., "Rapid MIMO-OFDM software defined radio system prototyping," in *IEEE Workshop on Signal Processing Systems*, 2004, pp. 182–187.

[2] R.M. Rao, W. Zhu, S. Lang, C. Oberli, D. Browne, J. Bhatia, J.F. Frigon, J. Wang, P. Gupta, H. Lee, D.N. Liu, S.G. Wong, M.P. Fitz, B. Daneshrad, and O. Takeshita, "Multi-antenna testbeds for research and education in wireless communications," in *IEEE Communications Magazine*, 2004, vol. 42, pp. 72–81.

[3] P. Gupta, Weijun Zhu, M.P. Fitz, Heechoon Lee, D.N. Liu, and S.W.G. Wong, "Field test results for space-time coding," in *Asilomar Conference on Signals, Systems and Computers*, 2003, pp. 243–247.

[4] *http://taps.rice.edu/hardware/.*

[5] *http://xilinx.com/products/tables/fpga.htm#v2p.*

[6] *http://maxim-ic.com/MAX2829.*

ACKNOWLEDGEMENTS

Special thanks to both Maxim and Xilinx for their support throughout this project. This project is funded by NSF grants ANI-0325971, EIA-0224458 and EIA-0321266.