A Wireless Adapter Architecture for Mobile Computing

John Trotter and Mark Cravatts
AT&T Bell Labs.,
600 Mountain Av.,
Murray Hill,
NJ 07974

Abstract

We present the architecture of an indoor wireless adapter card for use in nanocellular wireless networks. The card, called FAWN (Flexible Adapter for Wireless Networking), includes an RF modem, a CPU and an interface to a host computer. The card supports mobility, allowing portable machines to handoff between base-stations while maintaining a high speed, low latency wireless connection. By executing the MAC algorithms on the FAWN card the host computer is able to run its applications at full speed. In addition, this approach enhances the card's applicability for use in a wide variety of machines as it presents a simple and universal interface. The on-board CPU can also be used as the controller, allowing the card to be used as the central component in embedded wireless designs. The card has been constructed and is being used as part of a larger wireless project in AT&T.

Introduction

Advances in technology have enabled high speed wireless communication and this has spurred the growth of wireless, and now mobile computing systems. This technology has enabled many new wireless services and systems including a proliferation of wireless LANs which make the first step toward a mobile computing environment. We are constructing a wireless computing testbed to investigate how mobile computing system will be architected, integrated into existing and future infrastructure and how applications that run on the system will interact with the machine and other applications.

Several models for mobile and wireless computing have been proposed. Xerox's tab [8] uses a small hand-held device, called a tab, as a terminal which then interacts with the surrounding network of machines which contain the intelligence. This allows users to pick up and use an arbitrary tab and immediately have access to their environment. The tab system uses an infra-red link for data receive and transmit, and allows communication in office size cells which are connected to the installed backbone network. The Infopad system [5,9] uses a similar approach of a terminal device, this time connected using a higher speed RF modem to access interactive data. The Infopad also relies on the network to provide the resources, acting as a terminal for the data.

Intermediate approaches split the intelligence between the device and the network. Some of the computation can be carried out using powerful processing resources in the backbone system instead of on the mobile which has limited processing resources as well as a limited power budget. Applications take advantage of as much communications bandwidth as is available by adaptively altering the amount of processing on the backbone network verses the portable machine.

A third approach has the intelligence in the mobile device and uses the network to access other devices on a peer to peer basis. This approach is similar to the current model of networked computing, and this model is supported by wireless LAN systems running Mobile IP [6]. Mobile IP allows the definition of a "mobile subnet" which has many mobiles associated with it. When communicating with a mobile machine, data is sent to any one of several fixed hosts associated with that subnet. These hosts either know where the mobile is (i.e. which base station the mobile is communicating with), or can find out by asking a set of other base stations. The WaveLAN system [10] is a wireless LAN system that allows wireless extension of existing Ethernet networks, and has been used as the physical layer for several mobile computing projects.

We envisage a model of wireless untethered computing which allows continual connection of mobile machines to the network as users move around their office, corridors and conference rooms. The system will support several models of access, from terminals to intelligent mobile hosts. In order to support this model, portable machines must be equipped with suitable wireless interfaces, wireless base stations must be installed and the backbone network must be enhanced to support mobile users [1,5,7,11]. Our objective is to develop an indoor wireless computing environment that supports continual access by heterogeneous end systems in the presence of mobility, supports location aware applications and supports end-to-end performance requirements over the network. In order to be able to investigate these issues we needed to have complete control over the software and hardware of the wireless interface. Existing wireless designs were unsuitable for our experiments so we built our own hardware which allowed us to experiment with innovative handoff and MAC schemes. We are also able to use this approach to experiment with applications that are split between the mobile and the backbone network.

The System

A high level view of our system is shown in Figure 1. The model uses a wide area ATM backbone network, connected to local area communication fabrics which, in turn, connect to endpoints. The endpoint machines can be base-stations which provide the wireless hardware necessary to support a final hop to a mobile, a service provider such as a computation server or database server or a user's workstation. The complete system is called SWAN (Seamless Wireless ATM Network) [1]. A mobile computer can connect to the base-station using a wireless adapter, as can "personal terminals" which are devices that act as terminals onto the network. Each base-station normally provides an access point for a few offices, although cells may be smaller or larger, depending on the demand in that area. For example, a conference room may have two or more cells to accommodate many users in a small area, or the cells may be far larger in areas of low use such as corridors or outdoor walkways. A mobile host is able to roam around while maintaining connectivity, handing off between base-stations on the network even if the base-stations are on different local fabrics. Thus we support ubiquitous mobile computing anywhere there is a base station. If a mobile becomes disconnected for any reason it can reconnect once it moves within range of a base station. In addition to communicating using wired base stations, a group of users can set up a temporary network so they can exchange files or use a service provided by one of the hosts.

One of our main objectives is to support multimedia access over the network for the mobile systems. An important part of being able to provision multimedia services is being able to provide some form of Quality of Service (QoS) guarantee for applications that use the network. Asynchronous Transfer Modem (ATM) is expected to be able to satisfy these requirements by providing per Virtual Channel (VC) performance guarantees. Research investigating per VC QoS can be extended to the wireless domain if ATM is used from end-to-end. In addition, an ATM interface allows a mobile host to make informed packet routing decisions on to the wireless hop, whose available bandwidth will probably be lower that the wired backbone and may suddenly change. The QoS model also allows renegotiation of the bitrate of each VC which supports the changing bitrate which may well occur in a wireless system. By directly routing ATM packets through the basestation we avoid any protocol conversion or data repackaging which may of been necessary if a proxy at the base-station intercepted the data. Therefore, we chose to use ATM as our backbone network as well as for the wireless hop.

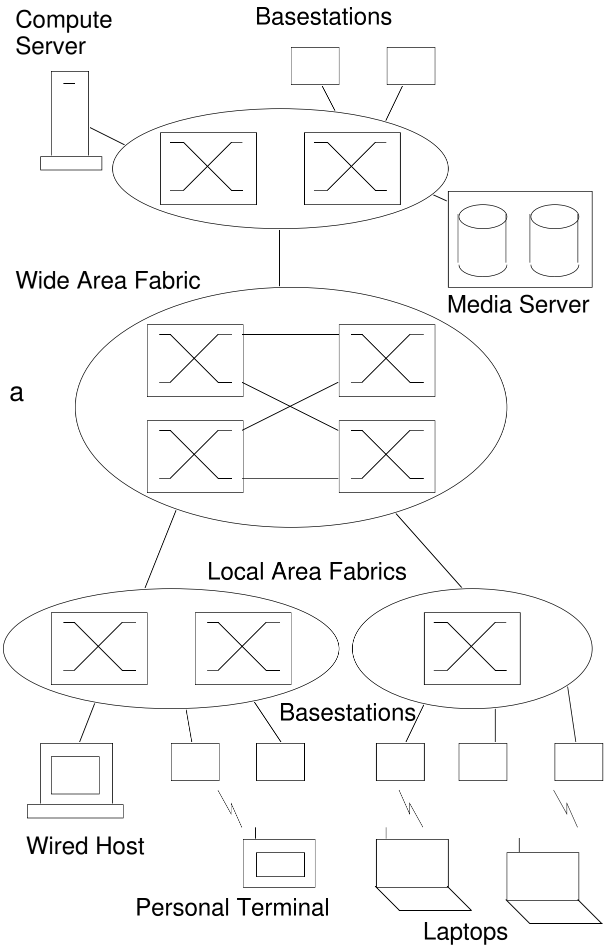

Figure 1. System overview.

To fully support multimedia traffic the network has to appear as seamless as possible. During handoff the data stream has to be rerouted from the old base-station to the new one with the minimum of loss so that the network looks continuous to machines using it. An important feature in our mobile system is the ability to handoff at high speed. This minimizes cell loss and allows for a high bitrate to be guaranteed and maintained when a VC is established. We use mobile directed handoff which allows the mobile to control the handoff. When the received power is too low the mobile begins searching for a better base-station, and if it finds one, it initiates a handoff. Because both the base-stations and the mobile are all informed during the move, the handoff processes can be carefully controlled to ensure that no data is lost in the transition.

Several other approaches to mobile computing use a direct sequence spread spectrum modems such as WaveLAN. While newer versions of the WaveLAN system offer up to 20Mbits/sec, this bandwidth has to be shared among many users. The problem of providing a guaranteed bitrate is complicated because the protocol does not allow guaranteed access to the channel. Other spreading codes can be used to increase the number of channels, but the aggregate bandwidth available using this technique is less than using a Frequency Hoping Spread Spectrum (FHSS) modem [4]. We chose to couple a MAC protocol (based on token passing) which guarantees channel access with a FHSS modem in order to support a high aggregate bandwidth. The modem we use gives us up to 1M bits/sec over the wireless link in the 2.4GHz ISM band, allowing each channel a guaranteed bitrate. Each mobile is assigned one virtual channel, and several modems allow many channels to be used simultaneously up to a limit imposed by the RF bandwidth. If even more channels are needed they can be time division multiplexed onto a single modem's channel.

In our system we consider a nanocell indoor wireless environment with fixed base stations that are interconnected with an ATM network. ATM packets are used end-to-end the network, and the packets are encapsulated for the final hop from the base station to the mobile unit. The system supports mobile directed handoff from one base station to another, which is done at very high speed and with low latency as part of the MAC layer.

The Architecture

In order to support the above requirements for multimedia the adapter card had to be able to handle ATM data, support fast handoff and provide the wireless bandwidth needed. The adapter card, called FAWN (Flexible Adapter for Wireless Networking) provides the necessary computing power to run the MAC and handoff software and provides an interface between the host computer and the RF modem. To enhance reuse, the FAWN adapter was designed so that it could be used as the wireless adapter card in either a base-station, a mobile or an embedded application such as a personal terminal. In order to ensure the widest compatibility with desktop and portable computers we chose to use a PCMCIA interface for the FAWN adapter.

The FAWN adapter set consists of two cards, a modem card and a CPU card, and several interfaces. The CPU card provides the CPU, data and program memory as well a buffer for data communication between the CPU and the PCMCIA interface, or between other processes running on the system. It also provides an interface for the modem card, a PCMCIA host and a peripheral card which can be used by other systems such as a personal terminal [3] or embedded base station. The CPU card can accept one or more modem cards which provide hardware for packet buffering, clock recovery and the interface to an RF modem

The FAWN adapter set can be used in one of several configurations. The base station configuration is shown in Figure 2. The base station controller is a computer which has an interface to the backbone ATM network as well as an expansion bus which is used to plug in several PCMCIA adapters. Each PCMCIA adapter accepts a FAWN CPU card which in turn interfaces with several modem cards. Because we use a FHSS modem, one modem card is required per channel, therefore so that the base station can provide enough channels for high demand areas several modem cards must be supported.

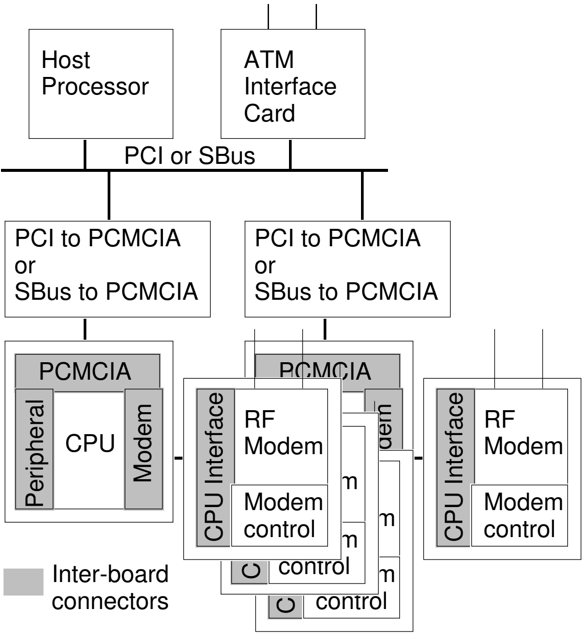

Figure 2. The FAWN card used in a base station

When used as an adapter for a mobile system, the FAWN card interfaces to one modem card and the PCMCIA connector in the mobile host. This configuration is shown in Figure 3.

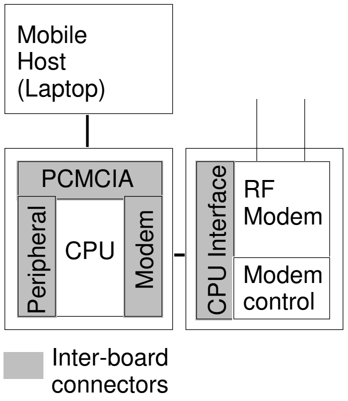

Figure 3. The FAWN card used in a mobile host

When the FAWN cards are used as an embedded base station the peripheral interface is used to support a network interface, such as ATM. This configuration is shown in Figure 4. One or more modem cards can be plugged into the CPU card, supporting fewer distinct channels than the base station. This version of the card can be used in areas of medium demand, like in an office or corridor

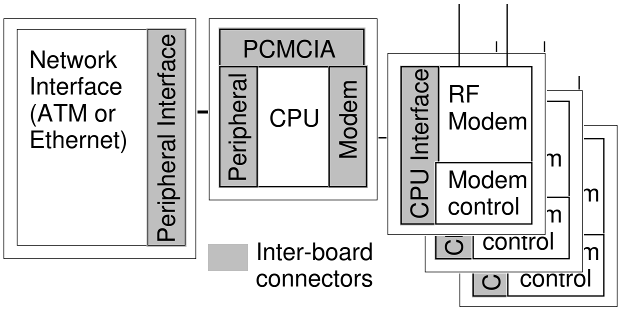

Figure 4. The FAWN card used as an embedded base station

Another incarnation is a personal terminal [3]. Here a peripheral with video and audio I/O capability is plugged into the peripheral connector allowing the FAWN card to provide the wireless link. In addition the FAWN card provides the necessary CPU to run the terminal's application. Here we take advantage of the card's ability to multitask, so that the MAC layer can run on the same CPU as the PDA application.

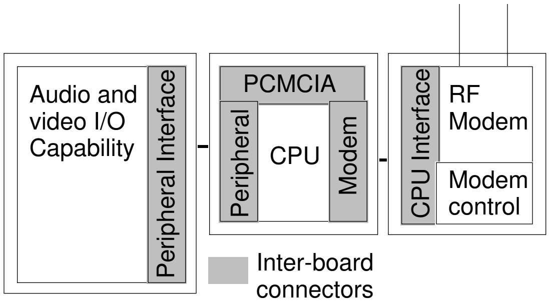

Figure 5. The FAWN board used in a personal terminal

Detail of the FAWN card is shown in Figure 6. The main parts are the PCMCIA interface that provides connectivity to the host machine, the CPU and RAM subsystem which provides the raw processing power, the dual port RAM system which provides a fast way of transferring information from the host CPU to the adapter card, and the modem interface which provides packet buffering and provides the logic to interface with the RF modem.

The following sections describe each part of the architecture in detail:

- PCMCIA Interface. The PCMCIA interface is implemented in an FPGA and provides an interface between the host computer and the adapter card. In normal operation, the interface allows the host machine to access control registers that configure the interrupts for the CPU on the card and control the reset, as well as providing access to the Dual Port RAM (DPR). The PCMCIA interface also allows the host computer to access the peripherals that the FAWN card's CPU would normally access by stalling the CPU, these include the other side of the DPR as well as the memory, the RF modem interface and the real time clock.
- Dual Port RAM (DPR). The DPR provides a high speed interface between the host computer and the adapter card. The DPR is used to implement the queues necessary for communication between the host and FAWN. By using a RAM structure (as opposed to a FIFO) the implementation of arbitrary queue structures with differing sizes and priorities is easy. When FAWN is used as an embedded controller, communication between the MAC process and the higher level processes still continues using the DPR, allowing a standard interface to be presented to all applications, wherever they run. The DPR provides support for semaphores to ensure that two identical locations are never accessed at the same time by the host and FAWN. 32K bytes of DPR are used in the prototype. The DPR also facilitates the conversion of the 32 bit word used by FAWN's CPU to the 16 bit word needed by the PCMCIA interface.
- The adapter card contains a 32 bit RISC CPU, the ARM 610 [2] running at 20MHz. This microprocessor was chosen because of its low power as well as it being a powerful 32 bit CPU. The prototype systems includes 4M bytes of memory for program and data and interfaces to another FPGA which controls the modem and provides buffering for the data packets. The ARM610 is powerful enough so that single CPU can control many modem boards as well as provide the CPU to control a peripheral board which

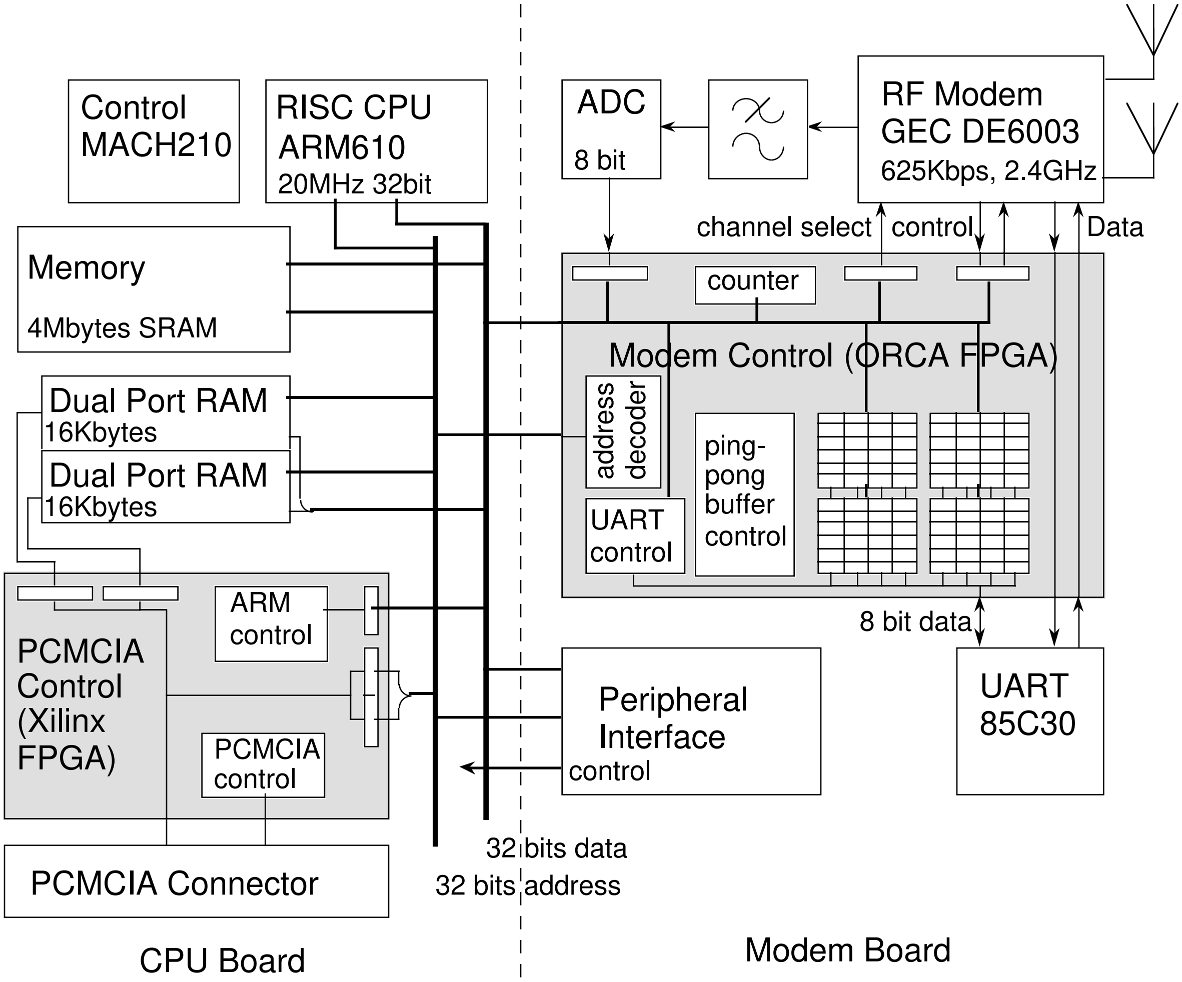

Figure 6. Detailed architecture of the FAWN board

contains an external network interface or other circuits. The CPU is also used to execute Forward Error Correction (FEC) code which ensures that erroneous cells are at least detected if they cannot be corrected.

- RF Modem Interface. The modem interface is implemented in an FPGA and includes a packet buffer and a hop controller. The packet buffer is 64bytes long and allows the buffering of a complete ATM packet as well as extra space for encapsulation and error control bits. The FPGA implements four such buffers, two for transmit and two for receive. During a receive operation one of the two receive buffers slowly fills up (at the data rate). When the buffer is full an interrupt is generated so that the CPU can empty the buffer all at once, meanwhile the other buffer begins to fill from the data stream. During transmit operation the CPU fills a transmit buffer then sets a bit to tell the FPGA that the buffer is ready to be sent. The buffer is then made available to the UART at the data rate. Once the buffer has been sent the FPGA generates an interrupt so that the CPU knows that the buffer is now available to be filled. Meanwhile the second transmit buffer can be sent. The provision of these buffers allows the CPU to be decoupled from the low level byte based transceive operations. The FPGA also provides the necessary control logic for all the modem control including hop control (selecting the channel for the modem). Because we use an FPGA we are able to interface with other faster modems as they become available.
- The modem FPGA also provides a real time clock which can be read and reset by the CPU as well as being able to interrupt the CPU when a programmed time has been reached. In addition, there is an interface to an Analog to Digital Converter (ADC) which converts the received signal strength to a digital value and makes this available to the CPU. The ADC can also interrupt the CPU once a threshold signal level has been crossed, this facilitates interrupt driven handoff. The FPGA also provides an interface to a UART which takes the raw bitstream from the modem and converts it to 8bit data bytes. We use SDLC encoding, which is a synchronous protocol that

guarantees that there are enough transitions to ensure that the UART remains synchronized.

- RF Modem. We use a GEC Plessey frequency hopping spread spectrum modem (DE6003) which operates in the 2.4Ghz ISM (Industrial, Scientific and Medical) band. The modem uses gaussian minimum shift keying as the modulation technique and the parts we have operate at a bitrate of 625K bits/sec (future pin-replaceable modems will operate at 1M bits/sec). The modem can operate at one of 83 channels, assigned for use on this channel by the FCC. The RF modem has two antenna ports for antenna diversity, so that the antenna with the best signal strength can be chosen during receive.
- Peripheral Interface. The design includes a peripheral interface that allows additional peripherals to be plugged in. Examples include an external Ethernet or ATM adapter or a display with an audio I/O device allowing the card to be used in a personal terminal.

Software

The software is split between the FAWN adapter and the host CPU, a representation is shown in Figure 7. When FAWN is used in an embedded system all the software runs on the CPU as different processes. To enable this multiprocess software structure, the FAWN system runs a basic kernel which provides support for threads and schedules the MAC tasks as well as the queuing scheme for the ATM packets. The CPU also executes forward error correction, error detecting and correcting packets of data. Because the card runs a kernel, it is simple to implement another thread which might be responsible for running ATM adaptation layer code. By using the FAWN board to execute the bulk of the code we are able to present a standard interface and therefore improve compatibility over the many different platforms that support PCMCIA. Running the time-critical code on the adapter ensures that the data is handled in a timely manner.

The MAC software runs as a high priority interrupt driven process on the FAWN CPU. On receiving an interrupt the MAC retrieves cells from the packet buffer, error checks and corrects the cells, then places the cell in the DPR interface. The queue manager operates using interrupts from the host and organizes packets in the DPR so they can be read by the host ATM interface. When FAWN is used with a host CPU, the host CPU reads the packets from the PCMCIA interface and executes the adaptation layer which makes the cells available to the application (in the case of the mobile host) or the ATM interface card. If the process that uses the cells runs on FAWN then it runs as another process thread and communicates with the queue management software using software interrupts.

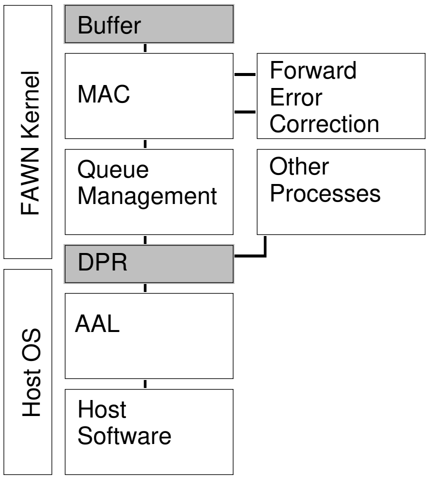

Figure 7. The software architecture

The system that uses the FAWN card, SWAN, also implements mobile directed handoff. Therefore the MAC interface is able to signal the base station when a move is about to occur, the old base station can then reroute the data stream to the new base station.

Implementation

The FAWN card is implemented as two PC boards, the modem board and the CPU board that plug together, these are shown in Figure 8. The two boards are designed to fit in the floppy drive bay of an AT&T GIS 3181 laptop computer, which is the machine we are using as the mobiles in our prototype. Eventually we plan to implement the design in a type III PCMCIA card that can be used in a wider variety of portable machines. Several modem cards can be plugged into the CPU card for use in a base station, and several CPU cards can be plugged into PCMCIA slots in the host computer. We are currently using Pentium based PCs as the base stations, and these interface with the ATM interface card, currently from FORE. The prototype is installed in our office environment and allows users of mobile systems to remain connected to the backbone network as they move.

Conclusion

In this paper we have described the architecture of an adapter card designed to support wireless mobile computing. We architected and built the card which is now in use in a prototype indoor wireless network. The card was architected and designed in a very flexible way, allowing the same card to be reused in base stations and mobiles. We are able to support high speed interfaces from the host machine to the adapter card using the DPR. Because the FPGA handles the byte level communication operations and only presents the CPU with complete packets the CPU can be left to run MAC level code and still have enough capacity to execute embedded application programs. The MAC and handoff design is aided by the interrupt driven ADC and the real time clock which are available on the card. The processor on the card is responsible for all the low level operations, simplifying the interface presented over the PCMCIA interface which in turn simplifies the software that has to be implemented on the host. This increases the ease with which the system can be integrated with new hosts.

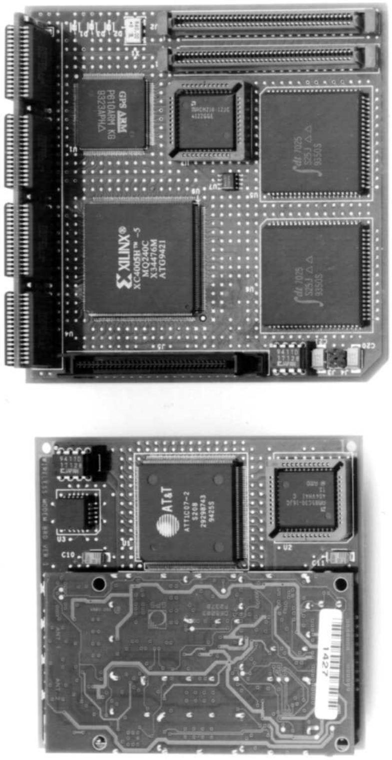

Figure 8. The CPU board and Modem board that make up the FAWN set.

References

[1] P. Agrawal, A. Asthana, M. Cravatts, E. Hyden, P. Krzyzanowski, P. Mishra, B. Narendran, P. Mishra and J. Trotter. "A Testbed for Mobile Networked Computing," Proceedings IEEE Communication Conference, Seattle, WA, June 1995

[2] ARM610 Data Sheet, Publication No DS3554, Issue No 2, March 1993 GEC Plessey Semiconductors, Sequoia Research Park, 1500 Green Hills Road, Scotts Valley, CA 95066

[3] A. Asthana, M. Cravatts and P. Krzyzanowski, "An Indoor Wireless System for Personalized Shopping Assistance," Proc. IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA. December 1994.

[4] D. Bantz and F. Bauchot. "Wireless LAN Design Alternatives" IEEE Network March/April 1994 Vol. 8 No. 2, pp.43 - 53

[5] B. Barringer, T. Burd, et. al. "Infopad: A System Design for Portable Multimedia Access." In Wireless 1994, Calgary, Canada, July 1994.

[6] J. Ioannidis, D. Duchamp and G.Q. Maguire. "IP-based Protocols for Mobile Internetworking" Proc SIGCOMM 91, 1991 pp. 234-245.

[7] K. Keeton, B. Mah, S. Seshan, R. Katz, and D. Ferrari. "Providing Connection-Oriented Services to Mobile Hosts." In Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing, pages 83-102, Cambridge, Massachusetts, August 1993.

[8] B. Schilit, N. Adams, R. Gold, M. Tso and R. Want, "The PARCTAB Mobile Computing System," Proceedings of the Fourth Workshop on Workstation Operating Systems. Napa, CA, October 1993, pp. 34-39

[9] S. Sheng, A. Chandrakasan and R.W. Brodersen, "A Portable Multimedia Terminal," IEEE Communications Magazine, December 1992, pp. 64-67

[10] B. Tuch, "Development of WaveLAN(tm), an ISM Band Wireless LAN," AT&T Technical Journal, July/August 1993, pp 27-37

[11] M. Weiser. "Some Computer Science Issues in Ubiquitous Computing." Communications of the ACM, 36(7):209-220, July 1993.