# A Testbed for Mobile Networked Computing

Prathima Agrawal, Abhaya Asthana, Mark Cravatts, Eoin Hyden, Paul Krzyzanowski, Partho Mishra, B. Narendran, Mani Srivastava and John Trotter

> AT&T Bell Laboratories Murray Hill, New Jersey, 07974, USA

#### Abstract

The rapid deployment of wireless access technology, along with the emergence of high speed integrated service networks, promises to provide users with ubiquitous access to multimedia information in the near future. We are building an experimental testbed system, SWAN (Seamless Wireless ATM Network), to mimic this emerging networking environment. Our wireless access network is organized according to a nanocellular design with base stations serving as a gateway for communication between the wired network and the mobile hosts in a cell. Normally, a mobile host sends and receives traffic through the base station in its current cell. But SWAN also supports direct ephemeral networking between a limited number of cooperating mobile hosts within a small domain. The heart of the testbed is a networking subsystem, FAWN (Flexible Adapter for Wireless Networking) that interfaces the standard PCMCIA bus to an RF modem. The FAWN interface is used with a PC or workstation connected to a wired backbone network or a portable device such as a laptop or palmtop computer. In addition, a user interface consisting of an LCD display, audio I/O, and a bar code reader has been built. When interfaced with FAWN this results in a portable wireless multimedia terminal.

### **1. INTRODUCTION**

The recent proliferation of wireless technology has not only enabled tetherless access to several conventional communication systems and services but also provided opportunities for creating new systems and services [1,2,7,8]. We believe that the wireless technology which enabled cordless and cellular telephones and had such an impact on use of communications services will have an even greater effect on the way we use computing services.

We are constructing a mobile computing testbed, named SWAN (Seamless Wireless ATM Network), in order to fully understand the systems aspects of providing ubiquitous and tetherless computing environments. The testbed will be used to study the consequences of mobility and the use of wireless communications links in a distributed environment. The study will encompass diverse aspects of mobility and wireless access such as their effects on system architecture, software and the structure of applications as well as the effectiveness of algorithms that manage connection establishment, handoff and rerouting.

The following sections describe the current state of our testbed in terms of the type of environment in which it operates, the enabling hardware technology that provides us with basic wireless connectivity, the low-level system software which abstracts the raw hardware, higher-level system software which manages connections, rerouting and handoffs and finally, the types of applications we aim to support.



Figure 1. Seamless Communication Model

### 2. ENVIRONMENT

Our view of the future networked computing environment which provides seamless, mobile, personal communications services is shown in Figure 1. The network model is based on a wide area, broad-band, wired backbone network with wireless hops to the mobile end systems (hosts). The wired network may consist of a hierarchy of local area networks, metropolitan area networks, and wide area networks. The wireless part of our model incorporates a radio-based network consisting of small office-room sized cells (25-100 square meters) inside buildings. The *base stations* act as gateways for communication between the wired network and the mobile end systems in a cell.

SWAN supports communication between mobile hosts both with and without the aid of a base station. In the first case, a mobile end system sends and receives all its traffic through the base station of its current cell. The base station in turn relays the data to another mobile end system within the same cell, or to a mobile end system in a remote cell, through the wired network and the base station responsible for the remote cell. In the second case, SWAN enables the creation of a limited life time or *ephemeral* network for direct communication between mobile end systems, without the assistance of a base station. Such connectivity would naturally be limited to a set of mobile end systems that have agreed to cooperate and are all contained within a limited geographical domain. For example, a set of laptops engaged in a session within a room, even in the absence of a base station in the vicinity. In this context, we are initially investigating simple point-to-point network topologies.

Mobile hosts can range from personal multimedia terminals, with minimal user-interface processing capability and which rely on network servers for computation (such as Berkeley's InfoPad [3] and our own Personal Mobile Terminal, PMT, described later in Section 3), to systems such as laptops that provide local application level computation and storage.

In general, it is perhaps too simplistic to characterize all mobile users into one category. Clearly, some users may perform a lot of stand-alone processing and little communication; some others may use mobile terminals mostly for messaging and yet others for information entry and access. The data handled may be text and still images as well as video and voice. Within wired networks, Asynchronous Transfer Mode (ATM) is expected to be capable of satisfying the various Quality of Service (QoS) requirements of these data types. The desire to provide mobile applications with such benefits has led us initially to an investigation into the possibility of asynchronously multiplexing the wireless bandwidth provided by a base station among the mobiles it services. Use of Virtual Circuits (VCs) simplifies switching and provides a mechanism for supporting per connection QoS contracts. While the extension of VC oriented packet switching paradigm in the wired backbone seems attractive, much research is needed to establish the practicality of this to the wireless part of the network [4]. This allows the service quality guarantees made in the wired network to be extended in an end to end manner. However, this also requires the ability to establish, reroute, and tear down VCs to and from mobile end systems, and to provide QoS guarantees for these VCs. The negative impact of this is the added overhead on the bandwidth limited wireless medium.

### **3. TESTBED HARDWARE**

Flexibility is a key design consideration. Our objective is to be able to administer and control the distributed testbed at all levels; to experiment with resource allocation, network control and fault management algorithms, and to prototype novel services and applications.

The baseline architecture of the system is shown in Figure 2. The base stations connect to the backbone network via an ATM interface card, and connect to the mobiles through custom radio interface cards. The radio interface cards communicate with similar cards installed in the mobile hosts, such as a laptop, which use them to access the bit stream. Additionally, the base station can communicate with a radio interface card used as an embedded controller in a Personal Mobile Terminal (PMT) system as shown in Figure 2. The backbone allows the base stations, and therefore the mobile units, to access services

provided over the network.



Figure 2. Testbed Hardware Architecture

Other approaches to wireless indoor data communications such as WaveLAN [5] use direct sequence spread spectrum in both the 900MHz and 2.4GHz ISM bands. The new version of WaveLAN provides 2Mbits/sec of aggregate bandwidth, but has to share the channel with other users, decreasing the available bandwidth as the number of users increases. In contrast, we use a GEC Plessey frequency hopping spread spectrum (FHSS) modem. This modem has higher per-channel bandwidth (1Mbits/sec) and can accommodate more users without compromising the bit-rate. The flexibility and computing power of our interface card gives us access to the lowest levels of the hardware and incoming and outgoing bit streams. This allows us to experiment with issues related to the MAC layer and handoff algorithms, provides us control at all levels of the software, and enables us to make power, bandwidth and processing capacity tradeoffs by appropriate partitioning and assignment of the software modules.

### 3.1 The FAWN Adapter

The key element in both the base stations and the mobiles is an adapter card. The card, called FAWN (Flexible Adapter for Wireless Networking), interfaces the base station or mobile with an RF modem and provides processing power to support media access in the testbed.

The base stations in our testbed are PCs to which several FAWN cards can be attached each controlling one or more RF transceivers. The FAWN card uses a PCMCIA bus to interface with the host computer. Similarly, a laptop computer with a PCMCIA slot can become a mobile host by plugging in a FAWN card.

The card has a low power, embedded RISC processor (ARM 610 CPU) that is responsible for controlling the RF modem and

other peripherals as shown in Figure 3. The FAWN card is configured for use with 2.4 GHz ISM (Industrial Scientific and Medical) band frequency hopping spread spectrum transceivers, although the transceiver interface can be easily modified by reprogramming some components. The processor runs at 20 MHz and provides sufficient processing capacity for performing the kernel, signaling and transport protocol functions. In case of the Personal Mobile Terminal, the processor on the FAWN card also supports parts of distributed applications.

The communication between the ARM processor and the host processor (in a base station or a mobile laptop) is achieved using a dual-port memory based interface over the PCMCIA interface. The interface is implemented in an FPGA (Field Programmable Gate Array). Two access modes are supported, one mode accesses the dual port RAM at full speed, the other mode accesses any of the CPU's peripherals, but takes longer because the interface has to arbitrate for the FAWN CPU's internal bus. Because each side of the dual port RAM can be accessed at full speed by both the host CPU and by the FAWN CPU the data transfers can occur at the maximum speed.



Figure 3. Architecture of the FAWN Adapter.

The modem controller is implemented in another FPGA and implements many of the low level functions that are necessary to support wireless access protocols. The RF modem is a 2.4GHz FHSS modem, providing a logic-level interface for data and control, as well as an analog received signal strength indicator. This band permits 83, 1MHz channels for frequency hopping. The GEC Plessey modem can support 83 channels and currently provides 625kbits/sec as raw bandwidth per channel (the same modem will go to 1.2Mbits/sec in the next version). The modem's interface allows the selection of 1 of 83 channels, the power level, and 1 of 2 antennas. The modem supplies a bit stream to the UART during receive and accepts a bit stream during transmit. The UART converts the bit stream from the modem to bytes during receive, stripping the relevant synchronization bits and providing bytes to the controlling FPGA. During transmit the UART adds synchronization bits and feeds the bit stream to the modem. The FPGA includes four 64 byte buffers which store packets of data to and from the UART. This allows the UART to asynchronously transmit and receive data without having to interrupt the FAWN CPU. The FPGA includes a resettable counter that operates at 1MHz and can be used as a real time timer for protocol and task scheduling functions. An analog to digital converter (ADC) and low pass filter is provided to allow the received signal strength to be read by the FAWN CPU. The FAWN card includes 4 Mbytes of RAM for program and data.



Figure 4. FAWN Cards.

The initial design shown in Figure 4 is implemented on two PC boards that fit in the space of the floppy drive of an AT&T 3181 laptop. The AT&T 3181 is the machine that we are using as the first mobile computers and we have plans to implement the next version in a type III PCMCIA form factor so the card can be used in a wide range of mobile computers.



Figure 5. PMT: A Handheld Multimedia Terminal

### 3.2 Personal Mobile Terminal (PMT)

An expansion slot is included on the FAWN card to allow other peripheral boards to be interfaced and be controlled by the FAWN CPU. The PMT uses this facility to build a stand-alone multimedia terminal.

The objective of the PMT is to present a simple user interface (see Figure 5). On the front of the box is a 320x240 dot-matrix LCD display along with three control buttons (whose functions are specified dynamically by the PMT server). It

provides bidirectional audio capability and has a set of headphones with a directional microphone attached to it. The side of the box has a bar code scanner used for scanning UPC codes. Another button on the front activates the scanner. All this circuitry resides on a peripheral card that attaches to the FAWN processor card.

PMT differentiates itself from PDAs and other small computers not only in its set of input/output interfaces but in that its functionality is determined by the server with which it communicates - i.e. its features and services are *context* dependent. By design there is minimal computation performed in the portable unit. Most compute intensive tasks are done in the server and the nature of information transfer in most PMT applications tends to be asymmetric in nature. Thus, while the down link from the server can support high quality digital images, the bandwidth requirement for the reverse link from the PMT is considerably less.

### 4. TESTBED SOFTWARE

Within the SWAN testbed, system software (called ETHERWARE) includes an embedded real time kernel for controlling peripherals on the FAWN card, as well as operating system and protocol support software for applications that run on the base stations and the mobile hosts. Figure 6 shows the structure of the host operating system and FAWN card software.

### 4.1 FAWN Embedded Software

Software within the FAWN interfaces must provide a number of quite demanding services. When configured as a wireless network interface, and connected to a base station, the FAWN embedded software must preserve the QoS requirements of any data which it handles while concurrently maintaining temporally sensitive functions such as the hopping sequence of the RF modem.

The ARM610 processor on the FAWN cards executes a real-time kernel which provides support for synchronization and the execution of multiple tasks according to their temporal requirements. A console driver is provided to relay any status and debugging information to the host via a circular buffer residing in the dual ported memory which appears in the host's PCMCIA address space. Two system tasks, the modem manager and the PCMCIA manager run on the FAWN card. The modem manager manages the RF modem and performs all of the functions required to support basic medium access and the transfer of raw data across the wireless channel. Within this task, timeliness is critical. The PCMCIA manager is responsible for maintaining the shared memory interface between the FAWN card and its host. While its timeliness is not as crucial as that of the modem manager, it is required to process packets which cross the shared memory interface in a manner which maintains their OoS.

When a FAWN card is interfaced to a handheld Personal Mobile Terminal, the system software has to provide an environment for the execution of applications in addition to maintaining these time critical functions. In this case, the modem and PCMCIA managers perform their usual functions with the exception that the PCMCIA manager communicates directly with low level protocol software. While the ARM610 is quite a powerful processor and has a low power consumption, the main goal of the system and application software, when operating in this configuration is to conserve power. To meet this goal, some applications may be split, so that they perform the bulk of their processing on a base station and use the wireless communications channel to interact remotely with their user.



Figure 6. Software Structure.

### 4.2 Base Station Software

Base stations perform a number of important functions including connection management, switching packets among mobiles and acting as gateways between the wireless network and the backbone. Since the cost of a single base station will greatly affect the overall cost of the network, one of our aims is to build base stations as cheaply as possible. This might be done using special purpose hardware which is optimized for performing the required switching functions. However, the nature of this type of network is such that it may benefit from having some processing power and storage at each base station. These would be used to hold and maintain local segments of distributed databases and to provide a nearby source of computational resources for use by less well endowed mobiles. The amount of processing power and storage that is useful at a base station is one of the areas which we are currently investigating.

Within the testbed, each base station is a Pentium class personal computer, running the UNIX operating system. Within the base station operating system, two types of device driver code are required. One driver module is responsible for managing the PCMCIA controller and, for each PCMCIA device which may be inserted into any of the implemented slots, a driver is required to attach that device to higher levels of system software. The PCMCIA management module takes advantage of the information provided by the controller to reconfigure the system software as cards are inserted and removed. The FAWN device driver interfaces the card to the low layers of the network protocol stacks. In the initial implementation, this interface provides for the transfer of ATM cells between the host protocol software and the FAWN card. Later refinements may offload some of the lower layer protocol processing onto the FAWN card's ARM610.

Applications gain access to the network via sockets. The resources for the socket and protocol code is shared by all socket instances. A socket represents a private instance of the ATM protocol stack. In addition to its use in the base stations, this type of host operating system is also used within the larger mobile hosts such as laptops.

ETHERWARE provides the basic support for our signaling, protocol and gateway implementations, and also enables us to provide local processing and storage for any distributed management algorithms and applications which require them. One interesting avenue of research is the allocation of resources within a base station, particularly with respect to providing processor time QoS contracts to applications while meeting the switching requirements of the local network traffic. We aim to investigate this to determine the extent to which operating system techniques which provide application level QoS contracts are relevant within a base station.

The following section discusses mobility management, which is typical of the type of distributed algorithms which make use of the local processing power provided by our base stations.

### 5. ACCESS, HANDOFF and MOBILITY MANAGEMENT

The key algorithms in the ETHERWARE implementation are those used to handle wireless access and mobility. These algorithms use a model in which the service quality requirements of applications are used to guide the operation of the protocols. There are four principal entities in our testbed: mobile hosts, base stations, wired switches, wired hosts as shown in Figure 1. These entities communicate to each other via a "signaling" protocol which handles addressing, location and data forwarding functions.

### 5.1 Physical and Media Access Control Layers

The physical layer access model is based on frequency hopping spread spectrum access. Each base station is capable of communicating on as many channels as there are modems. Each modem in a FAWN card within a base station is assigned a particular *hopping sequence*. This sequence is randomly generated to minimize the probability of collisions between neighboring base stations. A modem within a base station and the modem in its corresponding mobile, hop according to an assigned sequence.

### 5.2 Connection Handoff

At the lowest level, the event of handing off a mobile from one base station to another is under mobile initiation and control. It is a function of the power level a mobile perceives in its current channel, and thus every mobile periodically samples the received signal strength value that is provided by its RF modem [9]. In one of the schemes we are currently exploring, two received signal strength thresholds, *rsst1* and *rsst2* (*rsst1* < rsst2), are of interest here. The mobile initiates a handover whenever the signal strength on its current channel falls below *rsst1*. At the earliest possible time after this instant, the mobile begins a rapid scan of the entire set of available channels, until it finds one where the received signal strength is *rsst2* or more. Once such a channel has been found, the mobile attempts to synchronize itself with the ongoing communication on the new channel, sends out a request to register a message on that channel, and waits for a positive acknowledgement. There could be several reasons for such an acknowledgement not to be received - the observed signal might have been from a nearby mobile and not a base station, the base station that was heard may not be willing to admit more mobiles, the base station and its set of mobiles might have since switched to the next channel in their hopping pattern etc. To allow for all such possibilities, the mobile will continue its scan for a suitable channel if an acknowledgement has not been received within a certain timeout interval. Once a mobile is registered, and its identity has been made known to the new base station, subsequent signaling messages take care of rerouting communication paths from the old base station to the new. These are handled in higher layers, and are discussed elsewhere in this paper.

### 5.3 Mobility Management

The design of the signaling software assumes the existence of separate signaling protocols for the wired and the wireless network respectively, with base stations acting as gateways between the two logically distinct networks [6]. This allows us to build upon the signaling infrastructure used in the wired network while allowing the signaling in the wireless network to be customized for the unique requirements of that environment. The signaling protocol in the wired network is used to establish, tear down and rebuild connections. The signaling protocol in the wireless network is used to extend routes, remove route loops and trigger route rebuilds, in addition to the establishment, teardown and rebuilding of connections.

We have proposed modifications to conventional signaling in wired networks to support the provision of end to end connectivity in the presence of mobility. Our protocols are designed assuming a radio-based nanocellular access network with mobiles moving frequently between cells. The service quality seen by a VC is determined by the delay, loss and throughput characteristics of the network path over which it is routed. These characteristics are governed by two phases: the transient disruption experienced by a connection when a handoff occurs and the steady state behavior when a mobile is not moving across cell boundaries. The goal of the signaling software is to maintain end to end connectivity while minimizing the impact of mobility on application level quality of service and also ensuring that network resources are used efficiently.

# 6. APPLICATIONS

The testbed allows us to prototype a variety of mobile communications services. Broadly, we are interested in two classes of services. Services based on mobile end systems consisting of a standard laptop or palm top PC connected to the FAWN adapter for wireless access. And, services based on the Personal Mobile Terminal. We are also exploring services that exploit the ephemeral networking capability of SWAN.

Regardless of what the mobile end system is, two application level communication modes are of interest. The first mode is interactive, peer-to-peer, audio-visual communication between two mobile hosts (users). The second involves interaction between mobile hosts (users) and a fixed network server. In the latter context, the domain of the network server could be local (and indoors, such as in an office or a shopping mall [10]), or global (and outdoors, such as on a university campus.) Furthermore, the services can range from real-time, interactive, multimedia communication, to deferred messaging (e.g. traditional e-mail) and low bit rate paging systems. Our aim in studying these applications is two fold. Firstly, to determine how the various classes and modes of services impact the underlying hardware and software architecture of the system. Secondly, to design a common Application Program Interface (API) and support facilities for a wide range of mobile networked computing applications. In this section we briefly outline three applications that exploit the capabilities and features present in FAWN, PMT and SWAN.

#### 6.1 Location Based Services

Mobility brings about the scenario where the information or service the mobile user needs depends on the current location of the user. However, such a service should be available transparently, without the users having to consciously take into account their changing location. Location based services attempt to provide such transparency where the nature of the service or data provided may differ from one location to another. While such applications are most commonly found in wide area networks, they also arise in small in-building environments. An example of such a location based service in an indoor environment might be an user, carrying a FAWN assisted PC laptop, who needs to print a document when located in a new or unfamiliar part of the building. In this case, the local base station with its knowledge of the state of all the nearby printers, can choose one for the user, print the document, and also direct the user to the location of the printer.

Location based services may be provided by a centralized server in the network that responds to different users differently, depending on their current location. However, the key to efficiency is to base the service at the local base station or switch. Thus, the service is replicated at every "local application server", which are server processors attached to each base station (or a switch connected to a collection of base stations.) Each local application server needs to handle only information and requests relevant to local data. The advantage, of course, is that network traffic and latencies are reduced in accessing or updating this local information.

### 6.2 Two Way Multimedia Messaging

The existence of both audio and graphics interfaces in the PMT makes it an ideal PDA device for supporting two way multimedia messaging service. This service combines the best attributes of paging and wireless telephony. Voice, text and images can be exchanged between two PMTs (users), between a PMT and a server, or from a server to one or more PMTs. The information exchange can be interactive and real time, such as in a voice call. Or it can be deferred but real time such as in audio/video mail. As part of the deferred service we have an "alert" mechanism that notifies the user of a pending message or an automatic action taken based on: the time of the day, occurrence of a specified event/trigger, or the geographical location of the PMT (user). Messages are composite objects consisting of voice, text, images and video. The media used for a reply does not necessarily have to be the same as the media in which a message is received. Thus an e-mail received in text form may be responded with a voice mail, and a text message attached to an image may be sent in response to a voice query. As in traditional e-mail systems, deferred messages can be point to point, multicast or broadcast.



Figure 7. Example of an Ephemeral Network.

#### 6.3 Small Domain Ephemeral Networks

SWAN provides the infrastructure to instantly network two or more mobile end systems co-located within the radio frequency range of each other without the aid of an intervening base station. This capability is required in a setting in which a group of users, each with a FAWN assisted laptop, meet in an outdoor area, or in an area within the building where a base station is not installed. If the attendees in that meeting want to exchange files, or jointly edit a document, then SWAN provides the requisite networking capability to support such interaction. The system sets up the communications channels for the duration of the session and ensures that the connectivity is maintained even if the mobile moves around as long as it stays within a prescribed region. Once the session is over, all knowledge of that network is deleted. Figure 7 illustrates such a scenario. Users with mobiles A, B and C are engaged in a meeting. User with mobile A is acting as the scribe and keeps a record of the meeting notes. Let us assume that mobiles B and C are turned off. At the end of the meeting, mobiles B and C are turned on and all three users request SWAN to network them. User A, the scribe, can then request his meeting notes file to be copied to both mobiles B and C. The meeting is adjourned and the network disappears. Clearly, capabilities such as sharing of documents, joint editing and file transfers would be useful. An interesting open question is to determine what other capabilities would be useful in the context of small domain ephemeral networks.

## 7. CONCLUSION

The growing demand for untethered access within home, office and outdoors has provided boundless opportunities for creating new systems and services. Its potential in the mobile networked computing domain is only beginning to be realized. We have described the architecture of an experimental system, SWAN, that we are building to fully understand the system aspects and limitations of providing ubiquitous and tetherless computing environments. The flexibility and programmability of SWAN allows us to administer and control the distributed system at all levels. The hardware for the FAWN adapter and the Personal Mobile Terminal is complete and the design, development and testing of the ETHERWARE software is in progress. Our initial application focus is on providing untethered access to NEMESIS [11], a multimedia information service, through mobile hosts in an indoor office environment.

### REFERENCES

- T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing in Data Management," CACM, pp. 18-29, Oct 1994.
- [2] M. Weiser. "Some computer science issues in ubiquitous computing." Communications of the ACM, 36(7), pp. 209-220, July 1993.
- [3] B. Barringer, T. Burd, et. al. "Infopad: A system design for portable multimedia access." In Wireless 1994, Calgary, Canada, July 1994.
- [4] 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, pp. 83-102, Cambridge, Massachusetts, August 1993.
- [5] Bruce Tuch, "Development of WaveLAN(tm), an ISM Band Wireless LAN," AT&T Technical Journal, pp. 27-37, July/august 1993.
- [6] R. Caceres and L. Iftode. "The effects of mobility on reliable transport protocols." In International Conference on Distributed Computing System, Cracow, Poland, June 1994.
- [7] Mary G. Baker, "Changing Communication Environments in MosquitoNet," Proc. 1994 Workshop on Mobile Computing Systems and Applications,

Santa Cruz, CA., Dec. 8-9, 1994.

- [8] D. Comer and V. Russo, "Using ATM for a Campus-Scale Wireless Internet," Proc. 1994 Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA., Dec. 8-9, 1994.
- [9] P. Agrawal, D. K. Anvekar and B. Narendran, "Optimal Prioritization of Handovers in Mobile Cellular Networks," Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'94), pp. 1393-1398, The Hague, Netherlands, Sept. 20-23, 1994.
- [10] A. Asthana, Mark Cravatts and P. Krzyzanowski, "An Indoor Wireless System for Personalized Shopping Assistance," Proceedings IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, Dec. 8-9, 1994.
- [11] H. P. Katseff and B. S. Robinson, "Predictive Prefetch in the Nemesis Multimedia Information Service," ACM Multimedia 1994, San Francisco, Oct. 15, 1994.