SWAN: A Mobile Multimedia Wireless Network

Prathima Agrawal
Eoin Hyden
Paul Krzyzanowski
Partho Mishra
Mani B. Srivastava[‡]
John A. Trotter

AT&T Bell Laboratories
Room 2C-205, 600 Mountain Avenue
Murray Hill, NJ 07974
Email: mbs@research.att.com
Phone: (908) 582-7766 / Fax: (908) 582-5192

‡ Please address all communications to: Mani B. Srivastava

SWAN: A Mobile Multimedia Wireless Network

Abstract

SWAN (Seamless Wireless ATM Network) is an experimental indoor wireless network that investigates the combination of wireless access with multimedia networked computing in an indoor setting. It is based on room-sized pico-cells and mobile multimedia end-points. It enables users carrying multimedia end-points such as PDAs, laptops, and portable multimedia terminals, to seamlessly roam while accessing multimedia data resident in a backbone wired network. The network model of SWAN consists of basestations connected by a wired ATM backbone network, and wireless ATM last-hops to the mobile hosts. SWAN is one of the first systems to realize the concept of a wireless and mobile ATM network. Mobile hosts as well as basestations are embedded with custom designed ATM adapter cards called FAWN (Flexible Adapter for Wireless Networking). FAWN uses off-the-shelf 2.4 GHz ISM band radios.

After giving an overview of the SWAN network model, and discussing the challenges in making ATM wireless and mobile, the paper describes the first phase implementation of SWAN hardware and software. This initial implementation provides connectivity over the wireless last hop. We have investigated both native mode end-to-end ATM communication across the wired ATM backbone and wireless ATM links, and TCP and UDP communication using IP-over-wireless-ATM in the wireless link with IP forwarding and segmentation-reassembly modules at the basestations. The latter mode of communication has allowed experimentation with various readily available TCP and UDP based multimedia applications such as *nv*, *vat* etc. Typical TCP throughput is 227 Kb/s when the raw channel bandwidth being equally divided into 312 Kb/s each in the transmit and receive directions, while round-trip delays over the wireless hop range from 5 ms to 25 ms. Performance metrics such as throughput, delay and delay jitter are affected by various wireless link attributes. The paper presents experimental data that explores the relationship between performance and wireless link attributes such as the frame size used to transfer data over the air, and the number of mobiles that are sharing the link.

SWAN: A Mobile Multimedia Wireless Network

1.0 Introduction

1.1 Towards Integrated Service Wireless Networks

"Anytime anywhere" information access and processing are much cherished in modern society because of their ability to bring flexibility, freedom, and increased efficiency to individuals and organizations. Businesses can use ubiquitous information access to streamline workflow by allowing employees to access and update databases, and receive work orders, directly at their logical work site or point of service. Examples include access of patient charts and medical data by doctors and nurses from handheld terminals, and electronic delivery of trade orders directly to traders carrying PDAs on a busy stock exchange floor.

Wireless access technology, by providing ubiquitous and tetherless network connectivity to mobile users, has to some extent already realized the vision of anytime anywhere access in two specific domains: cellular voice telephony and wireless data networks. Cellular telephone networks have extended the domain of circuit-switched telephone service over a wireless last hop, while wireless data networks such as WaveLAN [Tuch93] for local area, Metricom's Ricochet for metropolitan area, and Cellular Digital Packet Data (CDPD) from various cellular carriers for wide area do the same for users of TCP/IP data packet networks.

The degree to which wireless access has succeeded in the voice and data domains has clearly been different. Despite many technical foibles, cellular telephone networks have been a market success, enjoying market growth rates between 35 to 60 percent per year for the past decade and a subscriber base of many tens of millions in the U.S. alone [Cox95]. The now mature analog cellular network technology is evolving into digital cellular networks, low-earth orbit satellite based cellular networks, personal basestations, and wireless PBXs. Wireless data networks, on the other hand, are relatively immature with a much smaller market penetration. A multitude of factors are responsible for this, ranging from low data rates and high cost in outdoor data networks, to the rather chaotic standards situation and consequent user frustration that exist in higher speed wireless local area networks. While the users of wireless data networks are not quite as numerous as that for wireless voice networks, their number is expected to triple from 1995 to 1997 as wireless data networks become easier and cheaper to use, and more powerful and less expensive mobile devices arrive [Shaffer95].

The state of wireless access can thus be summed as a choice between cellular telephone networks that provide voice bit rates with rigid service quality, and wireless data networks that look like wired data networks of yesteryears with limited data rates and no notion of service quality. In recent years wired networks have begun to evolve towards integrated service data networks with a multimedia oriented network model. This is due to the availability of user devices capable of multimedia communication (such as multimedia PCs and workstations), and the need for service providers to flexibly and efficiently multiplex multimedia connections in their networks.

The goal of SWAN mobile computing environment, being developed in the Networked Computing Research Department at Bell Laboratories, is to drive wireless networks towards a similar integrated service model. The dream is to enable mobile context-aware multimedia services to users carrying heterogeneous portable end-points with varying degrees of smartness. For example, in indoor settings, which is our initial focus, we want a multimedia networked computing and communication system to subsume the functionality and calling models of wireless LANs and PBXs. When deployed in a campus or an office, this integrated network will support audio/video calls on wireless communicators (with services such as call forwarding and conferencing); multimedia messaging and paging; multimedia applications on wireless PDAs, laptops, and terminals; and, access to movable wireless gadgets such as video monitoring cameras.

1.2 The SWAN Approach

Technically, realizing the SWAN vision is not just a bandwidth or system capacity issue. Equally important are controllable bandwidth and a quality of service (QoS) environment for multimedia in the presence of wireless and mobility. Clearly, technology trends are encouraging. Continual progress in wireless technology will soon make it possible to economically provide per user data rates of several Mbps, at least inside buildings. Already the readily available radio modems operating in the 900 MHz and 2.4 GHz industrial, scientific, and medical (ISM) bands, and in the newly allocated 1.91-1.93 GHz data PCS band offer 1-2 Mbps per channel of data rates. Higher speed radios, such as those operating in the 5.8 GHz will be available in not too distant a future. When operated in a pico-cellular configuration with spatial channel multiplexing and reuse, such multichannel radios can support a reasonable user density. Such wireless networks, together with the emerging integrated service wired network infrastructure and progress in audio/video compression algorithms, will permit seamless delivery of packetized multimedia information to a mobile user, at least indoors. Progress in packaging, display, and low-power circuits [Chandrakasan95] has resulted in portable multimedia end-points [Asthana94, Barringer94] that seamlessly integrate into a user's networked computing environment.

SWAN, as its expanded name *Seamless Wireless ATM Network* suggests, seeks to provide continual network connection to mobile heterogeneous ATM end-points. We are building a prototype system by exploiting the synergy between the above mentioned technology trends in multimedia information access, wireless access technology, and portable multimedia end-points. The available off-the-shelf radio technology is however somewhat premature for our needs. Besides obvious bandwidth limitations for large user densities, the typical ISM band radio cards often have limitations such as few if any software tunable channels, and fixed high RF power levels which are more suitable for building-wide wireless LANs than for room-sized pico-cells. Realizing such limitations of the available off-the-shelf radio technology, we have attempted to keep our system design largely independent of any specific radio.

The network model of SWAN consists of basestations connected by a wired ATM backbone network, and wireless ATM last-hops to the mobile ATM hosts. The system and network software in SWAN, called Etherware, uses novel connection management protocols developed by us to handle end-point mobility and schedule wireless resources. By allowing applications to register interest, in the form of call-back handlers, to low level events such as an impending hand-off or a change in wireless channel bit error rate, Etherware provides a platform for a variety of mobile context-aware multimedia applications (i.e., applications that adapt to changes in their operating environment).

The mobile hosts as well as the basestations incorporate custom designed adapter cards called FAWN (Flexible Adapter for Wireless Networking) [Trotter95]. Currently, an off-the-shelf 2.4 GHz ISM band radio is interfaced to FAWN. FAWN provides embedded programmable processor and reconfigurable hardware resources for implementing medium-access control (MAC) and air-interface subsystems to transport the cells[1] in ATM virtual circuits over the air. Software programmability and miniaturized physical size enable the development of heterogeneous portable end-points with varying degrees of smartness.

Enhancing ATM to make it wireless and mobile is a key feature of SWAN, which is among the first systems to implement this concept. SWAN's choice of ATM not only produces a homogeneous end-to-end network simplifying the architecture, but also has attributes such as QoS specifiable virtual circuits (VCs) which are quite useful in a wireless and mobile environment. ATM allows cell scheduling, error control, and hand-off management based on QoS parameters instead of treating all the traffic uniformly as current wireless data LANs do. Compared to research projects such as Berkeley's Infopad and UCLA's WAMIS, which too seek to provide a mobile user with audio, video, graphics, and data connectivity, a distinctive aspect of SWAN is indeed its use of ATM. For example, Infopad uses a TCP/IP backbone with a separate custom connection-oriented wireless hop transport mechanism between "dumb" terminals and basestations. WAMIS, which has adopted a packet-radio type ad hoc networking model with end-points also acting as relays, supports TCP/IP and virtual circuits.

Besides our and other [Condon95, Eng95] efforts within Bell Labs, NEC's WATMNet [French95] and ORL Cambridge's wireless ATM system [Porter95] are the two other concurrent wireless research systems that are implementing wireless and mobile ATM, though with different approaches and scope. Any mobile and wireless ATM network obviously requires solutions to problems such as appropriate hardware for transmitting and receiving ATM cells over the air, and VC connection and service quality management in the presence of mobility. SWAN's solutions to such lower level problems have been presented elsewhere [Mishra94, Trotter95, Srivastava96]. Compared to other wireless and mobile ATM efforts, the SWAN system is distinguished by novel low latency VC rerouting algorithms based on performance triggered rebuilds [Mishra94], custom reconfigurable and miniature wireless ATM adapter hardware [Trotter95], and support for heterogeneous end-systems ranging from laptops to dumb multimedia terminals [Asthana94].

In this paper we describe the overall system architecture of SWAN, its first phase implementation, and initial performance data. We present SWAN's ATM based network model in Section 2, and discuss the rational and challenges of wireless and mobile ATM in Section 3, Next, the first phase implementation of SWAN is described in Sections 4 and 5. This implementation provides connectivity over the wireless last hop. We have investigated both native mode end-to-end ATM communication across the wired ATM backbone and wireless ATM links, and TCP and UDP communication using IP-over-wireless-ATM in the wireless link with IP forwarding and segmentation-reassembly modules at the basestations. The latter mode has allowed experimentation with various readily available TCP and UDP based multimedia applications. Our experience and performance data on the initial system implementation are presented in Sections 6 and 7.

2.0 Network Communication Architecture of SWAN

Figure 1 shows a high level view of the network communication model adopted by SWAN [Agrawal95]. A hierarchy of wide-area and local-area wired ATM networks is used as the back-bone network. Wireless access is used in the last hop to mobile hosts. In addition to connecting conventional wired server hosts and client end-points, the wired backbone also connects to special switching nodes called basestations. The basestations are

1. The paper uses the word *cell* in some instances to refer to an ATM cell (unit of data), and in other instances to refer to the geographical area surrounding basestation in a cellular network. Both are well accepted standard terms, but the intended usage will usually be clear from the context.

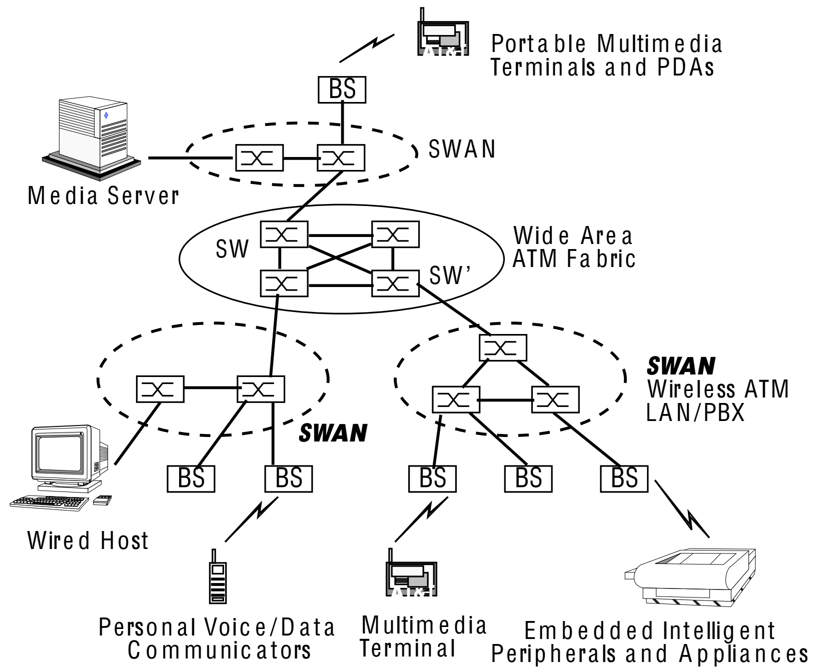
Portable Multimedia Terminals and PDAs
BS
SWAN
Media Server
SW
Wide Area ATM Fabric
SW'
SWAN
Wireless ATM LAN/PBX
SWAN
BS
BS
BS
BS
BS
Wired Host
Personal Voice/Data Communicators
Multimedia Terminal
Embedded Intelligent Peripherals and Appliances

Figure 1: Network Communication Model of the SWAN Wireless ATM Network

conventional PCs and workstations, and are equipped with the FAWN [Trotter95] custom wireless adapter cards. These act as gateways for communication between nearby mobile hosts and the wired network. Functionally, the basestations are special mobility-aware ATM switches that are located at the edge of the wired ATM cloud and have wireless link interfaces on one or more of the switch ports. In practice, given the slow speed of wireless links, the switching functionality may be realized in software instead of ATM switching hardware. The mobile hosts are also equipped with the FAWN adapter. The geographical area for which a basestation acts as the gateway is a room-sized pico-cell. Network connectivity is continually maintained as users carrying a variety of mobile hosts roam from one cell to another. Although mobile hosts in SWAN may have different local general purpose computing resources, all of them must have the ability to participate in network signaling and data transfer protocols. A mobile in SWAN sends and receives all its traffic through the basestation in its current cell.

Endpoints in SWAN range considerably in functionality and mobility. They include “smart” PDAs and laptops, “dumb” multimedia terminals, and wireless entities such as print servers and cameras that move infrequently. While dumb end-points are not expected to compute locally, even the smart end-points that perform considerable local computation are likely to off-load bulk of the processing to servers on the wired network. This is necessary to conserve available power and enhance battery life. Thus the processing at the smart end-points will be dominated by communication intensive tasks such as information filtering, compression and decompression, encryption and decryption, and capture and presentation.

Our model uses end-to-end ATM, over both the wired network and the wireless last hops. This is in contrast to the use of connectionless mobile-IP [Ioannidis91, Teraoka91] with present day wireless data LANs such as WaveLAN [Tuch93]. The choice of ATM was motivated by advances in video compression algorithms and availability of higher bandwidth RF transceivers which permit the transmission of packetized video to a mobile. Support for multimedia traffic over the wireless segment was a driving force in SWAN. Adopting the connection-oriented paradigm of ATM virtual circuits over the wireless hop allows quality of service guarantees associated with virtual circuits carrying multimedia traffic to be extended end-to-end. Other researchers, such as Raychaudhuri and Wilson [Raychaudhuri94], have also advocated the use of ATM based transport in multiservices wireless personal communication networks. Rajagopalan [Rajagopalan95] has proposed an architectural option for integrating wireless access and mobility into existing ATM networks by using special mobility-aware anchor nodes.

Extending ATM's virtual circuit paradigm all the way through to a mobile host necessitates continuous rerouting of ATM virtual circuits as a mobile host moves [Keeton93, Acampora94, Biswas94, Toh95]. Indeed, Toh's measurements at Cambridge [Toh95] show that VC rerouting can be quite fast. The small cell sizes and the presence of quality of service sensitive multimedia traffic make this problem particularly important in SWAN. Virtual circuits carrying audio or video, as far as possible, need to be immune from disruptions as a mobile host *hands-off* from one basestation to a neighboring one. ATM signaling protocols need to accomplish the task of virtual circuit rerouting with minimum latency. At a lower level, the medium access control (MAC) and air-interface layers must support efficient transport of ATM cells, low latency mobile hand-offs, per-VC error control, and cell scheduling according to VC QoS parameters. Chandler et. al. [Chandler94] have described an air-interface for ATM cell transport in a CDMA wireless network. SWAN's solutions to these connection management, signalling protocol, MAC, and air-interface problems can be found in [Mishra94, Srivastava96], and form the algorithmic basis of our system implementation.

3.0 Rationale and Challenges of Mobile and Wireless ATM

Before attempting to answer SWAN's approach to mobile and wireless ATM, one needs to answer what is the rationale of using mobile and wireless ATM, and what are the challenges in implementing it. These questions are interesting because of at least two reasons. First, ATM was designed for an environment where the hosts do not move and are connected to a switch via a relatively error free and high speed point-to-point wired link. It is not obvious what enhancements are needed for ATM to work well in a mobile and wireless setting. Second, there is the inevitable comparison with using IP, instead of using ATM, in a mobile and wireless environment. Advocating mobile and wireless ATM, particularly when existing products such as wireless LANs and CDPD are IP based, naturally makes one a participant in the on-going serious and somewhat religious ATM versus IP debate [Crowcroft95] in the data networking community.

3.1 Rationale

Before delving deep into the rationale question, it must be pointed out that there is a homogeneity argument in favor of wireless and mobile ATM as envisioned in SWAN. ATM is a scalable technology, and appears likely to be at the core of the future multimedia networks. Therefore, extending the QoS specifiable ATM virtual circuit model over the wireless hop leads to a homogeneous end-to-end network simplifying the architecture. In the wired world, extending ATM to the desktop has faced hurdles such as an entrenched and cheap last hop access technology (ethernet) and higher cost of fiber and optical transceivers that were required until the recent emergence of twisted-pair ATM. There are no such entrenched standards or obvious cost disadvantages in using ATM for wireless last hop access.

The rationale for using wireless and mobile ATM can be explored along two dimensions that distinguish ATM: "cellification" and "QoS-specifiable VCs". By "cellification" we refer to the fact that ATM uses fixed and small sized cells (48 byte body with 5 byte header). By "QoS-specifiable VCs" we refer to the use of switched virtual circuits whose service quality parameters are negotiated at set-up time by the end-points with the network, and the network goes through a process of admission control and resource allocation before the connection is set up.

Superficially, “cellification” appears to be a complete disaster for wireless ATM with a terrible waste of bandwidth for the last hop due to a large ATM cell header relative to the cell body. The reality, however, is more subtle. First, the fine-grained multiplexing as provided by ATM cells is well suited to slow speed links because it leads to lower delay jitter and queuing delays. Second, wireless links have high bit error rates. Therefore, one cannot use very large packets because that would lead to unacceptably large packet loss probability and not work well with retransmission based error control. Third, radios also impose limits on maximum continuous transmit burst. For example, this limit with the radio currently used in SWAN (Section 4.3.1) is 10 ms @ 625 Kbps, or 6250 bits = 781 bytes. This poses a natural size limit on the data transmission unit. If one were to use IP transport, with an MTU (Maximum Transmit Unit) of 781 bytes, the header overhead with IP datagrams would in the best case be 20/781 = 2.6%. By comparison, the overhead in the case of ATM is 9.4%. In reality, the two would be closer once overheads such as MAC level header, synchronization bits etc. are taken into account. Fourth, one can be smart with ATM cell headers. By limiting the number of active VCs to a mobile (something which many wired ATM adapters also do), one can use smaller number of bits to represent virtual circuits for the duration of the wireless hop. For example, the 24-bit VCI/VPI can be transparently replaced by an 8- or 16-bit id over the wireless link between a basestation and a mobile. A second technique that may used is to cluster multiple ATM cells on the same VC into a variable sized MAC frame, and share the VC id fields. In essence, in these two techniques the basestation acts as a gateway that can change cell format for the duration of the wireless link. Aside from the bandwidth waste argument, another “cellification” related argument against ATM in the wired world is that the small cell sizes lead to cell processing constraints due to the limited time available per cell in a high speed fibre optic network. Clearly, this is not an issue with wireless networks. In short, it suffices to say that the “cellification” issue over the wireless last hop is quite complex with many trade-offs.

The rationale for wireless and mobile ATM is quite strong from the “QoS-specifiable VC” perspective. The ATM VCs provide a useful traffic flow id for use at the wireless link level. For example, the MAC layer can use the VC ids to meaningfully allocate and schedule the shared wireless channel resources. MAC layers in current IP or IPX based wireless data LANs cannot do such intelligent wireless resource allocation. Similarly, the link level error control mechanism can be suitably adapted on a per VC basis depending on the characteristic of the individual VC. Further, ATM’s notion of specifying per-VC quality of service (QoS) is quite useful in a wireless and mobile network carrying multimedia traffic. QoS specification can be used, for example, to give some connections higher priority during hand-offs, or to use rerouting policies tailored to the traffic characteristics. In general, the VCs with QoS parameters in wireless and mobile ATM provide the ability to meaningfully distinguish the data packets being sent over the air, and not treat all of them according to one generic policy. Wireless and mobile ATM suffers neither from the rigid synchronous structure of cellular voice and PCS systems, nor from the inherently best-effort delivery model of wireless data LANs. It must be pointed out that the IP-based internet protocols are also being extended to allow multimedia applications that need service quality guarantees to express those needs to the network via resource reservation protocols such as RSVP [Zhang93]. To make the network honor traffic guarantees, it has been proposed that routes with reservations should be pinned or locked in place. This points to a possible convergence of the IP and the ATM service models. Research problems with QoS-specified pinned IP routes in a wireless and mobile network will be similar to those with QoS-specifiable ATM VCs.

3.2 Challenges

ATM was designed for an environment where the hosts do not move, and the transmission medium is a relatively error free and high speed point-to-point wired link. However, in a SWAN like environment, the hosts move and the transmission medium is a relatively noise and burst error prone shared medium of modest bandwidth. The change from a wired static world to a wireless mobile world can have unexpected pitfalls as the experience of various researchers [Caceres94] with TCP performance in a mobile-IP wireless environment has shown - packet loss in the wireless hops due to noise, burst error, or hand-off is falsely interpreted by TCP as being due to congestion, leading to poor performance. Clearly, it is not obvious what enhancements are needed for ATM to work well in a mobile and wireless setting.

The issues that need to be successfully addressed in making ATM work in a SWAN-like network fall into two somewhat orthogonal categories: mobility related problems at the higher level, and wireless related problems at the lower level. From a mobility perspective, the key ATM issue is that of virtual circuit management in the presence of host mobility. Obviously, the VC route needs to be continually modified as the hosts at either end move during the lifetime of a connection. The rerouting must be done fast enough so as to cause minimal disruption to the transport layer. The signalling protocols for VC establishment, rerouting, and tear-down must function properly even in the presence of host mobility at both ends during the signalling phase itself. Any rerouting must maintain the sequence of ATM cells at the end-points. Clearly, there would appear to be “obvious” solutions to some of these problems from analogous problems in the connection-oriented cellular and PCS world. However, the scale of the problem here is different in many dimensions, rendering the naive solutions too inefficient. The hand-off frequency is much larger due to small cell size - imagine a person strolling down a hallway. Each terminal can terminate a large number of VCs, leading to a large number of VC reroutes on each hand-off. The statistical multiplexing inherent in ATM and multimedia traffic requires a QoS renegotiation at the new basestation, compared to the relatively simple frequency or time slot allocation in cellular and PCS basestations.

Although wireless is a last hop issue, end-to-end ATM requires various components of the last hop to be aware of ATM traffic. From a wireless perspective, the key ATM issues are providing lower layer support for ATM QoS, and efficient transport of ATM over a slow noisy air medium. The first issue, QoS support, arises from the need for the MAC subsystem to participate in admission control at the time of VC admission, and renegotiation at the time of hand-off. Usually, MAC protocols are targeted at coordinating access of the shared air resource among multiple mobiles. With ATM, the MAC protocols need to schedule the air resource among multiple VCs at multiple mobiles. The second issue, efficient transport of ATM over the air, arises from the high header-to-payload ratio of ATM cells in an already low bandwidth air medium, and the high wireless link noise and burst error rate. To address the former, one can use header compression and cell clustering techniques mentioned in Section 3.1. To address the latter, link level error control schemes are needed. Here, ATM VCs offer the advantage that link level error control schemes can be selected appropriately for each VC.

Finally, besides the specific mobility and wireless issues, there is a third related problem of what services API should a mobile and wireless ATM network offer to native mode ATM applications. Current ATM APIs are tailored for a static wired environment, and only offer basic services for VC establishment, tear-down, and data read/write. For a SWAN-like network, the API ought to be able to hide wireless and mobility from applications that wish to operate transparently, and to send wireless and mobility related events to applications that wish to adapt to changing network conditions. For the latter class of mobility and wireless adaptive applications, the lower layers of the network, such as MAC, must allow the higher layers to register interest in specific low level events, such as an imminent soft-handoff, and be able to send events back to the higher layers and the application when the specified low level event occurs.

Rest of the paper describes the architecture and implementation of SWAN, and how it addressed some of the above mentioned challenges in making ATM wireless and mobile.

4.0 Mobility Management and Wireless Access Architecture

An initial system design of SWAN is complete and functional. Its focus is on the local-area domain, and the wireless last hop. The wireless last hop consists of basestations and mobiles. SWAN uses room-sized cells, each of which has a basestation equipped with one or more radio ports. Generic PCs and Sun workstations are used as basestations by plugging in a wired ATM adapter card and one or more FAWN RF wireless ATM adapter cards, and running necessary ATM switching and signalling software. The use of PCs and workstations also allows the basestations to host application and system processes if need be. The mobiles, at the other end of the wireless hop, include portable computers with an adjunct ATM wireless adapter or multimedia terminals with embedded ATM wireless adapter. The wired ATM backbone is currently composed of Fore's ATM switches. The software part of SWAN is called *Etherware* which consists of software modules running on the basestation CPU, on the mobile host CPU, and on an embedded CPU on the wireless adapter card.

Taking a top-down view, in this section we describe the high level architecture in terms of the protocols and algorithms embodied in Etherware. The first phase hardware and software implementation is described in the next section.

4.1 Layers in Etherware

The key algorithms in Etherware are those used to handle ATM mobility and wireless access. 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 SWAN: mobile hosts, basestations, wired switches, wired hosts. These entities communicate with each other via a "signaling" protocol which handles addressing, location and data forwarding functions. Being directly interfaced to the wireless link, of particular interest are the mobile hosts and the basestations. Figure 2 shows a logical view of the wireless and mobile ATM stack realized by Etherware at these two entities.

At the highest layer in the stack is the ATM Connection Management function which offers a mobile ATM API to the local applications, and embodies state machines that participate in the necessary signalling to establish, maintain, and tear-down VCs with specified QoS parameters in the presence of mobility. The signalling is done not only with the peer ATM Connection Management state machines at other mobile hosts, basestations, wired switches, and wired hosts, but also with the lower layers locally to negotiate resource allocation at VC setup and reroute, configure per-VC entries, and register interests in low level events. The layer below does the necessary functions to transport data between the local applications and the available ATM link interfaces. At the mobile hosts this is simply segmentation-reassembly to the single (wireless) interface. At the basestation, however, an ATM cell switching functionality is also needed to route cells among the multiple wired and wireless ATM link interfaces. Below this data transport and switching layer are functions associated with the wireless link for (i) scheduling of cells in the VCs on the basis of the specified rate, (ii) link level error control using FEC and/or selective retransmission optionally specified on a per-VC basis at setup, (iii) medium access control (MAC) that supports QoS, and (iv) air-interface for efficient transport over the air.

4.2 ATM Mobility Management

The design of the signaling software assumes the existence of separate signaling protocols for the wired and the wireless network respectively, with basestations acting as gateways between the two logically distinct networks [Caceres94]. This allows us to minimize the changes required to 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, teardown and rebuild connections. The signaling protocol in the wireless network is used to extend routes, remove route

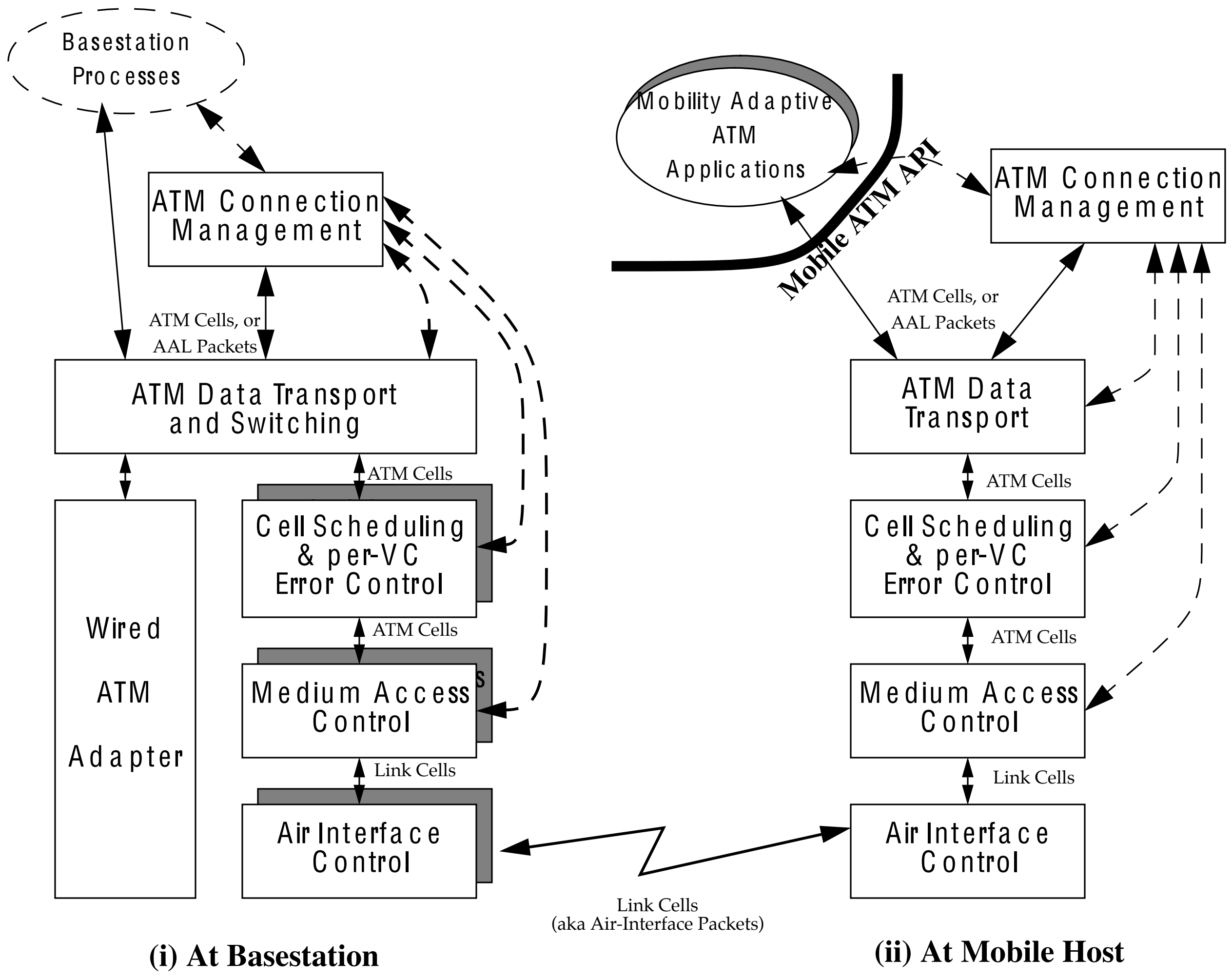
Basestation Processes
ATM Connection Management
ATM Cells, or AAL Packets
ATM Data Transport and Switching
ATM Cells
Cell Scheduling & per-VC Error Control
ATM Cells
Medium Access Control
Link Cells
Air Interface Control
Wired ATM Adapter
Link Cells (aka Air-Interface Packets)
(i) At Basestation
Mobility Adaptive ATM Applications
Mobile ATM API
ATM Connection Management
ATM Cells, or AAL Packets
ATM Data Transport
ATM Cells
Cell Scheduling & per-VC Error Control
ATM Cells
Medium Access Control
Link Cells
Air Interface Control
(ii) At Mobile Host

Figure 2: Wireless and Mobile ATM Stacks

loops and trigger route rebuilds, in addition to the establishment, teardown and rebuilding of connections.

SWAN uses modifications to conventional signaling in wired networks to support the provision of end to end connectivity in the presence of mobility. The protocols are designed assuming small cells, with mobiles moving frequently between cells. The service quality seen by a virtual circuit 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.

The protocols have some novel features. Since most of a mobile's movement will be within a local domain, the protocols perform *path extensions* only following a cell boundary crossing. However, when a domain boundary is crossed, building an extension will cause a more costly network path. This is used as a trigger to rebuild the routing tree. The signaling software allows an application to provide information about its quality of service

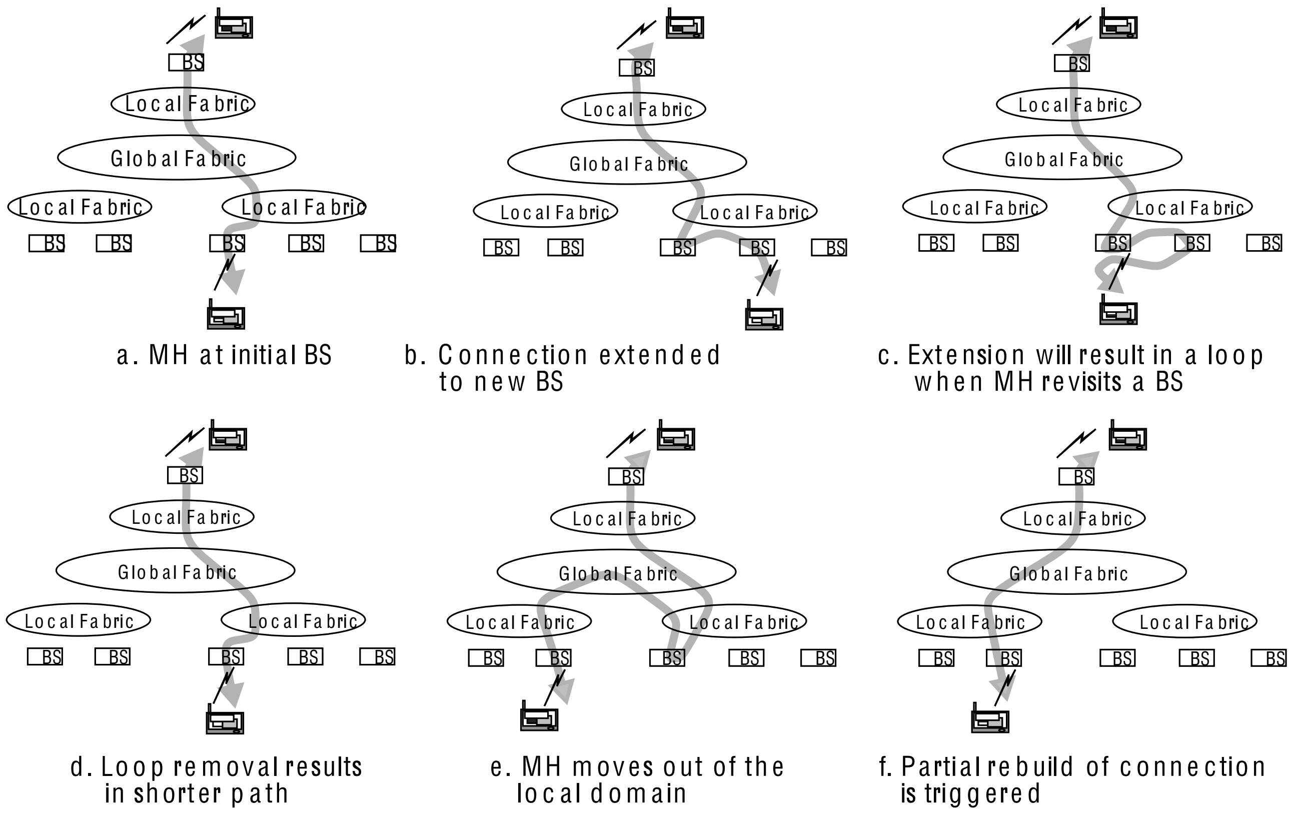
BS
Local Fabric
Global Fabric
Local Fabric
Local Fabric
BS
a. MH at initial BS
b. Connection extended to new BS
c. Extension will result in a loop when MH revisits a BS
d. Loop removal results in shorter path
e. MH moves out of the local domain
f. Partial rebuild of connection is triggered

Figure 3: VC Rerouting Using Extension. Loop Removal, and Rerouting

requirements which are used to guide the actions taken following a handoff. A simple optimization that is used to improve network efficiency during the path extension process is to detect and eliminate loops, i.e. when the same basestation appears twice in the path. While the details of the above protocols for connection establishment and rerouting in the presence of mobility have been presented previously in [Mishra94], Figure 3 summarizes the key aspects of our VC rerouting protocol.

4.3 Medium-Access Control and Air-Interface for Wireless ATM

Since carrying multimedia traffic to the mobiles is a major goal in SWAN, the two important drivers for the medium access control (MAC) and physical layer control subsystem are low latency hand-offs and support for multiple simultaneous channels in a given cell. In addition, explicit allocation of wireless resources among ATM virtual circuits is crucial. Finally, at least initially, simplicity of implementation was considered desirable. Further, as shown in the next section, we have based our wireless ATM adapter implementation on software and reconfigurable hardware so as to make algorithmic enhancements to the MAC and air-interface feasible.

In light of these considerations, the basic lower layer strategy used in the SWAN prototype is to assign each mobile in a cell to its own radio port, or channel, on the basestation. The available channels are distributed among the basestations, with several channels to a cell, and basestations are accordingly equipped with multiple radio ports. The medium access control implementation is thus simplified, having to deal only with the management of multiple channels, and not with the sharing of a channel by multiple mobiles. This is adequate for the use of our prototype in individual offices and small rooms with only a few (2-3) mobile end-points each. More demanding usage patterns, such as handling conference rooms, will indeed require the ability to share a channel among multiple mobiles. This is being addressed in our on-going work. It must be emphasized that the preceding simplified view of the lower layer is only an artifact of the current simplified implementation, and not inherent to the SWAN architecture.

The implementation of the air-interface and medium access control does to some extent depend on the type of radio being used. Therefore, before describing the air-interface and medium access control in the following subsections, we briefly describe the radio transceiver in the next subsection.

4.3.1 Radio Transceiver

The SWAN system is largely independent of the specific radio being used. With SWAN being a pico-cellular architecture, the only architectural requirement is that the radio transceiver provide multiple software controllable communication channels which can be spatially multiplexed and reused. Details such as the band of operation (ISM, PCS etc.), and whether the radio uses direct sequence or frequency hopping spread spectrum are not factors per se in the choice of the radio.

To meet our research needs, SWAN currently uses an off-the-shelf radio card which is a 625 Kbps half-duplex 2.4 GHz ISM band slow frequency hopping transceiver with two power levels and two selectable antennas. FCC requirements dictate that the radio must hop pseudo-randomly among at least 75 of the 83 available 1 Mhz wide frequency slots in the 2.405 to 2.4835 GHz region such that no more than 0.4 seconds are spent in a slot every 30 seconds. The radio incurs an 80 us penalty per hop. Communicating transceivers hop according to a pre-determined pseudo-random hopping sequence.

A *channel* in the current implementation of SWAN's wireless hop naturally corresponds to a hopping sequence, or a specific permutation of 75 to 83 frequency slots. Channels corresponding to two different frequency hopping sequences interfere with each other when they occupy the same frequency slot at the same time. Therefore geographically co-located channels use *weakly orthogonal* hopping sequences such that the chances of two different channels being in the same frequency slot at the same time is minimized. SWAN uses a family of 22 distinct hopping sequences, or channels, distributed among the basestations in various pico-cells. More than one channel can be allocated to a basestation, and a basestation needs to have a separate radio for each channel assigned to it. The same channel cannot be assigned to two basestations in cells that can mutually interfere. The mobiles have only one radio, and at any given time operate in a specific channel.

The radio limits transmission burst to 10 ms. At 625 Kbps, the maximum size of an air interface packet is 6250 bits. The radio provides a rated bit error rate of 1E-5. This translates into an ATM cell loss probability due to noise of less than 0.5%. While being much larger than what is easily available on the wired backbone, this cell loss probability is overshadowed by frequency slot collision in co-located channels. For example, if two channels using 75 long frequency hopping sequences collide even once every sequence, a loss of 4% occurs. This illustrates a fundamental capacity loss problem when frequency hopping spread spectrum radios are used in a system with multiple co-located channels. Besides direct frequency slot collisions, one also has the problem of interference from adjacent frequency slots, and the adverse impact on frequency slot reuse of the high level modulation that is needed to get increased bit rates. In general, techniques such as smart hopping and information spreading across hops are more crucial in frequency hopping based wireless links than techniques targeted at noise.

4.3.2 Mapping of ATM Cells to the Air-Interface Packets

Air-Interface Packets, or *Link Cells*, are the data units sent by the medium access control layer to the air interface control layer at the transmit end, and received by the medium access control layer from the air interface control layer at the receive end. The air interface control layer directly controls the radio, and would

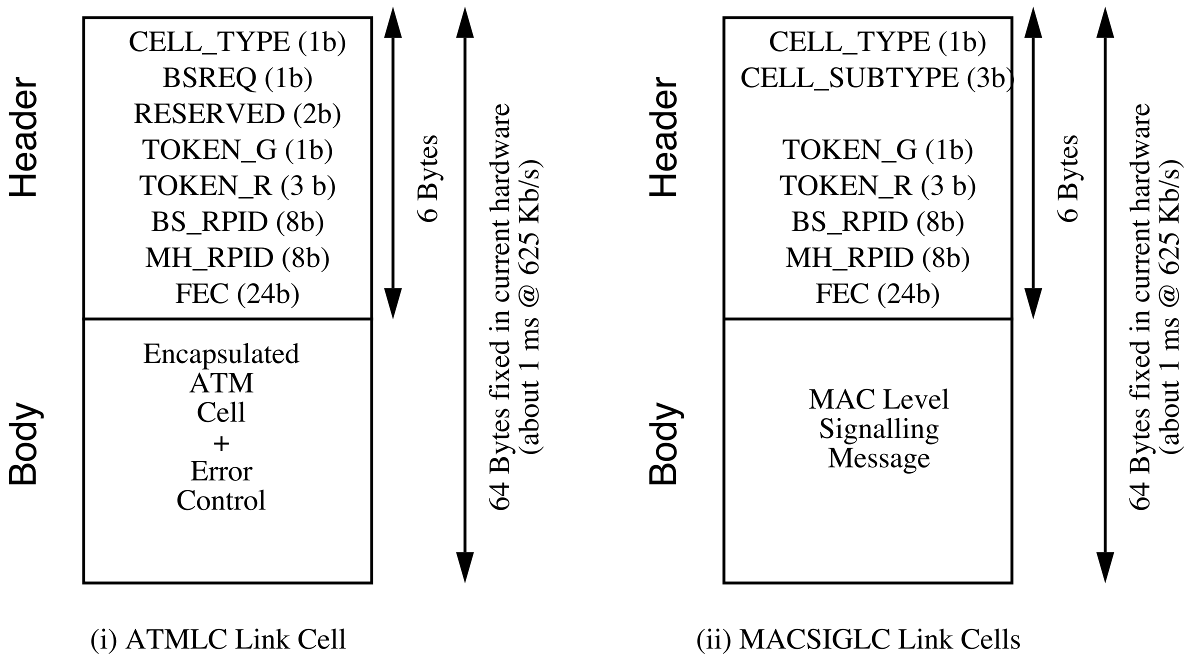

Figure 4: Format of Link Cells (Air Interface Packets)

typically be realized in hardware as an air-interface controller (or, physical layer controller). Digital radio cores in their simplest embeddable form offer serial bit stream input and output, and the air-interface controller is expected to do parallel to serial conversion and channel coding at the transmitter, and clock recovery, bit framing, channel decoding, and serial to parallel conversion at the receiver. To take advantage of the readily available off-the-shelf serial communications controller chips that do these functions, the air-interface controller in SWAN incorporates a serial communications controller operating in a synchronous mode, resulting in the well known SDLC (synchronous data link control) protocol being used over the air. In one transmit burst, transmitters send one or more SDLC frames separated by the SDLC SYNC bytes, with each SDLC frame consisting of one or more Link Cells.

There are several types of Link Cells. One type, called ATMLC, is defined to carry an encapsulated ATM cell. Several other Link Cell types are defined for MAC level signalling. For example, the simple MAC protocol currently being used defines the following signalling Link Cells: CRLC for connection request by a mobile that powers up, HRLC for hand-off request by a mobile, SYNCLC for idle channel, and CHRCLACK1, CHRLACK2, and CHRLCACK3 for handshake during registration of a mobile at a basestation. Collectively the Link Cells for MAC level signalling are referred to as MACSIGLCs. All Link Cells are composed of a fixed 6 byte header, and a body whose contents depend on the type of the Link Cell. As shown in Figure 4, the header has an 8-bit statically assigned basestation radio port id (BS_RPID) field, an 8-bit dynamically assigned mobile host radio port id (MH_RPID) field, a 1-bit CELL_TYPE field indicating whether the Link Cell is of type ATMLC or not, 7 bits defined by the MAC protocol, and a 24-bit FEC field that uses a (8,4) linear code to forward error correct the preceding 24 bits. Obviously, in the case of a non-ATMLC Link Cell, the MAC protocol would use some of the 7 reserved bits to disambiguate among the various signalling Link Cell subtypes. The current MAC protocol, as Figure 4 show, uses 3 bits for this purpose. The basestation radio-port id BS_RPID is a logical id statically assigned at set-up such that no two radio-ports in radio vicinity have the same id. This logical radio-port id is mapped by the basestation to the wired network address of the basestation, and the radio-port id within the basestation. Similarly, the mobile host radio-port id MH_RPID is a logical id that is assigned to a mobile host by a basestation radio port when the mobile registers at that basestation radio port. MH_RPID is unique among the mobiles registered on the same basestation radio port.

The Link Cell bodies in the case of MACSIGLCs are small and typically contain various address information. The ATMLC body encapsulates an ATM cell. Currently, the 53-byte ATM cell is stored as is, together with link level error control information (2-byte CRC-16 for ATM cells in VCs using link level retransmission). In future, we plan to compress the ATM cell header by two mechanisms: (i) stripping the 8-bit ATM header error control (HEC) field at the transmitter, and recalculating it at the receiver, in the presence of link level error control, and (ii) transparently mapping the 24-bit VCI/VPI fields to a smaller 8- or 16-bit VCID for the duration of the air transport. The latter mechanism has the implicit trade-off of allowing a smaller number of active VCs to a mobile host.

It must be noted that the first phase implementation of the air-interface controller, described in Section 5.1, uses fixed 64-byte sized Link Cells. This has the negative side effect that there are five wasted bytes (8% bandwidth loss) for each ATMLC when no link level error control is used (three wasted bytes per ATMLC with link level retransmission), and a much larger number of wasted bytes for signalling Link Cells with consequent higher MAC level signalling overhead. The hardware is, however, reconfigurable, and a future version of the air-interface controller will remove this limitation.

4.3.3 Token Passing MAC Protocol

The time between two frequency hops on a channel is called the *Hop Frame*, which is sub-divided into several Link Cells. Access to the channel is regulated by a token passing mechanism, with the basestation acting as the master for handing out the token. The hand-off is mobile initiated which transmits *Hand-off Request Link Cells* (HRLC) based on measurements of current basestation power. On the other hand, basestation searches on its idle radio ports for mobiles that are seeking a basestation.

The basic protocol for access regulation on a channel is that of token passing, with the basestation acting as the central arbiter that decides who gets the token, and hence the transmission privilege. The token can be held for at most N=8 link cell duration, which is slightly less than 10 ms, the length of maximum allowable continuous transmission burst. This maximum interval is used to detect lost tokens in noisy channels. The token information is a part of the link cell header, in the form of two fields: TOKEN_G (1 bit) and TOKEN_R (3 bits). The G field with value 1 is used to indicate that a token is being granted to the receiver (for <= N link cell duration). The R field indicates the number of ATM cells that are queued at the sender, information is used by the basestation in scheduling the token. In the case of an idle channel, the control will just pass back-and-forth with G=1, and R=0. Finally, in case the token is lost due to noise, the mobiles do nothing (and time-out) while the basestation takes over the control and resets the token passing protocol.

The token passing approach to medium access control allows the basestation to act as a coordinator and scheduler of the available bandwidth within the cell. By taking into account QoS parameters of individual VCs when scheduling cells, the basestation can ensure that the lower layers do not make null and void the higher ATM level service quality assurances. Such would be the case, for example, with the CSMA and CSMA/CA based peer-to-peer medium access control with inherently non-deterministic delays as found in the IEEE 802.11 standard, CDPD, and various wireless LAN products.

Closely related to medium access control is the control of frequency hopping. Instead of using a hopping scheme based on measuring hopping interval in terms of real time, the number of token passes are counted to measure the length of the hop frame. In particular, hopping is done after every M=8 token grants from mobile to basestation. Of course, token losses are a nuisance in this scheme, and a straightforward counting of tokens will not work. Therefore the *effective number of token passes* are counted to decide when to hop. The effective number of token passes is the actual number of token passes plus the number of time-outs while waiting for the token.

4.3.4 Wireless Hop Scenarios

Scenario 1: New Mobile Entering the Network. After power-up a mobile begins to transmit a "Connection-Request" link cell (CRLC). This is done using a random initial frequency hopping sequence, and is at a fast rate whereby the mobile jumps to the next frequency slot in the sequence if no basestation responds to the connection request link cell. The body of CRLC consists of the globally unique mobile id and a hop sequence id. The hop sequence id defines the hopping pattern that had been assigned to the basestation radio port at start-up, and the mobile uses this hop sequence id to later hop in synchrony with the basestation radio port. The information in CRLC body is protected by a forward error correction scheme based on an (8,4) linear code. Following CRLC is a reserved fixed time interval for an interested basestation radio port to acknowledge via a CHRLCACK1 cell. Contained in CHRLCACK1 is an 8-bit logical id that the basestation assigns to the mobile for the duration of mobile's connection to the radio port. Following successful reception of CHRLCACK1 by the mobile, an exchange of CHRCLACK2 and CHRCLACK3 take place to finish the 3-phase handshake that constitutes the mobile registration process.

Scenario 2: Mobile Doing a Hand-Off. A mobile continually measures the RF power $P_{current}$ of packets it receives from its basestation. Two power thresholds are defined: P_{min} and P_{thresh} ($> P_{min}$). When $P_{current}$ falls below P_{thresh} but is still above P_{min}, the mobile initiates the process of *soft hand-off* by beginning to periodically transmit a "Hand-off Request" link cell (HRLC) with periodicity proportional to P_{thresh}-$P_{current}$. In addition, the mobile sets the "Basestation request" bit (BSREQ) in the header of all the link cells it transmits. This indicates to idle basestations as well as the current basestation that a hand-off is needed. The body of the HRLC consists of the globally unique mobile id, a hop sequence id, and the id of the current basestation. Like in CRLC, the body of HRLC is also protected by an (8,4) forward error correcting linear code. The handshake that follows an HRLC is a 3-phase handshake similar to that in the case of a CRLC as described above. Following HRLC is a reserved fixed time interval for an interested basestation to acknowledge via a CHRLCACK1 cell. The current basestation radio port maintains radio silence during this reserved interval if it receives a HRLC from its mobile. In case the power $P_{current}$ falls below P_{min}, the mobile assumes that its connection to the current basestation has been lost, and begins to continually transmit HRLC and switches to a fast hop rate. The fast hopping rate not only reduces the effect of frequency collision with other channels, but also reduces the average time to find a new basestation, thus helping in the goal of low hand-off latency. Of course, soft hand-off described earlier is the primary mechanism for low latency hand-offs.

Scenario 3: Basestation Activity at an Idle Radio-Port. The basestation with one or more idle ports actively hunts for mobiles that might want to connect. This is done according to the following process. First, using hints from the wired backbone, a frequency slot is chosen for the idle radio-port such that none of the radio-ports in the parent basestation or on neighboring basestations are using that frequency slot. The idle radio-port hops to the frequency slot thus chosen. Next, it measures power at that frequency and snoops for link cell headers. If no activity is detected at that frequency slot, a new frequency is chosen and the hunt restarted. If activity is detected, but link cell headers show that the BSREQ bit is not set then the basestation assumes that the mobile is not interested in a hand-off, and it again restarts the hunt at a new frequency. Otherwise, the basestation radio port waits for a CRLC or a HRLC link cell, or for the channel to become idle. If CRLC or HRLC is received, the basestation initiates the registration process for the powering-up mobile or for the handing-off mobile, as

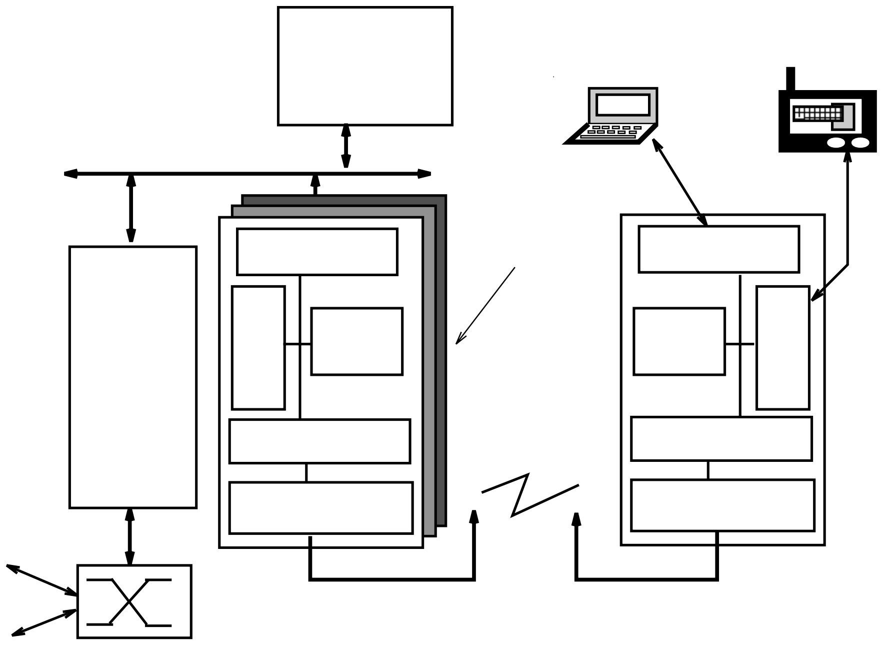

Figure 5: Architecture of SWAN System

the case may be.

5.0 Hardware and Software Implementation

The previous section presented a layered architectural overview of SWAN. This section describes the first phase implementation of that architecture. Figure 5 shows the various hardware and software components in the implementation, with a particular focus on the entities in the wireless hop. The following subsections describe the various components in detail.

5.1 The FAWN Network Adapter

From a hardware perspective, the key idea behind SWAN's wireless last hop is the use of a single reusable ATM wireless adapter that interfaces to one or more digital-in digital-out radio transceivers on one side, and to one or more standard busses (PCMCIA or PC bus, in our implementation) on the other side. The adapter, called FAWN for *Flexible Adapter for Wireless Networking* [Trotter95], provides reconfigurable data processing resources in the form of FPGAs and a software-programmable embedded ARM610 RISC processor. This flexible and reusable adapter card provides a uniform mechanism for making devices "SWAN-ready".

The FAWN card, whose architecture is shown in Figure 6, has an embedded ARM610 processor that is intended to implement low level ATM and MAC protocols, signaling, scheduling, and queue management software. A dual-port RAM provides a high-bandwidth communication link to the host. In addition, the host can also directly access memory mapped peripherals on the ARM bus at a slower speed via a cycle-stealing arbiter. A mezzanine connector on the main card allows for one or more radio link interface cards to be connected. Each radio link interface card has an FPGA-based reconfigurable air-interface controller, a serial communication controller, an A/D converter for signal power measurement, and the radio which was previously described in Section 4.3.1. Various air-interface functions, such as packetization-depacketization and signal strength monitoring, are implemented on the FPGA. The serial controller interfaces to the air-interface controller on one side and to the serial data input and output of the radio on the other side, and provides SDLC (synchronous data link control) based communication over the radio link. A second mezzanine connector brings out a peripheral expansion bus which allows a peripheral card to be memory mapped into the ARM address space.

In the current implementation of the reconfigurable air-interface controller, the communication between the air-interface controller and the ARM processor is done via two sets of ping-pong Link Cell hardware buffers, one for transmit and another for receive. The Link Cells are assumed to be of 64-byte size, leading to some bandwidth wastage when transporting ATM cell as discussed in Section 4.3.2. A finite-state machine in the hardware does the depacketization into byte stream that is necessary for the outgoing Link Cells in the ping-pong buffers to be sent to the serial controller, and the packetization of the incoming byte stream from the serial controller. The communication between the air-interface controller and the ARM processor is interrupt driven, and the hardware buffers in essence serve to reduce the interrupt rate on the ARM to once per Link Cell (approximately 0.8 ms with the current fixed 64-byte size Link Cells and the 625 Kbps radio).

The FAWN adapter emphasizes functional flexibility and reconfigurability, and its miniaturized physical dimensions enable portability. A FAWN card with one radio interface plugged-in has dimensions of 10.8 cm (W) x 1.9 cm (H) x 11.4 cm (D), and fits conveniently into the removable floppy drive bay of many laptop and laptop computers. The card, excluding the radio transceiver, consumes 2W of power. The radio transceiver that is currently used (previously described in Section 4.3.1) consumes another 0.6W in the receive mode and 1.8W in the transmit mode.

Figure 7 shows a photograph of the FAWN processor card and the FAWN radio interface card.

5.2 Basestation Architecture

Figure 8 shows the high level architecture of a typical basestation in SWAN. A basestation consists of

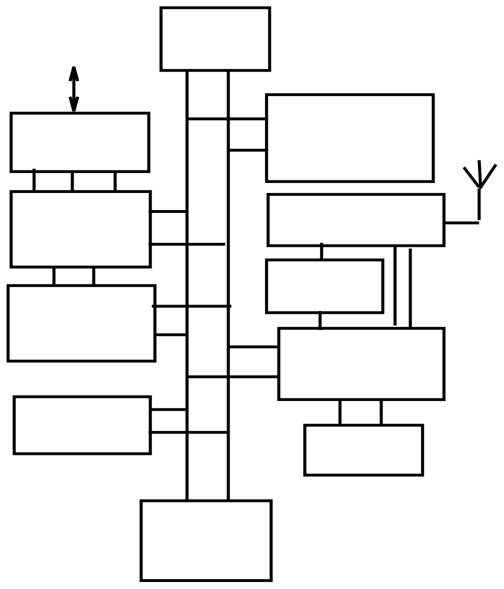

Figure 6: Architecture of the FAWN Wireless Adapter

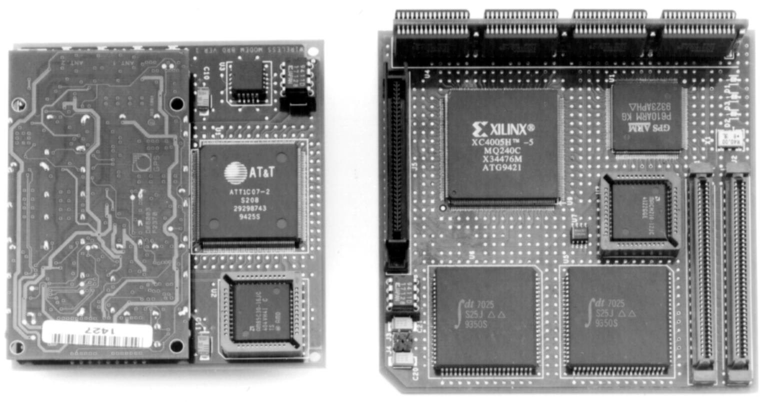

Figure 7: Photographs of the FAWN Radio Interface Card and FAWN Processor Card

multiple FAWN cards plugged into its backplane, with each card handling multiple radio transceivers. Each radio transceiver is assigned a channel (frequency hopping sequence) that is different from channels assigned to a radio in the current or neighboring basestation. Typically, in SWAN, a basestation has fewer than 3-5 radios per basestation. The preceding basestation organization results in a cellular structure where each cell is covered by multiple co-located channels. A mobile in a cell is assigned to one of the radio ports on the basestation, and frequency hops in synchrony with it. While, from the hardware and software perspective, a basestation can have an arbitrarily large bank of radios, there are limits imposed by the spectrum allocation and the type of radio. For example, as described in Section 4.3.1, SWAN uses a family of 22 codes with its current radio, each of which can be viewed as a communication channel. Obviously, since the current radio is a frequency hopping one and the codes correspond to frequency hopping sequences, the channels are inherently mutually interfering due to frequency slot collision and adjacent frequency slot interference.

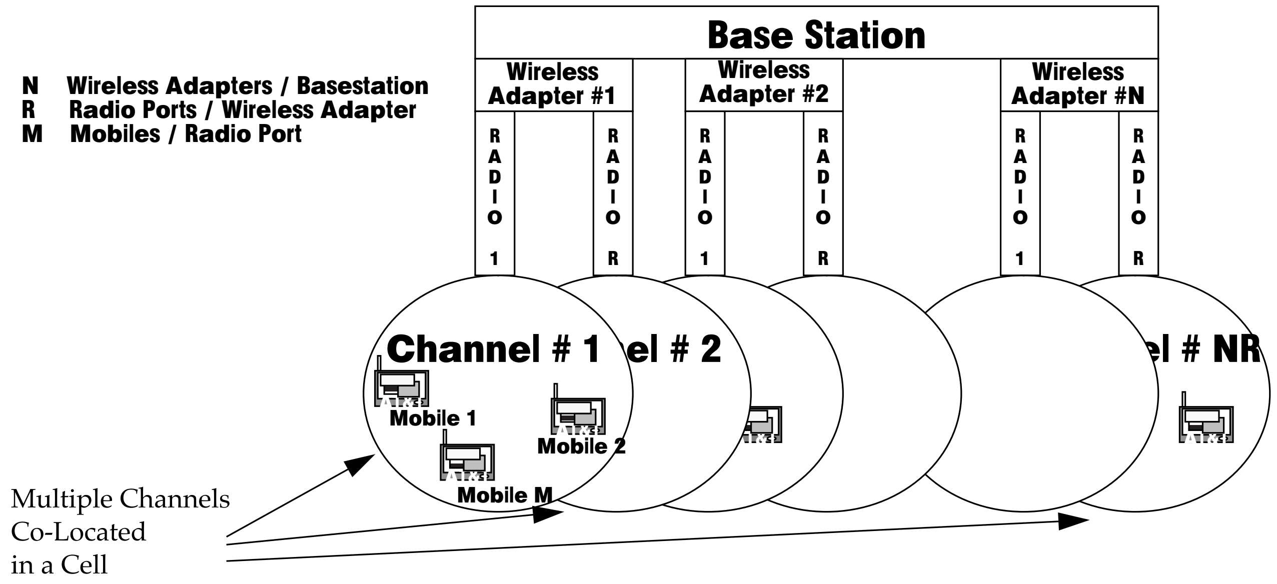

Figure 8: Abstract View of a Basestation in SWAN

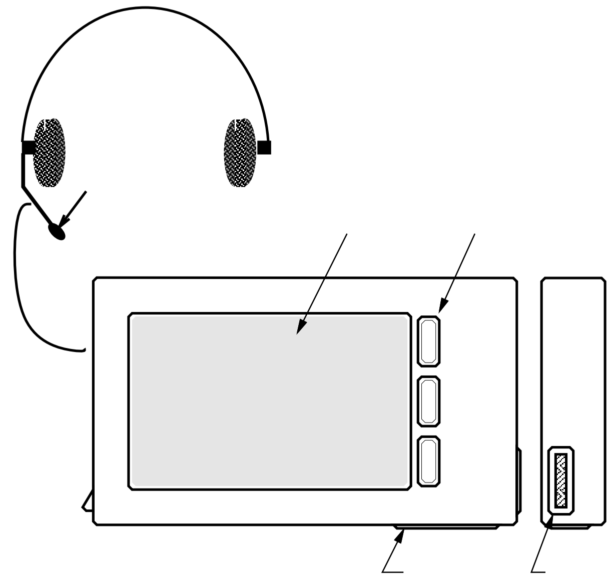

Figure 9: PMT: Dumb Multimedia Terminal in SWAN [Asthana94)

The current implementation uses PCs running Linux and Sun workstations running SunOS as the hardware platform for the basestations. In the case of PCs, the FAWN cards are interfaced to the CPU via ISA-to-PCMCIA bridge cards, while in the case of Suns SBUS-to-PCMCIA cards are used. Clearly, PCMCIA is a fairly slow 16-bit non-DMA bus and is not adequate when multiple radio link interfaces are plugged into a single FAWN card. However, it is adequate for our initial experimentation where we use one radio per FAWN card, and has allowed us to rapidly construct a prototype system using the same FAWN card on the basestations and the mobiles. In future, special versions FAWN cards may be developed for the basestations by using higher speed busses instead of the PCMCIA bus.

5.3 "Smart" and "Dumb" Mobile Hosts

SWAN has two types of mobile hosts - smart hosts and dumb hosts. Smart hosts, which have local general purpose computation resources, are built by connecting FAWN cards to laptops via PCMCIA interface. More interesting perhaps are the dumb mobile hosts in SWAN. These are multimedia terminals [Asthana94] that are formed by connecting various peripherals to a FAWN card via the peripheral expansion bus on FAWN. No other processor for local computation is provided except for the ARM processor embedded on FAWN itself. These multimedia terminals, which are called PMTs for *Personal Multimedia Terminals*, present a simple user interface (Figure 9). On the front of a PMT is a 320x240 dot-matrix LCD display along with three control buttons. A set of earphones with attached directional microphone plug into the side of the PMT to provide bidirectional audio capability. Another side of the PMT has a bar code scanner, controlled by a dedicated button. The bar code scanner can be used in applications such as inventory database access in a warehouse, patient record access by nurses and doctors in a hospital, and product information access by shoppers in a department store.

All the circuitry necessary for PMT resides on a peripheral card that plugs into the peripheral expansion connector on FAWN. In addition to the 2W consumed by the embedded FAWN card, and 0.6W/1.8W consumed by the radio (in receive/transmit mode), the PMT consumes 0.57W, for a total of 3.7W/4.9W in receive and transmit modes respectively.

Unlike PDAs and laptops, and somewhat similar to Xerox PARC's Tab terminal [Weiser93], Berkeley's Infopad [Barringer94], and Zenith's Cruisepad [Zenith95], a PMT does minimal local computing. Its functionality is entirely determined by a server on SWAN's wired network. The functionality, features, and services provided by PMT are context dependent, being defined by the server it is communicating with.

5.4 Etherware Network and System Software Implementation

The Etherware software, which implements the layered system architecture outlined in Section 4.0, is shown in Figure 10 from the point of view of a basestation. A similar implementation exists at the mobile hosts. The

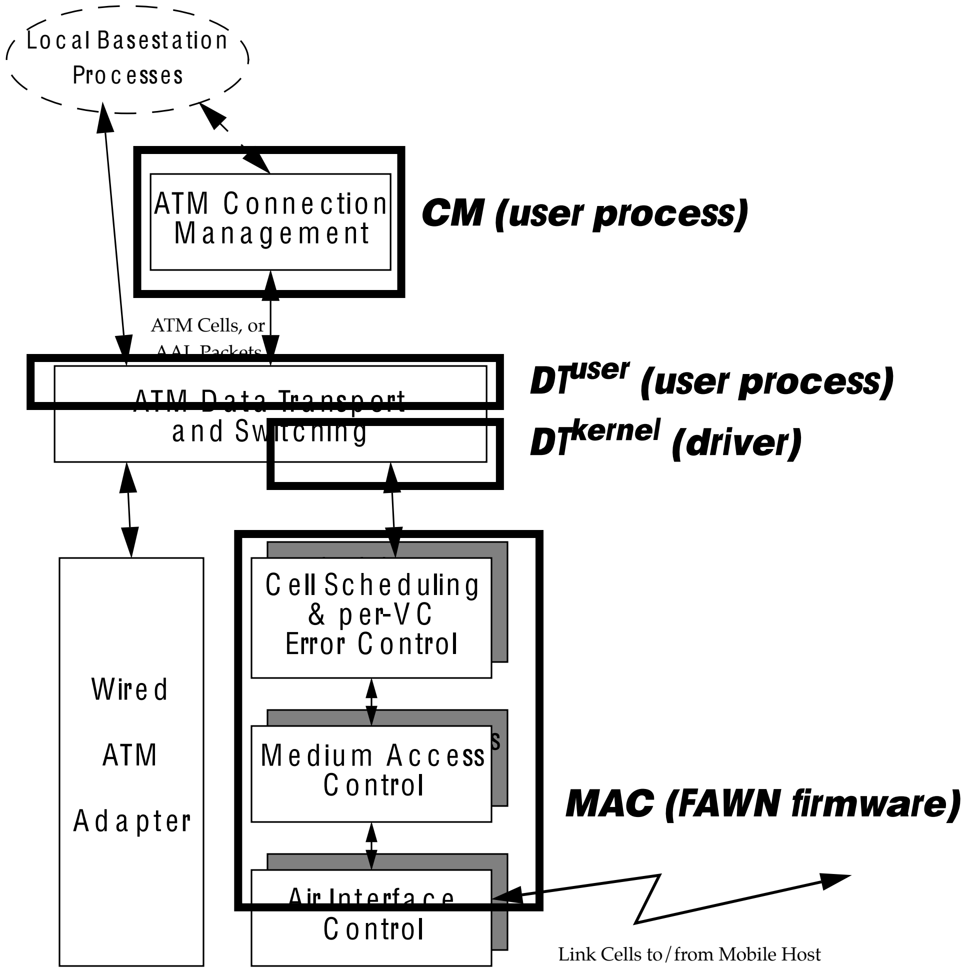
Local Basestation Processes
ATM Connection Management
CM (user process)
ATM Cells, or AAL Packets
ATM Data Transport and Switching
DTuser (user process)
DTkernel (driver)
Wired ATM Adapter
Cell Scheduling & per-VC Error Control
Medium Access Control
MAC (FAWN firmware)
Air Interface Control
Link Cells to/from Mobile Host

Figure 10: Etherware Software Implementation At the Basestation

figure also shows the correspondence between the actual software modules and the architecture functions in Figure 2. The key software modules at a basestation are: (i) CM, a user-space resident module, corresponding to the ATM Connection Management function in Figure 2, (ii) DT, a partially user-space (DT^{user}) and partially kernel-space driver (DT^{kernel}) resident module, corresponding to the ATM Data Transport and Switching function in Figure 2, and (iii) MAC, a FAWN adapter resident embedded ARM software module, encompassing the cell scheduling, per-VC error control, medium access control, and parts of air-interface control functions in Figure 2. In addition, there is an API library that is linked into the applications.

Corresponding modules also exist at the mobile hosts, except that the algorithms used are different, being tailored for mobile host functions. The API library and the CM modules also exist at wired ATM hosts. Mobility aware switches in the wired backbone will need to have functionality equivalent to CM. At the present time we have not made any of the switches, except for the basestation switches, in our wired ATM backbone to be mobility aware.

It is clear that much of Etherware is currently implemented via user mode processes SunOS and Linux, instead of kernel code. This was done to rapidly prototype an initial implementation so that we can experiment with the algorithms, particularly for the logically complex signalling protocols. Of course the ease of implementation came with an often considerable performance cost due to increased process context switch overhead and copying of data buffers across the user and kernel address spaces.

The DT module primarily switches ATM cells amongst the various wired and wireless ATM adapters on the basestations; the CM module participates in ATM level signaling to establish, tear-down, and reroute ATM VCs; and, the MAC module allocates and schedules wireless resources in response to signaling messages from the CM module, and implements low-level medium access control and inter-cell hand-offs by the mobile end-points. PCs and workstations used as basestations act as wired hosts as well, and run application processes in addition to the three modules mentioned above. In essence, basestations in SWAN are computers equipped with banks of radios. The software view at mobile end-points is similar, except that a mobile end-point has only one ATM adapter (FAWN) and the CM, DT, and MAC modules at the mobile have somewhat different functionalities.

The MAC module maintains a table of per virtual connection information to help schedule the wireless resources among the multiple ATM VCs going over a wireless channel. When a new VC needs to be opened, the CM module sends a request to the MAC module indicating the bandwidth requirements as the channel time T1 needed by this VC over a period of time T2. The MAC module uses this information to either accept or deny admission to this new VC. This bandwidth specification is also used by the MAC module to schedule transmission of cells.

The ATM signalling among the CM modules at various nodes is done on a reserved virtual circuit, and the signalling between the CM module and the MAC module at a node is done on a separate reserved virtual circuit. The CM module, the DT module, and the applications communicate among themselves using standard SunOs and Linux IPC primitives (shared memory segments and pipes). In the future, we expect to migrate all of DT functionality into the kernel code, at which time CM and applications will access DT by opening special device files.

5.5 Support for TCP and UDP Applications

The three modules, CM, DT, and MAC, together provide the basic wireless and mobile ATM functionality. We had also set an early milestone for SWAN to provide IP connectivity because native mode ATM applications are still rare, and because a complete implementation of mobile and wireless ATM is a long term goal. On the other hand applications using IP-based internet protocols such as TCP and UDP are readily available, and there already exists much previous experience with IP in a wireless and mobile context (for example, the WaveLAN product from AT&T).

IP connectivity is provided in SWAN by using IP/ATM segmentation-reassembly modules at the mobiles and the basestations, together with IP forwarding capabilities of the Linux operating system on the PC-based basestations, and a reserved VC over the wireless link for carrying IP traffic. The segmentation-reassembly module is simply a network interface driver that has ben added to the IP protocol stack, and treats the reserved

VC as a network link with 48-byte frames. The IP forwarding capability of Linux allows the basestation to act as an IP router, forwarding IP packets between the reserved VC on the wireless link and other IP network interfaces. In effect, SWAN provides IP connectivity by using IP-over-wireless-ATM. With this approach IP-level mobility is a simple matter of using the mobile-IP implementation already available for Linux [Gupta95]. Using this IP-over-ATM connectivity, mobile hosts within SWAN have immediate access to all the IP-based applications, including those requiring connections to machines outside the SWAN network.

An alternative approach, which we have not yet explored, to provide IP connectivity in a SWAN like network would be to provide IP over an end-to-end ATM VC that is set up to carry the IP traffic between any two SWAN nodes (wired or wireless). Only segmentation and reassembly would be needed at the two ends, and mobility would come for free because of the underlying mobile and wireless ATM. While more complex to implement, such an approach would be more elegant with only a single model of mobility.

5.6 Run Time Software Environment

A run time environment, available under Linux and SunOS, provides support for code development and control of FAWN cards. The control interface is implemented as two special files, /dev/fawn/mem and /dev/fawn/ctl in our systems. The former allows host resident processes to read and write memory and registers in the address space of the ARM610 processor that is embedded on FAWN. The latter allows FAWN control registers to be manipulated by writing string commands, and the FAWN status registers to be read by reading a string. For example, the commands:

```
$ echo reset > /dev/fawn/ctl
$ cat k.i > /dev/fawn/mem
$ echo run > /dev/fawn/ctl
```

when issued on a mobile host or a basestation by a suitably privileged user cause the FAWN card to be reset, the binary image k.i to be loaded for execution by the embedded ARM610 processor, and the execution to begin.

6.0 Audio and Video Transmission over SWAN

The SWAN system is still under development, and native mode ATM applications for SWAN together with signaling and MAC layer support for ATM quality of service guarantees are being developed. Experiments with ATM have so far been restricted to the performance measurement utility *netperf*, and a version of the X11 video conferencing tool *nv* that we have ported to our ATM API. Pending the availability of more native mode ATM applications for SWAN, we have used conventional TCP/IP based multimedia (and other) applications such as *nv*, *vat*, and *xmosaic*, as the initial drivers of SWAN. This is accomplished by using the IP-over-ATM approach of Section 5.5.

We have experimented with several scenarios of multimedia information transmission using the IP-over-ATM mechanism. Figure 11 depicts some of the possibilities. In one such application the mobile user sees and hears live video and audio feed, being transmitted from a camera and microphone equipped SGI Indigo. The popular internet audio- and video-conferencing tools, *vat* and *nv*, are used in a host-to-host mode. The live feed can also be replaced by pre-recorded video streams, such as movie and cartoon clips, transmitted by *nv_play* from a remote host. In a related experiment, a mobile user can use vat alone to conduct a voice conversation with another mobile user, or with a user on a wired host. This is a nascent step towards our eventual goal of fully subsuming a multimedia PBX-like functionality in SWAN. A final application scenario that we have experimented with is to let a mobile user play, in real-time, video and audio streams being fetched using the

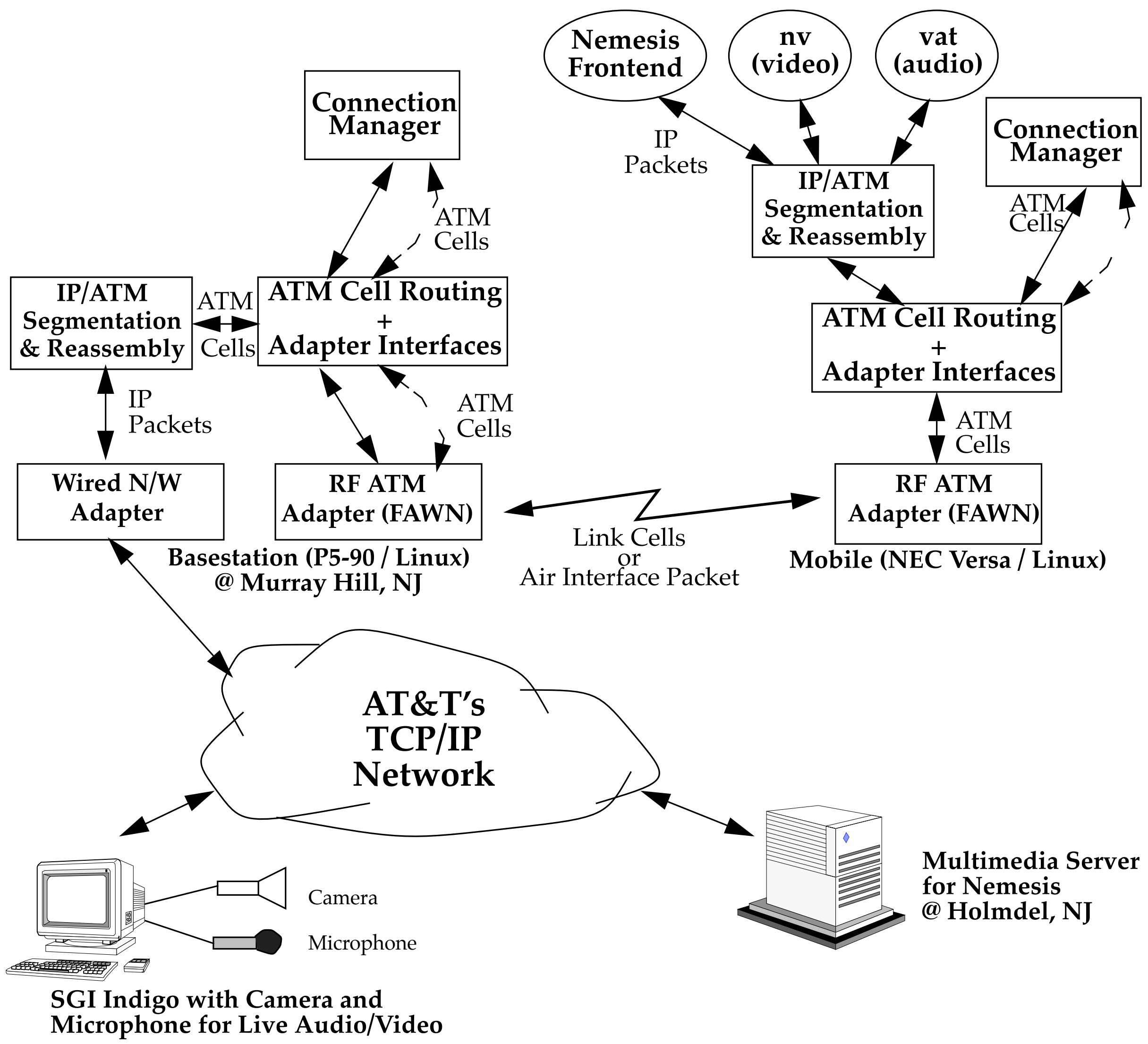
Nemesis Frontend
nv (video)
vat (audio)
Connection Manager
IP Packets
IP/ATM Segmentation & Reassembly
Connection Manager
ATM Cells
ATM Cells
IP/ATM Segmentation & Reassembly
ATM Cells
ATM Cell Routing + Adapter Interfaces
ATM Cell Routing + Adapter Interfaces
IP Packets
ATM Cells
ATM Cells
Wired N/W Adapter
RF ATM Adapter (FAWN)
RF ATM Adapter (FAWN)
Basestation (P5-90 / Linux) @ Murray Hill, NJ
Link Cells or Air Interface Packet
Mobile (NEC Versa / Linux)
AT&T's TCP/IP Network
Camera
Microphone
Multimedia Server for Nemesis @ Holmdel, NJ
SGI Indigo with Camera and Microphone for Live Audio/Video

Figure 11: Using SWAN for Transmission of Multimedia Streams

Nemesis [Katseff94] multimedia service from a multimedia archives server containing audio, MJPEG video, and accompanying documents and viewgraphs of talks given at AT&T. Nemesis uses adaptive rate-control so that a reasonable video quality is obtained at the mobile over the wireless link. The JPEG decompression is done in software at the mobile.

To conduct the above experiments with vat, nv, and nemesis, we used 486-based PC laptops running Linux as mobile hosts. The NEC VERSA and AT&T Safari that we used have built-in audio I/O, and had FAWN adapters plugged in via their PCMCIA ports. The IP-over-ATM experiments were done with a MAC that divided the channel bandwidth (nominally 625 Kbps) equally in the receive and transmit directions, and as the following section shows, the reliable TCP throughput measured in these experiments was about 227 Kb/s in each direction. With such bandwidth, and software decompression at the mobile CPU, we got 10-15 frames per second with Nemesis.

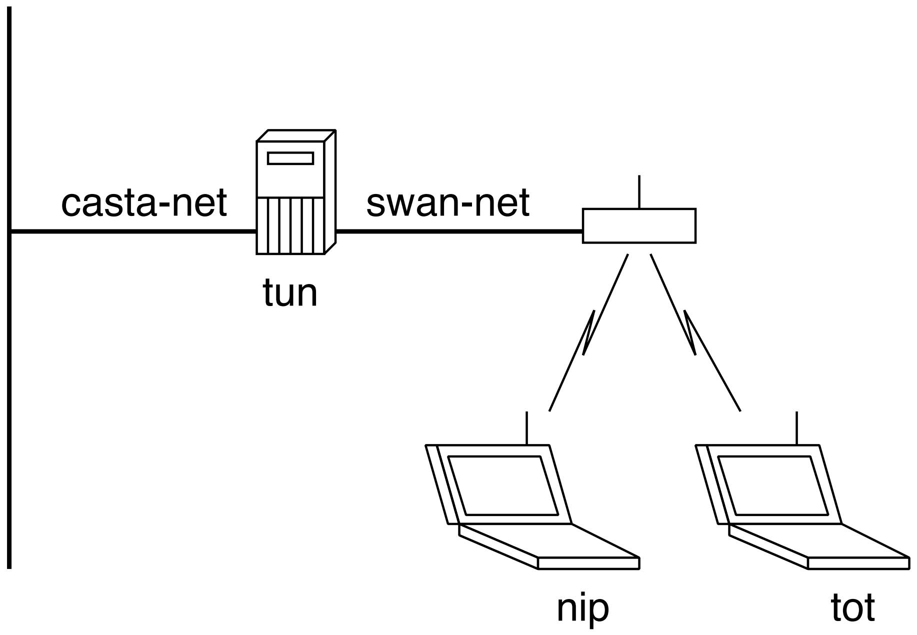

Figure 12: Network Configuration for Performance Measurements with IP/ATM-60

7.0 Preliminary Performance Measurements

The performance measurement experiments that we have conducted so far fall into two categories. Initially, we focussed on understanding the characteristics of the wireless hop and the effect of various low level design choices. For this we used IP connectivity with a simplified MAC which gives the token to the basestation radio port and the mobile for N Link Cells each repeatedly. In effect, this makes each channel a TDD (Time-Division Duplex) channel, with a frame length of N. Further, since only IP traffic is being carried in these experiments, the MAC uses a non-standard ATM cell with 60 bytes of data body. This gives us maximum performance by eliminating the wasted bandwidth, as discussed in Section 4.3.2, due to the fixed 64-byte Link Cells that are currently supported by FAWN's air-interface controller. We refer to these experiments as using IP/ATM-60 connectivity. In a second set of on-going experiments, we are measuring the performance of end-to-end ATM connectivity. This of course uses the standard 53-byte ATM cell with 48-byte body over the wireless link as well.

7.1 Experiments with IP/ATM-60 Connectivity

Figure 12 shows the topology of the network used during experimentation. *Tun* is a 90 MHz Pentium/PCI PC, configured as a gateway between *swan-net* and *casta-net* (the departmental ethernet); *nip* and *tot* are respectively a NEC VERSA 100/4 and an AT&T Safari 3181 notebooks. In the results which follow, TCP throughput was measured using *ttcp* [2] to transfer a 1 MB file from *tun* to *nip* over the wireless link, with no other traffic on the link. The default buffering in ttcp was used. The antennas were placed sufficiently close to each other that the error rate on the channel was minimal. A nearby spectrum analyzer showed that no abnormal interference was present at the time of the experiments. Using the standard MAC frame size of 10 cells for both transmit and receive frames, the test file was transferred in 36.97 seconds, yielding a TCP transfer rate of 28.36 KB/s (227 Kb/s).

2. Available as ftp://ftp.sgi.com/sgi/src/ttcp/.

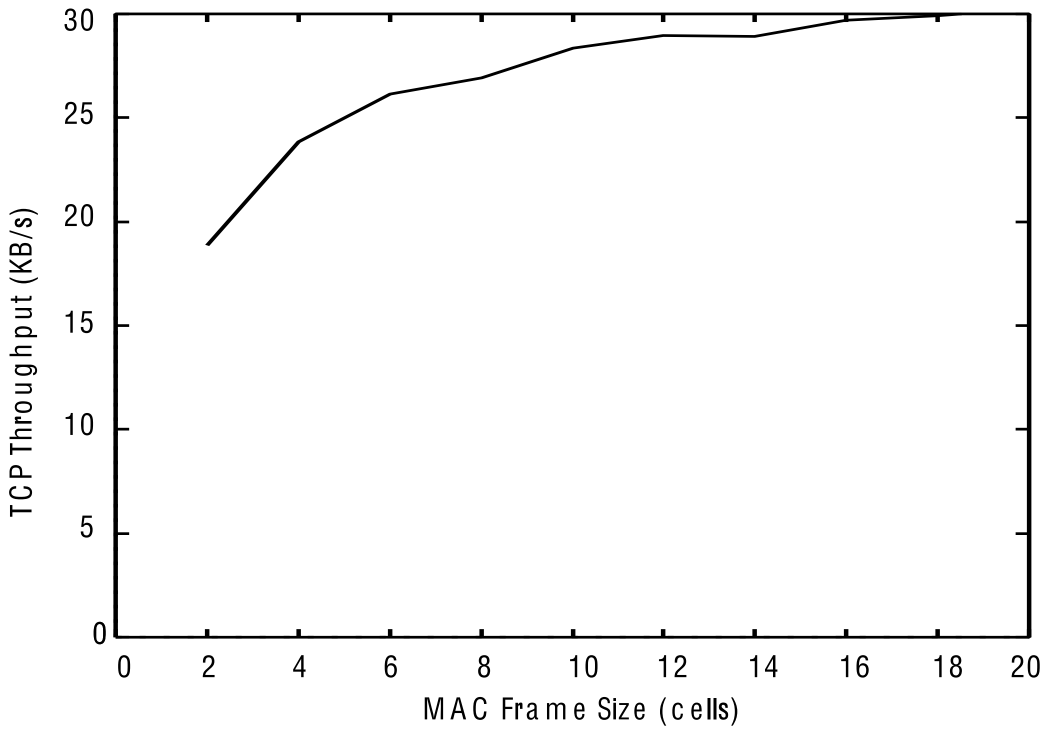

Figure 13: TCP Throughput vs. MAC Frame Size

7.1.1 Effect of MAC Frame Size on TCP Throughput

MAC frame size, which is the size of one transmission burst between two communicating MAC level entities, is an important parameter over which the system designers have significant control. Figure 13 shows the effect that varying MAC frame size has on TCP transfer rate. This figure reveals that small frame sizes give rise to low transfer rates. This is to be expected because it takes some time to turn the wireless link from transmit mode to receive mode and vice versa, due to overheads such as link synchronization. Small frame sizes do this turn around more frequently, yielding lower data rates. The graph shows that a choice of 10 cells per frame is quite acceptable; transfer rate drops off rapidly for smaller frame sizes, but much larger frame sizes do not obtain significantly higher transfer rates.

7.1.2 Effect of Transfer Size on Round Trip Delay

Figure 14 shows a plot of round trip delays on the wireless link, measured by ping time, for a range of payload sizes. The times are the average times calculated from 10,000 individual samples for each payload size measured. Payload sizes were measured in 60-byte increments, with 60-byte being the length of data body used in the air-interface packets for these IP/ATM-60 experiments. Payload sizes smaller than A fit into a single cell and result in a constant ping time. From A to B, increasing payload sizes correspond to a proportionate increase in the number of cells transmitted increasing ping time at a rate of approximately 2 ms per cell. B to C represents an increase in ping time caused by the payload starting to span two frames. The average ping time is increased by about 25 ms because the time to transmit a payload now has to include an unused receive frame between the two transmit frames. From D to E we can expect a similar increase due to the transmission now including two unused receive frames. Since the MTU (the Maximum Transmission Unit specified to the IP layer) is 1024 bytes, payload size at this stage is increased by one or two cells due to IP fragmentation.

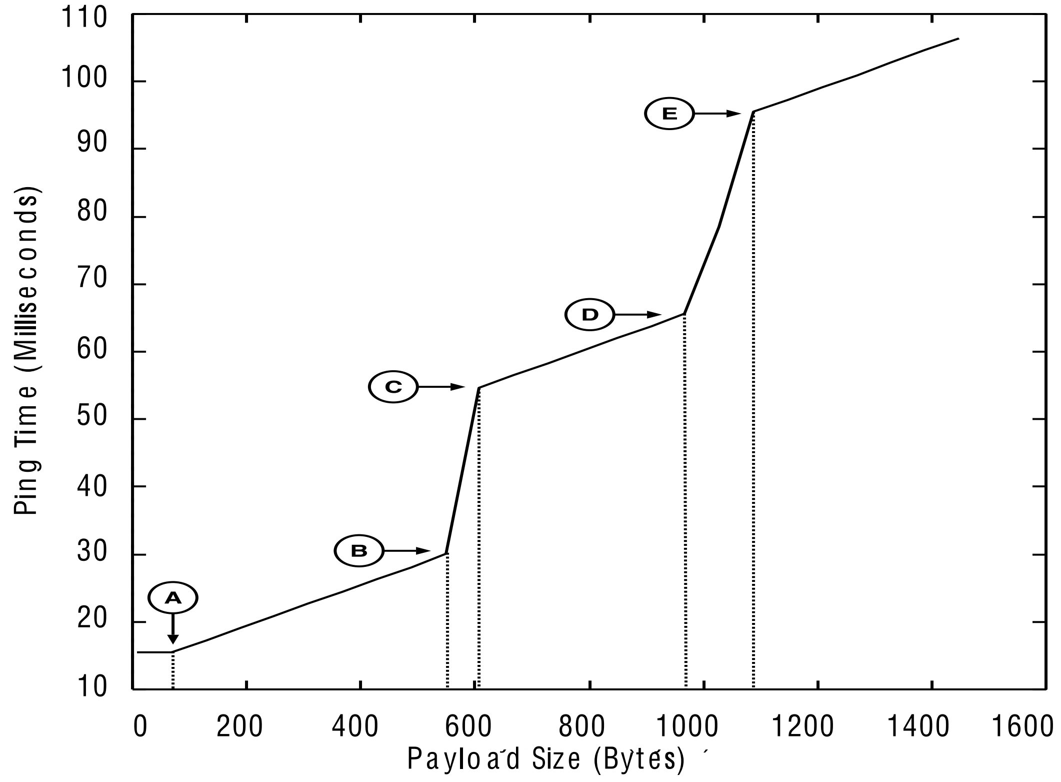

Figure 14: Ping Time in Milliseconds vs. Payload Size in Bytes

7.1.3 Effect of Packet Arrival Times on Delay Jitter

The MAC frame structure induces noticeable effects in end-to-end delay jitter. Figures 15 and 16 show the variation in ping times for two sequences of 200 consecutive packets transmitted at intervals of 40 ms and 50 ms respectively. In both cases the payload consisted of two cells. In Figure 15, the smallest ping time is observed when both ping cells are generated at the end of a transmit frame so that the corresponding reply arrives at the start of the next receive frame. The ping period is not perfectly synchronized with the frame period, so a gradual drift occurs. As the arrival of the ping cells is further removed from the end of the transmit cycle, the observed response times increase. In the worst case, the ping cells are generated at the extreme end of the transmit frame, such that one of the cells misses the current frame and has to wait for the next transmit frame. The same process is occurring in Figure 16, except that in this experiment, the ping period of 50 ms causes consecutive transmissions to occur on alternating transmit and receive frames. This results in a sequence of alternating shorter and longer ping times.

7.1.4 Discussion of IP/ATM-60 Results

Our results highlight some interesting challenges. Firstly, a TCP transfer rate of 28.36 KB/s shows that we are utilizing about 227 Kb/s or 73% of our 312 Kb/s bandwidth in a single direction. Secondly, the wireless channel is implemented within a shared medium and we have less control over it than we would have over a wired, switch based network, so the provision of any Quality Of Service (QOS) guarantees which are typically associated with ATM networks is made much more difficult. The ping round trip delay times plotted in Figures 15 and 16 are good examples of the challenges faced. The half of round trip times are essentially an estimate of end-to-end delay, an important QOS parameter. Even on a simple point-to-point link such as the one used for our experiments, the effects of having to share the wireless medium between just transmit and receive channels

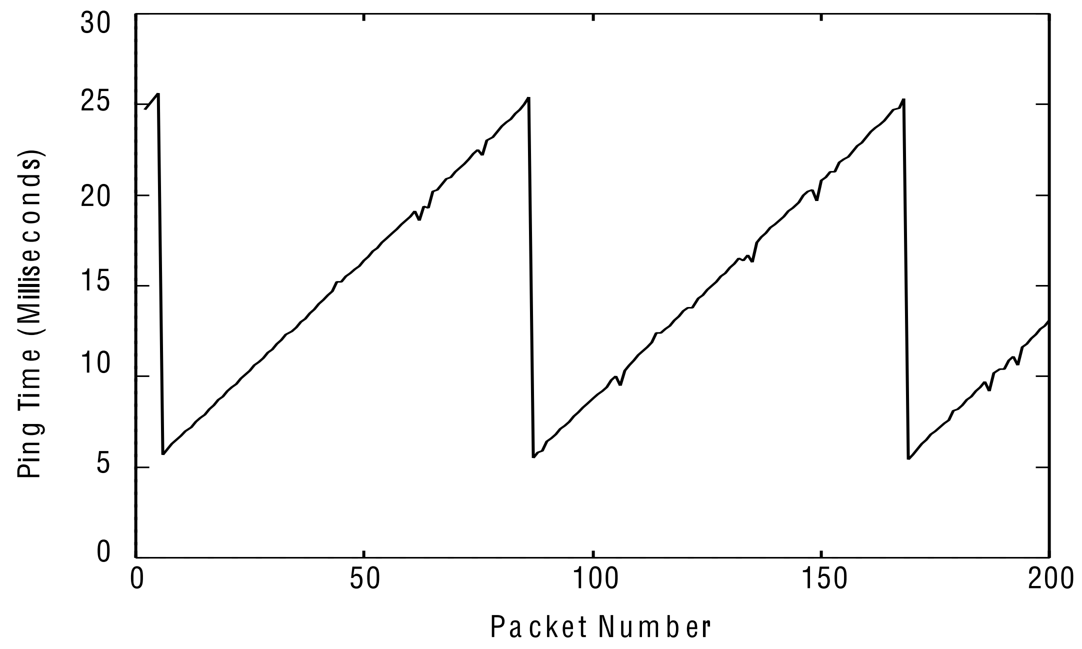

Figure 15: Ping Time in Milliseconds vs. Packet Number.
(Time between successive packets is 40 ms)

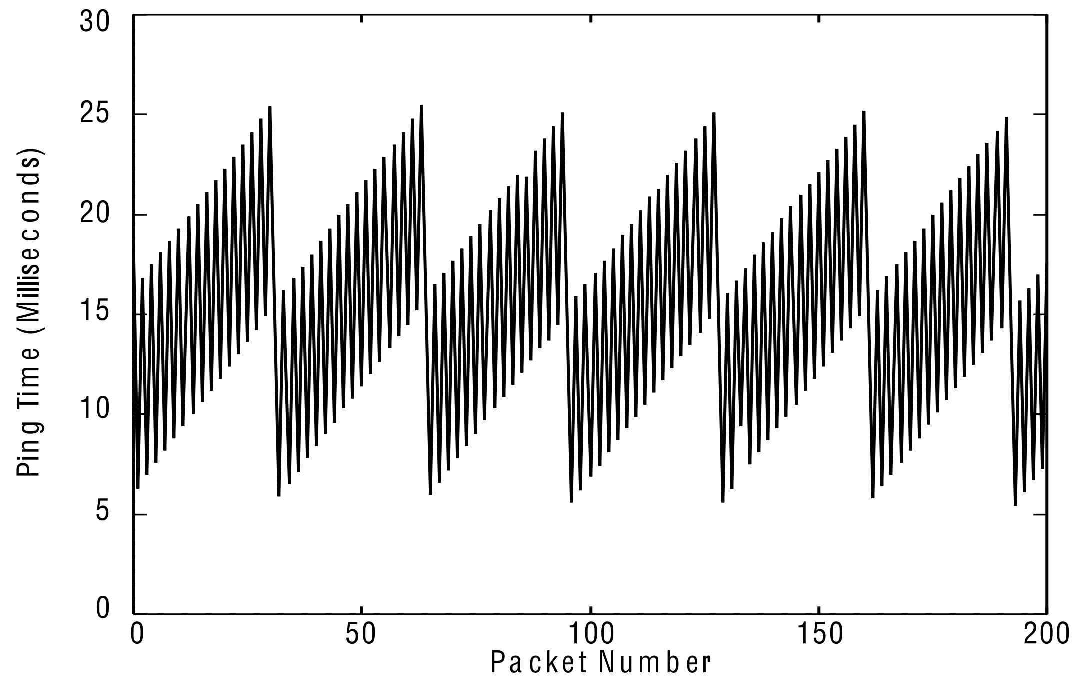

Figure 16: Ping Time in Milliseconds vs. Packet Number.
(Time between successive packets is 50 ms)

shows a marked effect on the achievable delays and delay jitter.

7.2 Experiments with ATM Connectivity

We are currently conducting experiments with ATM connectivity in SWAN. The two applications that we have used are the *netperf* performance measurement tool, and the *nv* video conferencing tool. We modified both these applications to work with our ATM API library. Using the implementation of Etherware software described in Section 5.4, we have obtained the following preliminary ATM throughput numbers in the presence of no hand-offs:

Raw Radio Link Bandwidth	312 Kbps each way
MAC Level Total Throughput	280 Kbps each way
MAC Level User Data Throughput	210 Kbps each way
End-to-end ATM User Data Throughput	190 Kbps each way

These numbers show the effect of certain features of FAWN as well as the inefficiencies of the user space implementation of the Etherware software. The discrepancy between the 280 Kbps MAC level total throughput, and the 312 Kbps raw radio link bandwidth is due to the SDLC overhead as well as because the current air-interface controller implementation does not allow full pipelining of the double Link Cell buffers so that an interrupt handler context switch worth of delay is inserted between two successive Link Cells. The 210 Kbps MAC level user data throughput is simply a result of the fact that only 48 out of 64 bytes, of each ATMLC Link Cell are being used for user data. The remaining 16 bytes are used for Link Cell header, ATM header, and wasted due to the fixed Link Cell size. The bandwidth of 210 Kbps corresponds to (48/64)*280 Kbps. Since we had no hand-offs, there was no loss of throughput due to signalling overheads. The cell loss due to noise was also quite negligible, with typically less than 0.25% cell loss rate observed.

The 190 Kbps end-to-end ATM user data throughput was measured using netperf for a virtual circuit that went from a Sun workstation, across a Fore ATM switch in the wired backbone, to a SWAN basestation, and then over the wireless link to the mobile host. The 190 Kbps measurement is the throughput as seen by the receiver with no transport layer retransmission being done by the sender. As is obvious, we are currently unable to drive the wireless link to its full MAC level ATM user data capability of around 210 Kbps. This is a result of the user space implementation of Etherware, in particular the DT^{user} module in Figure 10. ATM cells from the wired network have to traverse from the Fore driver to DT^{user} (which uses the null ATM adaptation layer AAL0 to read and write ATM cells from the wired ATM adapter). DT^{user} then sends the cells to the FAWN card. A reverse sequence of action takes place on the receive path. Clearly, the user space process DT^{user} is in the ATM cell data path between the wired network and the wireless link. This results in overheads due to context switches and extra kernel-user space data copying. Further, a bug in Fore's ATM driver sometime prevents us from transporting ATM cells in blocks of multiple cells in the AAL0 mode when cells are being both sent and received. This magnifies the inefficiencies of user space implementation. As a result, the DT^{user} module is unable to send or receive data from FAWN at full bandwidth, resulting in about 10% bandwidth loss with just one VC. The loss is greater with more VCs because of increased overheads in DT^{user}. We are currently migrating the DT^{user} functionality into a kernel resident driver, so that it will no longer be the bottleneck.

8.0 Summary

The SWAN project is exploring the seamless delivery of multimedia information to mobile users roaming in an indoor setting. This is being accomplished by a synergistic exploitation of wireless access and QoS specifiable ATM virtual circuits. The first generation system is largely functional. Using IP delivery over ATM, we already have initial experience with live as well as stored video and audio transmission using nv and vat, and also with access to a multimedia archives server via Nemesis. We are currently implementing native mode ATM applications for SWAN, and characterizing the end-to-end ATM performance in the presence of hand-offs.

9.0 Acknowledgments

We would like to acknowledge the help and contributions of Abhaya Asthana, Mark Cravatts, and B. Narendran in the SWAN project. Abhaya is the person behind the PMT multimedia terminal in SWAN. Mark helped in the hardware implementation of PMT as well as FAWN. Narendran is working on the directory and location servers for SWAN.

10.0 References

[Acampora94] A. Acampora and M. Naghshineh. "An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM networks." *IEEE Journal of Selected Area in Communications*, vol. 12, no. 8, pp. 1365-1375, October 1994.

[Agrawal95] P. Agrawal, et. al. "A Testbed for Mobile Networked Computing." In *Proceedings of 1995 IEEE International Conference on Communications (ICC '95)*, pp. 410-416, June 1995.

[Asthana94] A. Asthana, M. Cravatts, and P. Krzyzanowski. "An Indoor Wireless System for Personalized Shopping Assistance." In Proceedings of *IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.

[Barringer94] B. Barringer, et. al. "Infopad: A System Design for Portable Multimedia Access." *Wireless 1994*, pp. 382-396, Calgary, Canada, July 1994.

[Biswas94] S. Biswas and A. Hopper. "A Connection Management Scheme for a Mobile Radio LAN." In *Proceedings of the IEEE International Conference on Personal Wireless Communications*, Bangalore, India, August 1994.

[Caceres94] R. Caceres and L. Iftode. "The Effects of Mobility on Reliable Transport Protocols." In *International Conference on Distributed Computing System*, Cracow, Poland, June 1994.

[Chandler94] D. P. Chandler et. al. "An ATM-CDMA Air Interface for Mobile Personal Communications." In *Proceedings of PIMRC'94*, 1994.

[Chandrakasan95] A. P. Chandrakasan, and R. W. Brodersen. *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.

[Condon95] Condon et. al. "Rednet: A Wireless ATM Local Area Network Using Infrared Links." In *Proceedings of the First International Conference on Mobile Computing and Networking*, November 1995.

[Cox95] Donald C. Cox. "Wireless Personal Communications: What Is It?" In IEEE Personal Communications, pp. 20-35, April 1995.

[Crowcroft95] Jon Crowcroft et. al. "A Rough Comparison of the IETF and ATM Service Models." In *IEEE Network*, pp. 12-16, November/December 1995.

[Eng95] K. Y. Eng et. al. "BAHAMA: A Broadband Ad-Hoc Wireless ATM Local Area Network." In *Proceedings of 1995 IEEE International Conference on Communications (ICC '95)*, pp. 1216-1223, June 1995.

[French95] L. French and D. Raychaudhuri. "The WATMnet System: Rationale, Architecture, and Implementation." In Proceedings of IEEE Computer Communication Workshop, September 18-20, 1995.

[Gupta95] Vipul Gupta and Ben Lancki. "Mobile-IP for Linux." Available from the CS Department, State University of New York, Binghamton, NY, at *ftp://anchor.cs.binghamton.edu/pub/Linux-MobileIP/Linux-MobileIP.tar.gz.*

[Ioannidis91] J. Ioannidis et. al. "IP-based Protocols for Mobile Internetworking." In *Proceedings of ACM SIG-*

COMM '91 Conference, September 1991.

[Jain95] R. Jain, et. al. "PC-Notebook Based Mobile Networking: Algorithms, Architectures, and Implementation." In *Proceedings of 1995 IEEE International Conference on Communications (ICC '95)*, June 1995.

[Katseff94] H. P. Katseff, and B. S. Robinson. "Predictive Prefetch in the Nemesis Multimedia Information Service." In *Proceedings of ACM Multimedia 1994*, San Francisco, October 1994.

[Keeton93] 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.

[Mishra94] P. P. Mishra, and M. B. Srivastava. "Call Establishment and Rerouting in Mobile Computing Networks." *Private Communication*, September 1994.

[Porter95] J. Porter and A. Hopper. "An Overview of the ORL Wireless ATM System." *IEEE ATM Workshop*, Washington D.C., September 1995.

[Rajagopalan95] B. Rajagopalan. "Mobility Management in Integrated Wireless-ATM Networks." In *Proceedings of the First International Conference on Mobile Computing and Networking (MobiCom '95)*, pp. 127-135, November 1995.

[Raychaudhuri94] D. Raychaudhuri and N. D. Wilson. "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks." In *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 8, pp. 1401-1414, October 1994.

[Shaffer95] Richard A. Schaffer. "High-level Computing (Mobile Technology)." In *Forbes*, vol. 156, no. 8, pp. 116, October 9, 1995.

[Srivastava96] M. B. Srivastava. "Medium Access Control and Air-Interface Subsystem for an Indoor Wireless ATM Network." In *Proceedings of the Ninth International Conference on VLSI Design*, Bangalore, India, January 1996.

[Teraoka91] F. Teraoka et. al. "A Network Architecture Providing Host Migration Transparency." In *Proceedings of ACM SIGCOMM '91 Conference*, September 1991.

[Toh95] C-K Toh. "The Design and Implementation of a Hybrid Handover Protocol for Multi-Media Wireless LANs." In *Proceedings of the First International Conference on Mobile Computing and Networking*, pp. 49-61, November 1995.

[Trotter95] J. Trotter, and M. Cravatts. "A Wireless Adapter Architecture for Mobile Computing." In *Proceedings of 2nd USENIX Symposium on Mobile and Location Independent Computing*, pp. 25-31, April 1994.

[Tuch93] B. Tuch. "Development of WaveLAN, an ISM Band Wireless LAN." *AT&T Technical Journal*, pp. 27-37, July/August 1993.

[Weiser93] M. Weiser. "Some Computer Science Issues in Ubiquitous Computing." In *Communications of the ACM*, Vol. 36, No. 7, pp. 75-85, July 1993.

[Zenith95] Zenith Data Systems Cruisepad Product Introduction. "Cruisepad is not a PDA." In *LAN Magazine*, Vol. 10, No. 2, pp. 20, February 1995.

[Zhang93] L. Zhang et. al. "Resource ReSerVation Protocol (RSVP)." In *IEEE Network*, 1993.