Performance Evaluation of Connection Rerouting Schemes for ATM-based Wireless Networks

Ramachandran Ramjee, Thomas La Porta, Jim Kurose, Don Towsley

***Abstract*— Supporting mobility in ATM-based broadband networks with wireless access links poses many technical challenges. One of the most important of these challenges is the need to reroute on-going connections to/from mobile users as these users move among base-stations. Connection rerouting schemes must exhibit low handoff latency, maintain efficient routes, and limit disruption to continuous media traffic while minimizing reroute updates to the network switches. In this paper we propose, describe an implementation, and experimentally evaluate the performance of five different connection rerouting schemes. We show that one of these schemes, which operates in two phases, executes very fast reroutes (with a measured latency of 6.5 milliseconds) in a real-time phase and, if necessary, reroutes again in a non-real-time phase to maintain efficient routing. The scheme also results in negligible disruption to both audio (e.g., a 1-in-100 chance of a single packet loss at CD-quality audio rates of 128kb/s) and low bit rate video (e.g., a 2-in-100 chance of a single packet loss for 1Mb/s video) traffic during connection rerouting. Based on these results, we conclude that simple handoff schemes coupled with a connection management architecture are sufficient for supporting low bit rate continuous media applications over ATM-based wireless networks.**

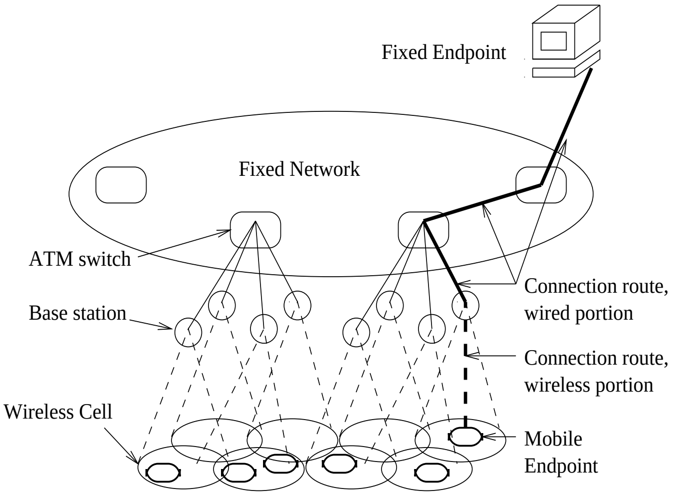

Figure 1: Wireless Network with mobile and fixed endpoints

1 Introduction

Wireless networks promise to make communication services available "anywhere", thereby removing the need for a tethered communication link. Such wireless networks are typically internetworked with a wired network (e.g., an ATM network) that provides both wired connectivity among base stations in the wireless network, as well as connectivity to other fixed endpoints. The interaction between wireless networks and wired, ATM-based, broadband networks poses many challenges. One of the most important of these challenges, the rerouting of on-going connections as mobile users move among base-stations, is the focus of this paper.

A typical wireless network with base-stations and ATM switches is shown in Figure 1. The service area in a wireless network is partitioned into cells. A cell is the region which receives its wireless coverage from a single base-station. In a typical scenario, the coverage of the cells overlaps and the base-stations are connected to each other and to fixed endpoints (e.g., hosts) through a wired, ATM-based, backbone network. A route connecting a mobile device to a fixed endpoint is indicated by a bold line in Figure 1.

In this paper, we consider end-to-end communication between a mobile host and a fixed endpoint[1]. When an end-to-end connection spans both an ATM and a wireless network, it is necessary to dynamically reroute on-going connections as the mobile users hand off from one wireless access point to another. Some of the important concerns in performing such connection rerouting are:

- limiting handoff latency;
- maintaining an efficient route;
- limiting disruption of continuous media traffic; and
- limiting network switch update rates due to rerouting

Limiting handoff latency is essential, particularly in micro-cellular networks where handoffs may occur frequently and users may suddenly lose contact with the previous wireless access point. The process of maintaining an efficient route can also potentially lead to disruptions in user traffic that are intolerable for continuous media applications such as packetized voice and video. Thus, it is important to achieve a suitable tradeoff between the goals of maintaining an efficient route and limiting disruption to continuous media traffic, while at the same time maintaining low handoff latency. In order to not overload the switches, this must be done while keeping the switch updates due to connection rerouting low.

Connection rerouting procedures for ATM-based wireless

This work was done while the first author was at University of Massachusetts, Amherst.

[1] By fixed endpoint we only mean that the endpoint is fixed for the duration of the handoff.

networks have been studied extensively [1, 3, 5, 7, 8, 9, 11, 14]. However, there has been very little *experimental* evaluation of the various rerouting procedures. In [14], rerouting is implemented such that there is no loss and very little disruption of user traffic. This is achieved by using buffers in a customized ATM switch that has knowledge of handoffs. Here, we are interested in studying connection rerouting procedures in a network using off-the-shelf ATM switches.

In this paper we propose, describe an implementation, and experimentally evaluate five different schemes for rerouting on-going end-to-end connections that span a wired ATM backbone network and then cross a wireless network to reach mobile end systems as shown in Figure 1. Our implementation and evaluation is done in a testbed consisting of an ATM-based wired network with fixed hosts, ATM switches and base-stations. The mobile hosts communicate with the base-stations using UDP/IP through a wireless link. We have also conducted the experiments by substituting an ethernet in place of the wireless link. This helps us isolate the disruption caused due to rerouting in the wired segment, and thus eliminate the impact of wireless link specific errors. The specific rerouting schemes we consider were chosen because of their simplicity in implementation; they require no special hardware support.

An important result of this work is that a relatively simple connection rerouting scheme is shown to result in only a very small probability of packet loss. In this scheme, we divide the rerouting process into a real-time component, in which the connection to the mobile host is quickly reestablished (with a latency of 6.5 milliseconds) as the mobile host moves, followed by a non real-time component in which the connection is rerouted through a more efficient path when necessary. We find that with CD-quality audio, this scheme results in no loss during the real-time handoff, and that one percent of the subsequent non real-time reroute operations suffer a loss of a single packet. Given the simplicity of this scheme (as well as the other four schemes we study), our results imply that complex connection rerouting schemes, such as those proposed in [5, 7, 8], are unnecessary for supporting connection rerouting for continuous media traffic in ATM-based wireless networks.

Each of the connection rerouting schemes we study in this paper require algorithms for determining the so-called "cross-over ATM switch" at which rerouting occurs. A second important contribution of this paper is thus the introduction of an architecture for connection setup in which connection servers maintain centralized knowledge of the connection route for the mobile users. We show how our schemes can make use of this architecture to determine the cross-over switch and perform connection rerouting.

The focus of this paper is on experimental evaluation of disruption caused by connection rerouting schemes. We note that a base-station will perform a variety of other functions including call admission control for providing guarantees on forced call termination probabilities [12], assigning frequency slots or codes, power control, etc. In this paper, we assume that sufficient bandwidth is available on the wireless as well as wired links so that the connection is not terminated during handoff. We also assume that the wireless cells (macro and/or microcells) are overlapping and the mobile endpoint can listen to multiple base-stations simultaneously, as is the case in most wide-area wireless networks.

This paper is organized as follows. In the next section, we classify the different connection rerouting approaches studied so far in the literature. We then describe our connection management architecture and five specific schemes for performing connection rerouting. The testbed and the audio/video traffic characteristics are discussed in detail in Section 4. We then discuss the performance characteristics of the five connection rerouting schemes in Sections 5, 6, 7 and 8. In Section 9, we discuss on-going work and other issues that are not addressed by this paper. Finally, we present our conclusions.

2 Classification of Handoff approaches

In this section, we provide a taxonomy of the various approaches that have been proposed in the past for performing connection rerouting during handoff. Since we are only concerned with the connection rerouting aspect of a handoff, we use the terms handoff and connection rerouting interchangeably in the remainder of this paper.

One of the first steps in performing connection rerouting involves the location of the *cross-over switch*. A cross-over switch is defined to be the farthest switch from the fixed endpoint that is also the point of divergence between the new and old routes connecting the mobile and fixed endpoint. We discuss our approach for cross-over switch discovery in the next section.

We now identify four general approaches towards connection rerouting:

- Optimistic handoff approach
- Ordered handoff approach
- Predictive handoff approach
- Chaining handoff approach

The goal of an optimistic handoff scheme is to perform simple and fast handoffs, with the *optimistic* view that disruption to user traffic will be minimal. The cross-over switch simply reroutes data traffic through a different path to the new base-station, with the connection from the source to the cross-over switch remaining unmodified. The footprint schemes of [1] fall into this category.

The goal of an ordered approach is to provide *ordered*, loss-less data delivery during handoffs. The incremental and multicast-based rerouting schemes of [7] fall into this category. However, complex protocols [7] with resynchronization mechanisms and buffering at the base-station are necessary to ensure loss-less connection rerouting.

Predictive approaches to connection rerouting were proposed in [5, 8]. The key idea here is *predict* the next base-station of the mobile endpoint and perform *advance* multicasting of data to the base-station. This approach requires the maintenance of multiple connection paths to many or all of the neighbors of the current base-station of the mobile endpoint.

Chaining approaches to connection rerouting have been proposed in [3, 9]. The basic idea behind this approach is to extend the connection from the old to the new base-stations, somewhat like a *chain*. Note that chaining results in increased end-to-end delay as well as less efficient routing of the connection.

3 Connection Management Architecture and Five Connection Rerouting Schemes

In this section, we first present our connection management architecture and show how we identify the cross-over switch. We then detail five specific connection rerouting schemes, embodying two of the four general approaches (*optimistic* and *chaining*) described in the previous section. Our choices for the rerouting schemes were primarily motivated by the need for easily implementable schemes.

Figure 2 shows a network architecture consisting of ATM switches, mobile hosts and several signaling servers. This architecture is based on a distributed call processing architecture proposed for ATM networks and extended for wireless ATM networks [10, 15]. The Connection Servers in Figure 2 are responsible for connection management. Their responsibilities include finding routes between communicating endpoints, computing end-to-end QoS measures, and modifying and releasing connections. Each ATM switch has an associated Channel Server which manages local resources such as setting up translation table entries and other fabric control functions. Each base-station also has an associated Channel Server which manages the bridging between wired and wireless links and handling handoff requests. There are other servers such as Call Servers and Location Servers which perform call processing and location management functions which are not relevant to our discussion here.

To establish a connection in this architecture, the node initiating the connection contacts the Connection Server, specifying the called party and the desired QoS measures [15]. The Connection Server computes an end-to-end route, instructs the Channel Servers involved in the route to allocate necessary resources, and returns successfully when a connection with the desired QoS measures can be accepted. If the route spans many switches or crosses administrative domains, the Connection Server can in turn contact other Connection Servers to establish partial routes to the desired destination. Rerouting during handoff affects only a small set of switches around the current base-station and is handled by a single Connection Server.

One salient feature in this proposal is that the route information is *centralized* in a Connection Server. This helps in identifying the cross-over switch immediately; the

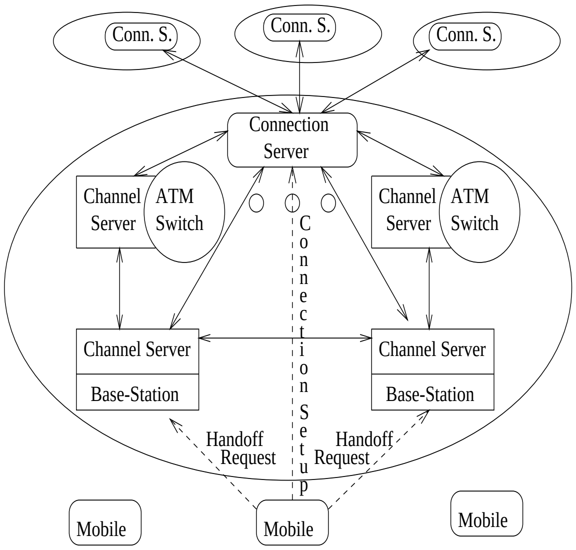

Figure 2: Connection Management Architecture

Connection Server simply determines the shortest path between the new base-station and the old route, and the point of intersection is the cross-over switch. If this particular route is heavily loaded or is unable to meet the QoS requirements for the connection, the Connection Server will attempt to find other routes [15]. The Connection Server then instructs the cross-over ATM switch to reroute the connection.

In order to reroute a connection, an ATM switch must modify its translation table entries that control its internal cell routing. A translation table entry in an ATM switch is a mapping between two 3-tuples, corresponding to the input and output of the switch [4]. Each 3-tuple consists of a port, a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI). The modification of the translation table entry for rerouting can be accomplished in three ways:

- *Break-make:* dismantle the old entry (Break) and then create a new entry (Make)
- *Make-break:* create a new entry (Make) and then remove the old entry (Break)
- *Atomic:* switch the mapping of the entry of the input port to the new entry

Most current ATM switches do not support the *atomic* operation. Thus, we have two ways of modifying the translation table in the ATM switch; these approaches will be taken in Schemes 1 and 2.

Just as the internal routing of cells in the ATM switch is controlled by a switch translation table entry, translation tables at the base-station control its internal routing. Thus, one can view the base-station as a simple switch which either forwards packets over the wireless link by bridging the protocols in the ATM and wireless networks (AAL5 to UDP/IP in our case) or forwards cells over the ATM network to the neighboring base-stations.

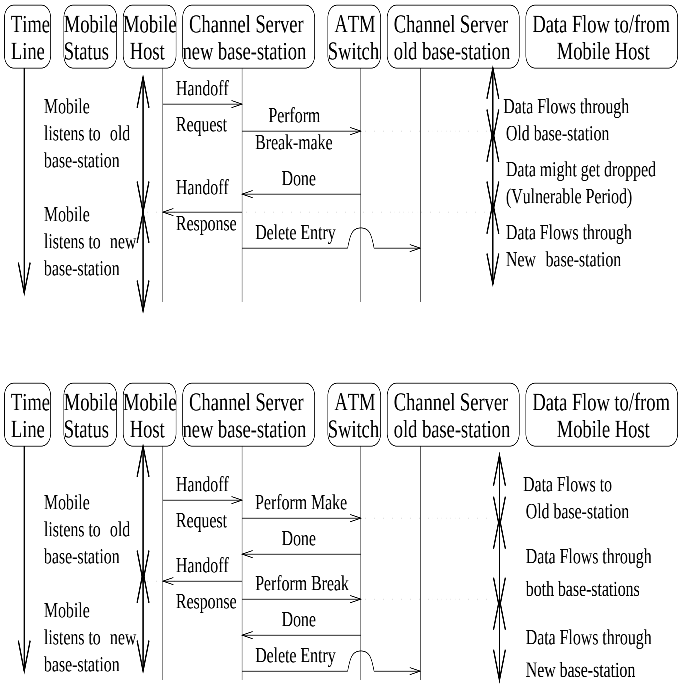

Figure 3: (a,b): Detailed flows of Schemes 1 and 2

We next look at the five rerouting schemes in detail.

3.1 Break-make and Make-break: Schemes 1 and 2

These schemes can be categorized as optimistic schemes based on our earlier classification. *Break-make* also borrows some ideas from the incremental reestablishment scheme [7] under the ordered handoff approach but does not include the buffering, forwarding and resynchronization mechanisms. *Make-break* is similar to the multicast-based reestablishment scheme [7], again, without any ordering or resynchronization mechanisms.

Figure 3 depicts the case when the cross-over switch is one hop away from both the new and old base-stations. When the cross-over switch is further than one hop away, the connection server will be needed to determine the cross-over switch; this case is illustrated in the context of the Scheme 5 in the next subsection.

In the *break-make* scheme, we execute two sequential operations; *break* (which removes the old translation table entry) and *make* (which creates a new translation table entry). This method results in a vulnerability period between these two operations at the switch during which the switch has no translation table entry and drops cells. We have been able to reduce the length of this period to approximately 3.5 milliseconds in our implementation. The effects of this vulnerability period on continuous media traffic are investigated in section 6.

In the *make-break* scheme, we first create a new translation table entry in the ATM switch (Make) and later remove the old translation entry (Break). This results in cells being multicast from the cross-over switch to both the new and old base-stations for a short period of time during the handoff process. Thus, there is a possibility that a mobile host will receive a packet from the old base-station, switch to the new base-station and receive the same packet as a duplicate. Note that there is no cell loss at the cross-over switch in this scheme. However, packets can still be lost because of congestion in the base-station or if only some of the cells of a packet make it to the new base-station (that is, a higher level data unit can be split due to rerouting).

The steps involved in these schemes are (see Figure 3):

- Mobile host sends a Handoff request message to the new base-station, identifying the old base-station.
- The new base-station adds local translation table entries for bridging purposes.
- The new base-station identifies the best route to the cross-over switch[2], allocates resources along the route and then finally exchanges messages with the cross-over switch, which executes break-make or make-break operations.
- The new base-station sends back a Handoff response message to the mobile host, instructing the mobile host to transmit/receive through the new base-station.

Finally, the new base-station also instructs the old base-station and the ATM switches on the old route to deallocate resources.

3.2 Chaining with break-make and make-break: Schemes 3, 4 and 5

The *chaining* scheme, Scheme 3, simply extends the connection route from the previous base-station to the new base-station. By provisioning some bandwidth using Virtual Channel (VC) or Virtual Path (VP) reservations between neighboring base-stations, chaining can be accomplished quickly (since the cross-over switch will not be involved). However, chaining will typically degrade the end-to-end performance (e.g., end-to-end delay) of the connection and the connection route is no longer the most efficient. This could lead to dropped calls if resources in the wired network are not available for chaining. On the other hand, the first two schemes lead to efficient routing of the connection, but involve longer handoff processing delays in order to locate the cross-over switch and establish the new route. This observation leads us to the design of two new schemes, Schemes 4 and 5, where we perform an immediate handoff using the chaining scheme and, if necessary, perform a reroute in non-real-time using break-make or make-break respectively.

We now illustrate the steps involved in Scheme 5 in detail (chaining followed by make-break). The steps involved in Scheme 4 are similar. Scheme 3 corresponds to the real-time phase of Scheme 5. The steps involved in Scheme 5 are (see Figure 4):

- Mobile host sends a Handoff request message to the

[2] It may or may not be the shortest route depending on the QoS requested and the load conditions in the ATM network.

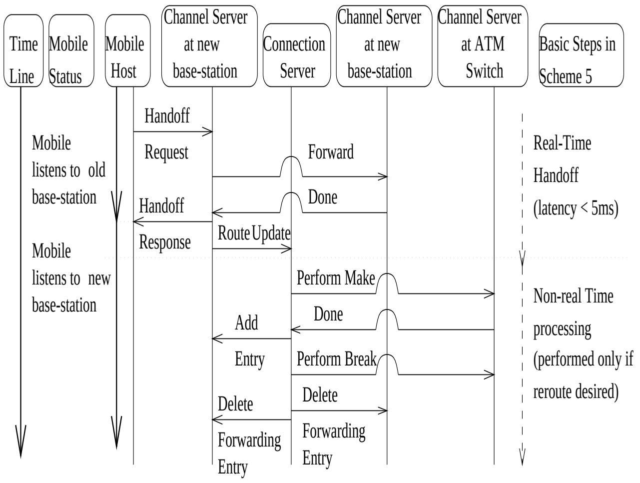

Figure 4: Detailed flows of Scheme 5

new base-station, identifying the old base-station and its Connection Server.

- The new base-station adds local translation table entries for its internal routing.
- The new base-station asks the old base-station to forward packets pertaining to the mobile host.
- The new base-station sends back a Handoff response message to the mobile host, instructing the mobile host to transmit/receive through the new Base-station. These four steps are accomplished in real time (with a latency of 6.5 milliseconds) to enable the mobile host to quickly switch over to the new wireless link. At this point, the chaining portion of the handoff has been completed.
- The new Base-station passes the updated route information to the Connection Server.
- The Connection Server performs necessary QoS computations on the new route. Note that it has centralized knowledge of a significant portion of the route and can perform this calculation easily. If it detects a possible QoS guarantee violation, or if fixed links are getting congested and route efficiency is desired, it undertakes the following steps. In all other cases, the handoff flow terminates here.
- This is the first step of the *make-break* portion of the handoff. The Connection Server identifies the best route to the cross-over switch, allocates resources along the new route and sets up a new routing entry in the cross-over switch. The switch now multicasts cells received from the source to both the base-stations.
- The Connection Server informs the new base-station of the completion of route change, which then starts using the new route.
- Finally, the Connection Server exchanges messages with the ATM switch, tearing down the old routing entry. The Connection Server also requests the old and new base-stations and switches in the old route to deallocate the old resources.

Although we have illustrated how our handoff proposals can work with a particular connection management architecture, the handoff schemes in this paper can be employed in any connection management architecture with appropriate cross-over switch discovery algorithms. Here, we have illustrated these schemes in the context of a particular call management architecture which facilitates the location of the cross-over switch.

We now briefly examine the content of the signaling messages shown in Figures 3 and 4. These signaling messages are only few bytes long. The longest message is the Handoff Request message from the mobile device. This message is 44 bytes long and includes the mobile identity, old base-station channel server identifier, and the 3-tuples (VPI,VCI,port) of the translation table entry at the nearest ATM switch (obtained initially through the Connection Setup Response messages and updated each time by the Handoff Response message). The Route Update message to the Connection Server contains the identity of the mobile endpoint and the two base-stations involved in the chaining scheme.

The messages to the ATM switch channel server (Perform Make, Perform Break, and Perform Break-Make) contain the necessary 3-tuples (VPI,VCI,port) for modifying the switch translation table entry. The ATM switch channel server is directly connected to the switch. The channel server uses the SNMP interface at the (FORE) ATM switch to modify translation table entries. We chose the SNMP interface as opposed to FORE's proprietary SPANS signaling protocol [4] to modify the switch translation table entries since we were able to obtain a smaller vulnerability period for the Break-Make operation through the SNMP interface. The messages to the base-station Channel Server (Add Entry, Delete Entry, Delete Forwarding Entry, and Forward) also contain only the necessary 3-tuples for the base-station to update its translation table entries. Thus, the signaling overhead generated by these messages is low.

4 Experimental testbed

Our testbed (Figure 5) consists of Sparc workstations, a Toshiba Laptop and a FORE ATM switch. The workstations serve as base-stations and fixed hosts while the Laptop acts as the mobile host. The fixed hosts, base-stations as well as the mobile host run the Solaris 2.4 Operating System. An ATM network connects the base-stations to the fixed host. The wireless link connecting the mobile host to the base-stations uses AT&T's WaveLAN product. The base-stations are connected to the wireless antennas, also called *Wavepoints*, through ethernet. The Wavepoints serve as MAC-level bridges, forwarding data packets from (to) the base-stations (via ethernet) to (from) the WaveLAN air interface. The ethernet and the wireless segments shown in Figure 5 are also isolated from other networks, shielding the testbed from extraneous network traffic (NFS and other traffic were carried on a separate ethernet link which is not shown in the figure).

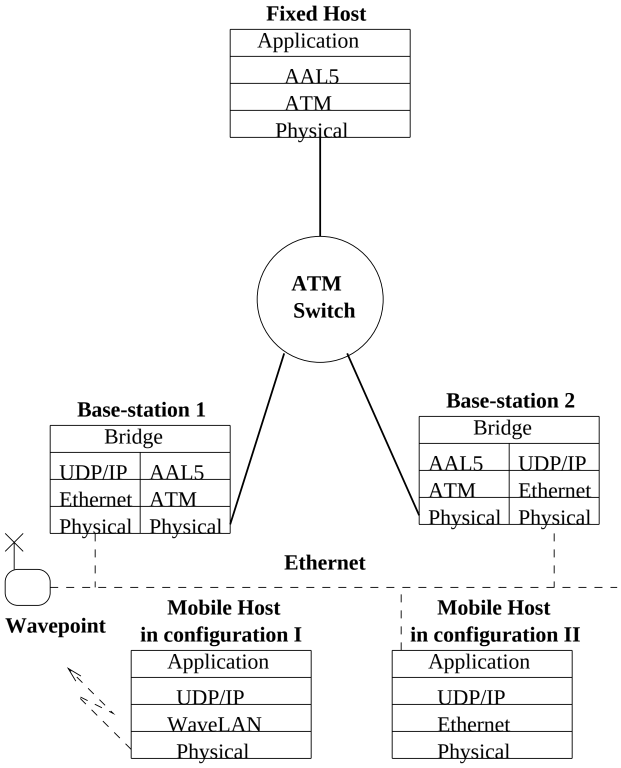

Figure 5: Experimental testbed

As mentioned earlier, we are primarily concerned with the effect of rerouting on the backbone network (ATM) portion of the connection. To eliminate the effects of the wireless media (interference, fading etc.), we also conducted experiments with the mobile device connected to the base-station through an ethernet. This allowed us to focus on the consequences of connection rerouting: loss, duplication, and reordering of packets. As we shall see later, the losses suffered in the experiments where the mobile host is connected through the wireless link is sometimes slightly larger than the losses in the experiments where the mobile host is connected through the ethernet.

The network protocol stack at the fixed host and the mobile host are different (see Figure 5). At the fixed host, we use the ATM Adaptation Layer (AAL5) protocol over ATM whereas at the mobile host, we use UDP/IP over WaveLAN. The base-stations perform the bridging function between the different protocol stacks. In a production system, this bridging function would likely be done in hardware and the effect of protocol conversions will be minimal. However, since we are primarily interested in packet loss caused to a single connection due to handoff, the impact of the overhead involved in user-level bridging and protocol conversion is unimportant.

Note that the main conclusions of this paper will apply to an end-to-end ATM system with wireless endpoints as well. A difference between our testbed and an end-to-end wireless ATM system is that, in our testbed, the base-stations are responsible for reassembling ATM cells into UDP packets. In an end-to-end ATM system, the mobile host would reassemble the ATM cells. However, cell loss during handoff results in the loss of one or more application packets in both systems, regardless of where reassembly is performed.

4.1 Experimental Configurations

We report experiments using two controlled configurations of the testbed.

In the first configuration, the mobile host was connected to the base-stations through WaveLAN using a single Wavepoint. Initially, we attempted to use two Wavepoints, each associated with its respective base-station. However, handoff across two AT&T's Wavepoints is possible only if the Wavepoints are connected through an ethernet segment. This restriction leads to the two handoffs (wireless link level and ATM reroute) interfering with each other, resulting in some anomalies. Therefore, we decided to use only a single Wavepoint in our experiments. Handoff requests were triggered periodically from the mobile host, with sufficient time between successive handoff requests to ensure that any connection reroute operation triggered by a previous handoff request is completed before the next handoff request is issued. Since we had only two base-stations in our testbed, the mobile host alternates the handoff requests between the base-stations. All measurements in this configuration were taken and averaged over 100 handoff attempts. We refer to this as configuration I in the remainder of the paper.

In the second configuration (referred to as configuration II in the remainder of the paper), the mobile host was connected to the base-stations through an ethernet. We were, thus, able to isolate the effects of wireless specific errors and also extend the results of video experiments on configuration I beyond 1 Mb/s. Again, handoff requests were periodically triggered from the mobile host. All measurements in this configuration were taken and averaged over 1000 or more handoffs. We also plot 90% confidence intervals for the data points obtained from these experiments.

We now describe the characteristics of the audio and video traffic sources. Using these traffic sources, we evaluate the performance of the five connection rerouting schemes.

4.2 Audio and Video Traffic

We characterize audio traffic by its timing component. Every T milliseconds, a 160 byte packet is transmitted by the source. For $T = 20$, this corresponds to telephone quality audio of 64kb/s. Note that in a typical audio conversation, there will be silence periods during which no packets are transmitted. During these periods, there would be no disruption of traffic during handoff. In our experiments, the audio source transmits continuously (i.e., there are no silent periods) so that we can detect all potential disruptions during handoff.

We performed experiments in both the downlink and uplink directions. In the downlink direction, the fixed host segments a 160 byte audio packet using AAL5, and transmits the four resulting ATM cells[3]. The base-station first

[3]The first three cells carry 47 bytes of audio data, 1 byte of AAL5

reassembles the AAL5 PDU. The bridging software then adds a UDP/IP header, and transmits the packet to the mobile host. In the uplink direction, the mobile host adds a UDP/IP header to the 160 byte audio packet and transmits the packet to the base-station. The base-station extracts the audio payload, segments the data using AAL5, and transmits the four resulting ATM cells to the fixed host.

The video traffic is characterized by a Constant Bit Rate (CBR) source sending packets every 30 milliseconds. This corresponds to a frame rate of approximately 30 frames per second, the frame rate for full motion video. We varied the packet size, and thus the resolution of the video, to observe the effect of handoffs on video of different bit rates. We assumed a CBR type of traffic so that our results are unaffected by the instant at which a handoff occurs.

In the following sections, we present the results of the experiments that we performed on the testbed.

5 Handoff latency in connection rerouting

Handoff latency is defined to be the time duration between the following two events at the mobile host: initiation of handoff request and the reception of handoff response. Table 1 lists the handoff latencies incurred by the five connection rerouting schemes using configuration I. Configuration II using ethernet results in latencies which are about 3 milliseconds lower. Note that the vulnerability period illustrated in Figure 3(a) differs from the handoff latency. We will examine the effects of both handoff latency and the vulnerability period on packet loss in the following sections.

Rerouting Scheme	Latency for 1 connection (in ms)	Latency for 2 connections (in ms)
break-make (b-m)	46.4	55.5
make-break (m-b)	37.7	46.6
Chaining	6.5	7.0
Chaining w/ (b-m)	6.5	7.0
Chaining w/ (m-b)	6.5	7.0

Table 1: Handoff latency in connection rerouting schemes

The handoff latency is slightly higher for the *break-make* scheme as compared to the *make-break* scheme because the *break-make* scheme involves two operations (Break and Make) at the switch before the handoff response can be sent whereas only one operation (Make) is needed in the *make-break* scheme (see Figure 3). The chaining scheme is fast because we preassign Virtual Channels (VCs) between neighboring base-stations and, thus, translation entries at the cross-over switch need not be changed. If VCs were not preassigned, the handoff latency in the chaining scheme would be comparable to that of the *make-break* scheme. Schemes 4 and 5 perform their rerouting operations after handoff and thus those operations do not affect the handoff latency of these schemes. Also, note that the handoff latency measurements are dependent on the number of connections of the mobile endpoint that must be rerouted. This is because each connection corresponds to a translation table entry in the switch. Therefore, rerouting multiple connections implies that multiple translation table entries have to be modified, resulting in higher latencies. As shown in Column 3 in Table 1, rerouting two connections results in slightly higher handoff latencies for all these schemes.

We next look at the packet loss incurred per handoff by the five connection rerouting schemes.

header and 5 bytes of ATM header; the last cell carries the remaining 19 bytes of audio data, 28 bytes of padding and the headers.

6 Performance of Break-make and Make-break (Schemes 1 and 2)

In this section, we investigate the impact of *make-break* and *break-make* schemes on the downlink and uplink traffic, respectively.

6.1 Downlink

We first present results of the experiments on audio and video traffic in the downlink direction.

6.1.1 Impact on Audio

In this set of experiments, the fixed host in Figure 5 transmits a 160 byte audio packet every T milliseconds. The mobile host initiates a handoff request every two seconds, resulting in the handoff activity shown in Figure 3. Sequence numbers are given to the packets by the source. By examining the sequence numbers of packets received at the mobile host, we detect packet loss and duplication. We compute the expected number of packets lost/duplicated per handoff from a set of 100 handoff attempts. We then vary T, the packetization interval for the audio source, and repeat the experiment. Our results, using configuration I, are presented in Figure 6.

In Figure 6, the bit rate of audio traffic varies from 9.8kb/s (T = 130ms) to 128kb/s (T = 10 milliseconds) from left to right on the x-axis. The y-axis represents both the expected packet loss in the two schemes as well as the expected number of duplicates in the *make-break* scheme. The two schemes perform similarly for bit rates of 16kb/s (T = 80 ms) or less. There is negligible loss in the case of the *break-make* scheme and no loss and negligible duplication in the case of the *make-break* scheme. However, the *make-break* scheme performs significantly better than the *break-make* scheme in the case of higher bit rate audio. For example, with CD quality audio (128kb/s, T = 10ms), we have an expected loss of 0.39 packets per handoff with the *break-make* scheme while we still have negligible loss and an expected duplication of 0.3 packets in the *make-*

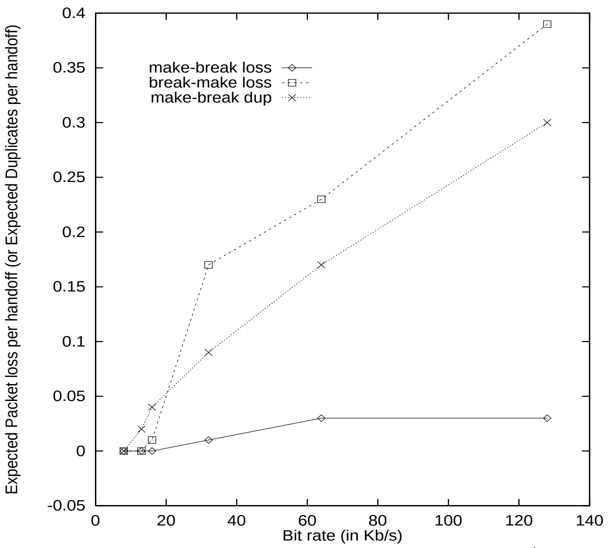

Figure 6: Disruption due to Schemes 1 and 2 (audio, downlink, configuration I)

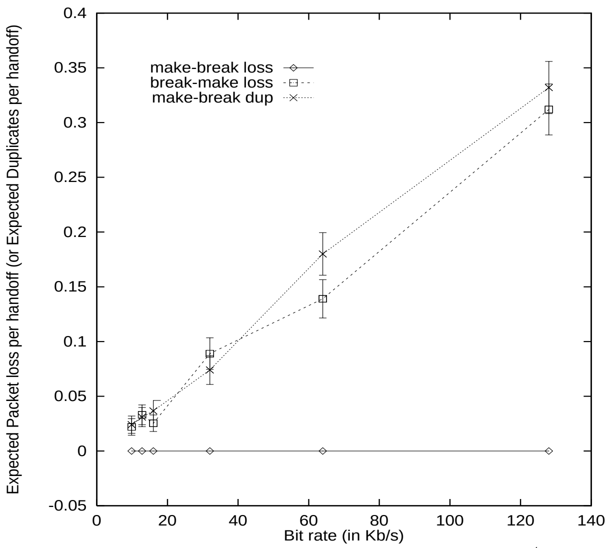

Figure 7: Disruption due to Schemes 1 and 2 (audio, downlink, configuration II)

break scheme. Note that the loss suffered in the *break-make* scheme is similar in magnitude to the packet duplication rate in the *make-break* scheme since the handoff latencies for both the schemes are similar (see Table 1).

Recall that packet loss in the *break-make* scheme occurs during the vulnerability period shown in Figure 3(a). Our measurements on our ATM switch shows that the length of the vulnerability period is approximately 3.5 milliseconds. For CD quality audio, a packet is sent every 10 milliseconds. Thus, on the average, we expect a disruption of about 0.35 packets per handoff which agrees with our experimental observations. Further, in the course of all our experiments, a single handoff never resulted in the loss of more than one packet. Thus, an expected packet loss of 0.35 packets translates approximately to the loss of a single packet every 3 handoffs. Also, note that the use of the *make-break* scheme requires that an application handle/distinguish duplicate packets, as in the case of an internet packetized audio application such as *vat* [6].

In Figure 7, we show the result of the same experiment using configuration II of the testbed (ethernet substituted for the wireless link). Since we were able to automate this experiment without any perturbation to the experimental setup, we repeated the experiment 1000 or more times and computed the 90% confidence intervals. The results are similar to the results using configuration I. However, losses in configuration I are slightly higher. On further examination, this was found to be due to occasional burst losses suffered in the wireless medium. This revalidates our choice of using an ethernet to eliminate the wireless specific effects and concentrate on the losses due to mobility and connection rerouting in isolation.

6.1.2 Impact on Video

In this subsection, we examine the effect of *make-break* and *break-make* schemes on video traffic. The fixed host in Figure 5 now transmits P bytes of video data every 30 milliseconds. The base-station again reassembles the AAL5 PDU and transmits the packet through the wireless link to the mobile host as a UDP/IP packet. As with the previous audio experiments, we compute the expected number of losses and duplicates averaged over 100 handoff attempts. In the case of video, however, we vary P (which represents the video frame size) while maintaining the timing constant at 30 milliseconds.

The loss suffered in the *make-break* scheme can be explained as follows. Larger packet sizes are used in transmitting high bit rate. Larger packets imply that a) a larger number of cells comprise one video packet and b) a higher latency is involved in the transmission of the packet in the low capacity wireless link. Larger number of cells per packet results in a higher probability that the *make* operation is performed when some of the cells of a packet have already been sent through the switch while, other cells of the same packet have not. In this case, the packet received at the new base-station will experience a Cyclic Redundancy Check (CRC) error when it is reassembled. The new base-station will not transmit this packet to the mobile host. If the mobile host manages to receive this packet from the old base-station, this event will contribute to fewer packet duplications; otherwise this event results in higher packet loss. The second factor, namely, the higher latency in the transmission of large packets in the wireless link, results in the packet transmitted by the old base-station not completely arriving at the mobile host before the mobile host switches to the new base-station. This leads to the reception of fewer duplicate packets in the *make-break* scheme, thereby increasing the amount of packet loss again. Thus, we expect to see higher packet loss and lower packet duplication in the *make-break* scheme as the bit rate of video traffic is increased. We observed little or no loss in the *make-break* scheme during the downlink experiments on audio primarily due to the low bandwidth and small packet sizes (160

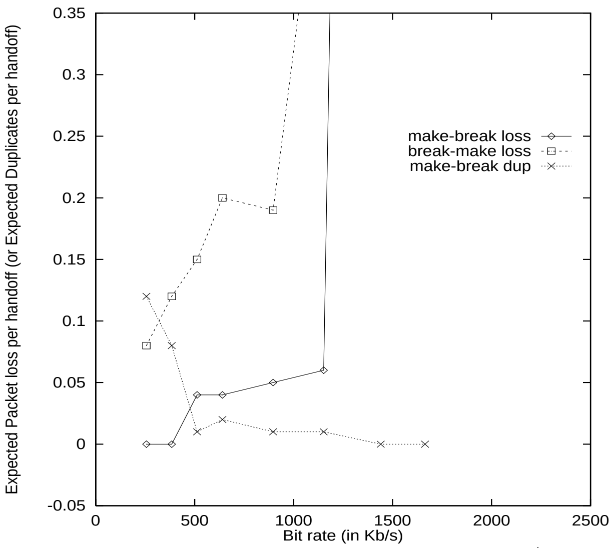

Figure 8: Disruption due to Schemes 1 and 2 (video, downlink, configuration I)

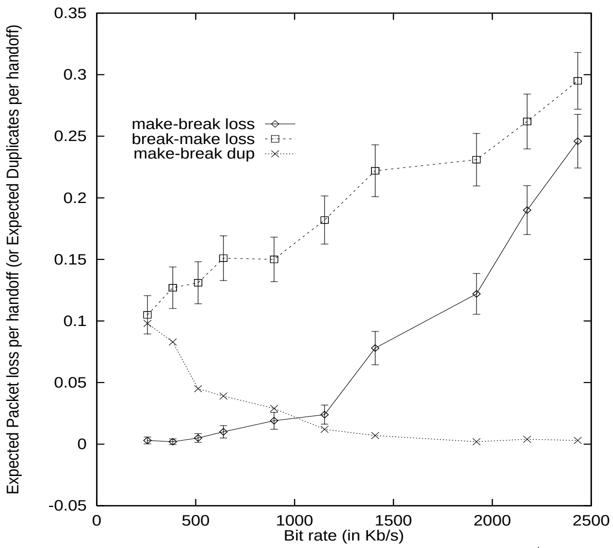

Figure 9: Disruption due to Schemes 1 and 2 (video, downlink, configuration II)

bytes) involved in audio traffic.

Figure 8 shows the effect of the two handoff schemes on video traffic using configuration I while Figure 9 does the same for configuration II. The bit rate of video traffic in Figure 8 varies from 256 kb/s to 1.6 Mb/s from left to right on the x-axis. As expected, the number of duplicates decrease and the number of packet losses increase with increase in the video bit rate in the case of the *make-break* scheme. Overall, the *make-break* scheme performs better than the *break-make* scheme. For example, the *make-break* scheme results in *negligible loss* (less than 0.05 packets per handoff) for video bit rates of up to 1Mb/s ($P = 3750$ bytes) while the *break-make* scheme results in a loss of about 0.2 packets per handoff. Furthermore, a single handoff never resulted in the loss of more than one video packet. Thus, a loss of 0.05 packets per handoff translates to loss of one packet every 20 handoffs. In the case of higher bit rate video traffic, the losses increase drastically in both the schemes. These losses are because of saturation in the total available user throughput of our wireless link (1.5Mb/s) and are not a result of connection rerouting.

When the data rate is below 1Mb/s, the graph in Figure 9 is almost identical to the graph presented in Figure 8 using configuration I, except that losses are slightly higher in the latter case. The higher losses are again due to occasional errors in the wireless media. Configuration II, besides isolating losses due to connection rerouting, also allows us to study the behavior of the schemes when the data rate exceeds 1Mb/s. In the case of higher bit rate video, we expect losses in the *break-make* and *make-break* schemes to be similar since, as explained earlier, packet duplication decreases significantly. Figure 9 corroborates this. For example, for 2.5Mb/s video ($P = 9120$ bytes) traffic, the *make-break* scheme results in negligible duplication and a loss of about 0.24 packets per handoff. The *break-make* scheme suffers a slightly higher loss of about 0.28 packets per handoff for 2.5Mb/s video because, in addition to the reasons for loss in the *make-break* scheme, the scheme also suffers packet loss when the switch does not have any translation table entries (during the vulnerability period).

These results indicate that the *make-break* scheme should be preferred for low bit rate video traffic. We also conjecture that dividing larger video packets into several smaller packets and transmitting these smaller packets every 30 milliseconds will significantly reduce the number of bytes lost per handoff in the *make-break* scheme for high bit rate video traffic. We are still investigating the effects of this optimization.

6.2 Uplink

In this set of experiments, the mobile host in Figure 5 transmits a 160 byte audio packet every T milliseconds. The base-station receives the UDP/IP packet and transmits the packet to the fixed host using AAL5.

Again, we calculate the expected loss per handoff for different values of T. Figure 10 shows the performance of the two handoff schemes for uplink audio traffic using configuration II (the results for configuration I are similar). In this case, there is significant gain in employing a *make-break* scheme over a *break-make* scheme for high quality audio traffic. For CD quality audio ($T = 10$ milliseconds), about 2 packets are lost during a *break-make* handoff while there is negligible loss for a *make-break* handoff.

The high loss rate for the *break-make* scheme in the uplink direction as compared to the downlink direction can be explained as follows: though the vulnerability period is only approximately 4 milliseconds, the handoff latency is approximately 45 milliseconds (See Table 1). Furthermore , our experiments indicated that the time between the deletion of the translation entry in the cross-over switch (by the *break* operation) to the time the handoff response is sent is approximately 20 milliseconds. Note that the mo-

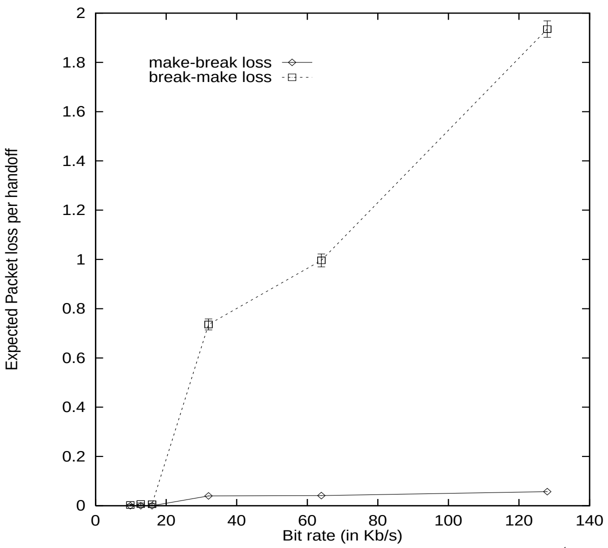

Figure 10: Disruption due to Schemes 1 and 2 (audio, uplink, configuration II)

bile host is transmitting to the old base-station until the mobile host receives a handoff response. Cells received by the cross-over switch during this period are dropped as the entry to route these packets to the source has already been deleted. For CD quality audio (128 Kb/s), packets are sent every 10 milliseconds and thus every handoff results in the loss of approximately 2 packets.

In the *make-break* scheme in uplink direction, the switch has two entries (corresponding to the old and new base-stations) active for a short duration for packets traveling towards the fixed host. As noted before, the mobile host transmits to the old base-station until the handoff process completes. During handoff, all cells thus eventually reach the fixed host. Since the switch does not replicate cells (as compared to the downlink case), there is no possibility of duplicate packets being received by the fixed host. However, there is a possibility of packet reordering in the uplink direction if the following two events occur simultaneously: a) a packet received at the switch from the new base-station (after the handoff response) makes it through the cross-over switch before a packet sent earlier to the old base-station and b) the switch has not deleted the old entry. The former event can happen only if the old base-station is experiencing congestion while the new base-station is lightly loaded; the latter can happen if the switch takes a long time before processing modifications to the translation table (for example, due to a backlog of switch management requests).

This orchestrated set of events is very unlikely in our schemes and no out-of-order packets were observed in our experiments. Note that there is no possibility of reordering in the downlink direction since the mobile host receives data packets from only one base-station at a time and base-stations process data on a first come first serve basis.

The results of video in the uplink direction are very similar to those for the downlink direction (except that the loss rates are higher; we have omitted the graphs for this case). In conclusion, our results indicate that having two different paths to the source active for a small duration of time during the handoff process is quite beneficial in reducing the disruption of uplink traffic.

7 Performance of Chaining (Scheme 3)

We now examine the chaining handoff scheme. Besides its simplicity, this scheme is attractive because there is no loss or duplication during the chaining process itself. In addition, the handoff can be accomplished quickly (6.5 milliseconds in our implementation) if we preassign Virtual Channels (VC) between neighboring base-stations for handoff purposes. However, as discussed before, long "chains" will likely result in performance degradation for the connection, and inefficient routing in the wired network.

In [9], the authors propose loop-detection and elimination as an optimization that reduces network inefficiencies. Note that unlike the basic chaining procedure itself, loop-elimination can result in packet loss without sophisticated synchronization and buffering schemes. To evaluate the magnitude of this loss during loop elimination, we performed the following experiment: the fixed host transmits 160 byte audio packets every T milliseconds. The mobile host is handed off from one base-station to the other and then back again. Since we are using chaining for connection rerouting, such a pair of handoffs will result in the formation of a loop. When we detect a loop, we eliminate it by simply tearing down the forwarding entry at the current base-station and the translation entry at the other base-station.

We experienced an expected loss of about 0.34 packets for each handoff involving loop elimination for CD quality audio (128 kb/s). Note that the process of eliminating loops is similar to that of the *break-make* scheme in that both these procedures involve "breaking" one route (the loop) and "making" another (forwarding to the mobile user). Thus, the process of eliminating loops results in a performance degradation similar to the *break-make* scheme. Based on this experiment, and the fact that loop elimination occurs infrequently, we conclude that loop elimination causes little disruption even when performed without sophisticated synchronization schemes.

8 Performance of Chaining with break-make and make-break (Schemes 4 and 5)

In the following set of experiments, we examine the performance of chained *break-make* and *make-break* schemes for audio traffic in the downlink direction. A fixed host again serves as the audio source. Upon receiving a handoff request from the mobile host, the new base-station first executes the procedures for the chaining scheme. The new base-station then transmits the handoff response message

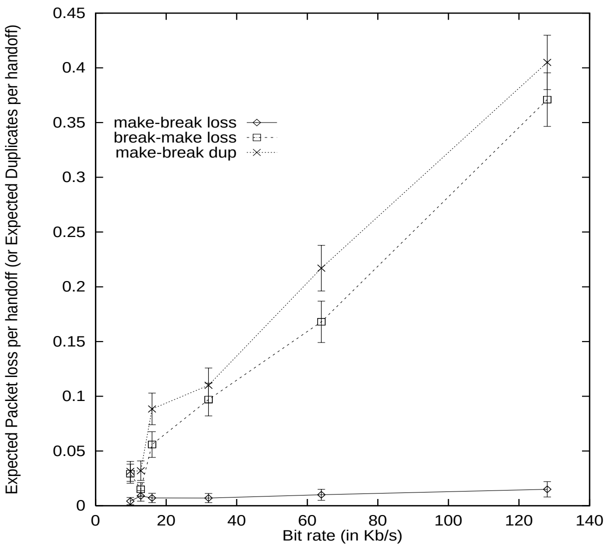

Figure 11: Disruption due to Schemes 4 and 5 (audio, downlink, configuration II)

to the mobile host so that the mobile host now starts listening and transmitting via the new base-station. The new base-station then initiates the *break-make* or *make-break* rerouting procedure. In our experiments, we repeat this entire set of operations and obtain values of packet loss for different values of T, the packetization interval of the audio source.

Figure 11 shows the impact of performing *make-break* and *break-make* immediately after chaining has been completed in configuration II. Even though the rerouting operation is now being performed for a longer connection length due to the earlier chaining operation, the graph in Figure 11 is very similar to Figure 7, which showed the performance of the two handoff schemes without chaining. Further, experiments with audio on uplink and video traffic also revealed performances very close to their unchained counterparts, Scheme 1 and Scheme 2. We conclude from these experiments that performing the chaining scheme first does not significantly alter the losses seen by an immediate rerouting operation, given that the chain is short. Thus, Schemes 5 and 2 (and similarly Schemes 4 and 1) suffer similar degradations while Scheme 5 has the additional flexibility (for example, the ability to effect a reroute for efficiency reasons) and very small handoff latency as compared to Scheme 2 (from Table 1, 6.5 milliseconds versus 37.7 milliseconds).

In conclusion, Schemes 4 and 5 combine the advantages of both optimistic and chaining schemes; they result in fast handoffs so that the mobile host is quickly connected to the new base-station during handoff. Furthermore, they can later employ an optimistic scheme, as needed, in order to make more effective use of bandwidth as well as to minimize disruption. These schemes are useful in cases when a user is handed off in a network that is lightly loaded or when the mobile user does not travel far during a connection. In such cases, the handoff performed using chaining does not disrupt the communication and since the network is lightly loaded, there will be no noticeable performance degradation due to the increased hop count. If the network becomes congested, or if the user moves far enough so that the effects of the extended chain are undesirable, either the *make-break* or *break-make* scheme can be applied to reroute the connection when convenient.

The choice between schemes 4 and 5 depends on the same factors that determine the choice between schemes 1 and 2, viz., whether the application can handle duplicate packets or not. For example, in the case of an audio application such as *vat* which handles duplicate packets, Scheme 5 would be the most appropriate choice, resulting in fast handoffs, negligible loss and efficient use of wired network resources.

9 Discussion

The first issue that needs to be addressed is whether a testbed consisting of a single ATM switch is realistic. We examine the impact of connection rerouting involving multiple ATM switches below. We first discuss the impact on loss and then discuss the impact on latency.

Recall that the vulnerability period in the cross-over switch has the biggest impact on packet loss. Even for connection reroutes spanning multiple ATM switches, the single cross-over switch at which the translation table entries are altered will be mainly responsible for packet loss. This is because the rerouting schemes first update all the switches in the new route and only then alter the entry at the cross-over switch (see Section 3.1). Schemes 4 and 5 capture some aspects of having two switches (one ATM switch and one base-station which acts as a simple switch) in the testbed. Moreover, the degradation suffered under Schemes 4 and 5 is not significantly different from that suffered under Schemes 1 and 2 which involve rerouting a connection of hop length 1. Thus, we conclude that the losses suffered due to multiple switch rerouting will not be significantly higher for the low-bit rate continuous media applications considered in this paper.

Regarding the impact of connection rerouting involving multiple ATM switches on handoff latency, we note that the handoff latency in Schemes 4 and 5 will not be affected since latency is determined only by the chaining of the neighboring base-stations. On the other hand, in Schemes 1 and 2, the handoff latency is directly proportional to the number of ATM switches that need to be updated along the new route. Thus, the separation of connection rerouting from the real-time phase in Schemes 4 and 5 results in a low handoff latency regardless of the number of switches involved in the rerouting operation.

Another issue is the number of reroute related updates or modifications that occur at the ATM switches. Note that in the chaining scheme, we can provision bandwidth between neighboring base-stations and thereby avoid modifying the switch routing entries. Thus, there is a clear trade-off between the amount of bandwidth provisioned and the number of reroute updates. In [13], we show through analysis how to determine the amount of provisioned bandwidth based on the switch processing capacity. Thus, the amount of provisioned bandwidth can be used as a tunable parameter for engineering network resources.

Provisioning bandwidth between neighboring base-stations also helps in recovering from overload conditions on the wired links. Several adaptive applications such as audio and video might prefer lower bandwidth for short periods of time instead of being forcefully terminated. By sharing the available provisioned bandwidth until a suitable new route is evaluated, we can trade-off degraded quality of service for a lower forced termination probability.

While connection rerouting due to handoffs is similar to rerouting due to the failure of network components, there are two important differences. First, handoffs are much more frequent than network faults. With frequent reroutes, the disruption caused to on-going connections has be minimized. On the other hand, in many cases, applications will be willing to tolerate some disruption due to rare network fault rerouting scenarios. Second, handoffs result in connection reroutes that are limited to a small geographic locality (example, neighboring base-stations). On the other hand, reroutes due to failures may involve re-establishing the entire connection.

There is some concern that errors during handoff might be higher than normal, affecting the quality of the wireless transmission media during the handoff process. We note that Scheme 5 is ideally suited to tackle this issue. By chaining, we incur zero loss in rerouting. Thus, we may be able to provide adequate quality to the mobile user during the handoff process itself and postpone the wired rerouting to a future point when the wireless media errors are less pronounced.

Another important issue is the case of non-overlapping transmission cells. We believe that Scheme 5 with the addition of simple buffering at the base-stations is also ideally suited for this situation. The current base-station can buffer the packets most recently transmitted to the mobile host. Upon receiving a forward request, it will resend these buffered packets and continue to forward other packets it receives as before. Thus, the few packets that the mobile host might have missed due to non-overlapping cells would be retransmitted to the mobile host. A similar solution has recently been proposed for handoff in IP networks [2].

10 Conclusion

In this paper we have proposed and examined five different connection rerouting schemes in ATM-based wireless networks. We began by classifying the different handoff approaches studied to date and identifying five simple and easily implementable schemes for our experimental study.

Of the five schemes studied, Scheme 5, which involved a real-time handoff using the chaining scheme and, if necessary, a non-real-time reroute using the *make-break* scheme, seems ideally suited for performing connection rerouting. The separation of the real-time nature of handoffs and efficient route identification in this scheme allows us to perform handoffs quickly and at the same time, maintain efficient routes in the fixed part of the network.

Further, as illustrated from the experiments performed with audio and low-bit rate video, the disruption to the continuous media traffic is quite minimal. Finally, by exploiting the centralized route management in an earlier proposed connection management architecture, the issues of cross-over switch discovery and maintaining the QoS measures for a connection across handoffs are easily addressed. In conclusion, our results indicate that simple rerouting schemes coupled with a connection management architecture are sufficient for supporting continuous media applications over ATM-based wireless networks.

References

[1] A.S. Acampora and M. Nagshineh, "An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks," IEEE Journal on Selected Areas in Communications, October 1994.

[2] R. Caceres and V. Padmanabhan, "Fast and Scalable Handoffs for Wireless Internetworks," in ACM Mobicom'96.

[3] K.Y.Eng, et al, "BAHAMA: A Broadband Ad-Hoc Wireless ATM Local Area Network," ACM/Baltzer Journal on Wireless Networks, Vol 1., No. 2, 1995.

[4] Fore Systems, "SPANS UNI: Simple Protocol for ATM Network Signaling", 1995.

[5] R. Ghai and S. Singh, "An Architecture and Communication Protocol for Picocellular Networks," IEEE Personal Communications, Vol 1., No. 3, 1994.

[6] V. Jacobson and S. McCanne, "Visual Audio Tool: vat," Lawrence Berkeley Laboratory, on-line software: ftp://ftp.ee.lbl.gov/conferencing/vat.

[7] K. Keeton, B.A. Mah, S. Seshan, R.H. Katz, D. Ferrari, "Providing Connection-Oriented Network Services to Mobile Hosts," USENIX Symposium on Mobile and Location-Independent Computing, 1993.

[8] K. Lee, "Adaptive Network Support for Mobile Multimedia," in ACM Mobicom'95.

[9] P.P. Mishra and M.B. Srivastava, "Network Protocols for Wireless Multimedia Access," In Proceedings of First International Workshop on Multimedia Information Systems, pp. 17-24, September 1995.

[10] T. F. La Porta, M. Veeraraghavan, P. Treventi and R. Ramjee, "Distributed Call Processing for Personal Communication Services," IEEE Communications Magazine, June 1995.

[11] J. Porter and A. Hopper, "An ATM based protocol for Wireless LANs," ORL Technical Report 94.2, Olivetti Research Limited.

[12] R. Ramjee, D. Towsley, and R. Nagarajan, "On Optimal Call Admission Control in Cellular Networks," in Wireless Networks Journal, Vol 3., No. 1, 1997.

[13] R. Ramjee, "Supporting Connection Mobility in Wireless Networks," Ph.D. Dissertation, University

of Massachusetts, Amherst, May 1997.
[14] C.K. Toh, "The Design and Implementation of a hybrid handover protocol for Multi-media Wireless LANs," in ACM Mobicom'95.
[15] M. Veeraraghavan, T.F. La Porta and R. Ramjee, "A Distributed Control Strategy for Wireless ATM Networks," in Wireless Networks Journal, October 1995.