An Agent Based Signaling Architecture for Supporting Mobility in Radio ATM Networks

Subir Kumar Biswas[†*]
[†]Computer Laboratory
University of Cambridge, UK.

Andy Hopper[‡†]
[‡]Olivetti Research Limited
Cambridge, UK.

Abstract

This paper presents a signaling architecture for supporting mobility in radio ATM networks. A new concept of mobile software agent, known as representative*, is used for insulating fixed network entities from the effects of user mobility. It is shown that depending on the physical locations of the mobile terminals, their representatives can be used for distributing the mobility management load within the fixed backbone network. The paper describes a location and a representative management scheme followed by a novel connection caching strategy which is used for implementing a family of low-latency and scalable ATM connection handovers. A prototype implementation of the proposed architecture and the associated experimental results are also presented for demonstrating the feasibility and performance of this agent based signaling scheme.*

Keywords: ATM, Wireless Networking, User Mobility, Signaling Protocol, Handover, Location Management, Multimedia Communication.

*Present contact address: NEC USA Inc., 4 Independence Way, Princeton NJ 08540. E-mail: skb@ccrl.nj.nec.com

1 Introduction

The rapidly advancing wireless communication technology promises ubiquitous computing facilities [1] where a mobile user can have location independent access to a vast amount of information and computing resources. Multimedia networking is another emerging technological trend of the 1990s and there exists an increasing demand for supporting multimedia traffic in wide and local area wireless network environment [2]. By now it has began to be understood that in order to satisfy variable quality of service (QoS) requirements of the multimedia streams, connection oriented approaches [3] like Asynchronous Transfer Mode (ATM) are more suitable compared to the conventional datagram based networking. However, design of an ATM wireless network requires to overcome some serious hurdles imposed by the mobility of the roaming network users. In addition to a complex connection routing policy, the end-to-end connectivity needs to be maintained during the relocation of its end users; this involves online rerouting and transfer of the connection state information from one wired network entry-point to another and this process is known as *handover* [2, 4].

This paper presents a *mobile ATM* signaling architecture for supporting connection oriented multimedia streams in a radio network environment. The present architecture attempts to achieve the following design goals. First, it has been decided that user mobility should be isolated from fixed network entities as much as possible; this would enable the existing fixed ATM protocols and applications to coexist with their wireless counterparts without any major modifications. Second, no major assumptions should be made about the characteristics of the backbone network so that the wireless services can be smoothly deployed within a broad range of existing wired ATM networks. Finally, mobility processing load should be uniformly distributed over the network and, if possible, should be dynamically re-distributed following the mobility of the roaming users.

In the present framework, we attempt to achieve the mentioned design goals by introducing the concept of a per-mobile software entity, known as *mobile representative*. A representative acts as an intermediate agent between its client mobile terminal and the other network entities. While isolating mobility from the non-mobile entities, the representatives also enable a network to distribute mobility processing load by dynamically migrating themselves following their client mobile users. An integrated management scheme for handling representatives and mobile user location information is presented. An ATM handover is dealt with by a novel connection caching scheme where a bank of connection *segments* are dynamically created and destroyed following the mobility pattern of an end user. During a handover, instead of creating a new end-to-end connection those segments are appropriately switched in and out for maintaining the desired connectivity. Finally, a prototype implementation of the architecture and a set of experimental results for its feasibility study are presented in this paper.

The remainder of this paper is organized as follows. Section 2 outlines the network model and a brief description of the major wireless services necessary for supporting mobility. The concept of mobile representative is introduced in Section 3. Section 4 presents an outline of the integrated location and representative management procedures. Mobile ATM signaling related issues, including handover, are addressed in Section 5. A prototype implementation of the proposed architecture and some experimental results are furnished in Section 6. Finally, Section 7 concludes the paper with a summary of this work.

2 Network Model and Services

The network is composed of a broadband ATM *backbone* and a wireless (radio) interconnect. The wireless part is partitioned into multiple *transmission cells*, with a control point, known as a *base station* (BS), in each of those cells. The base stations act as gateways between the radio and the ATM backbone. The relocatable terminals, known as *mobile units* (MUs), communicate with other network entities through the base stations and the backbone network[1].

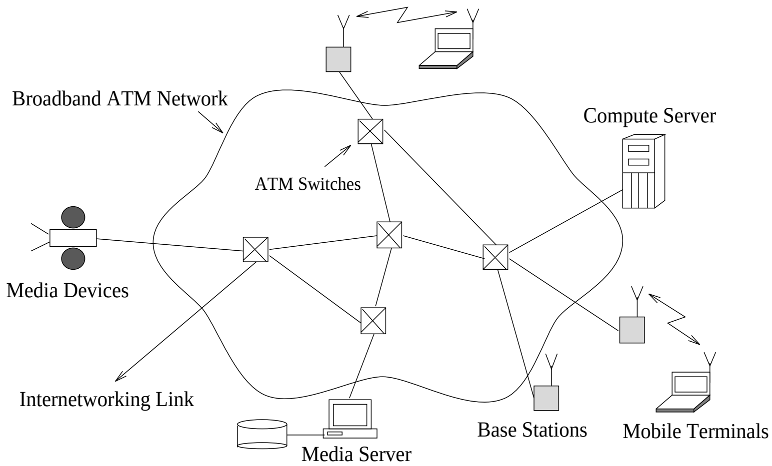

Figure 1: Wireless ATM networking environment

The wireless network consists of large number of very small transmission regions which are referred to as pico-cells. The spatial as well as temporal profiles of the pico-cells within a building are highly irregular and unpredicatable because of the complicated terrain and attenuation characteristics [11] of the building materials. To mitigate any unreliability, caused by the irregular pico-cell characteristics, a high degree of transmission cell overlapping is introduced where an MU is likely to be seen simultaneously by multiple base stations. This reduces the possibility of black-outs (complete network disconnection caused by deep radio fades) in the wireless part of the network. It is shown in Section 5.4 that this *pico-cell overlapping* is also used as a key component for designing low-latency connection handovers.

A layered representation of the major wireless network functionalities is provided in Figure 2. The mobility support services are functionally split into two independent partitions, namely, a mobility signaling module and a location/representative management module. Because

[1]Although it is possible for two neighbor MUs to communicate directly over radio, in order to reduce the higher layer design complexity such MU-to-MU traffic is always directed through the backbone network.

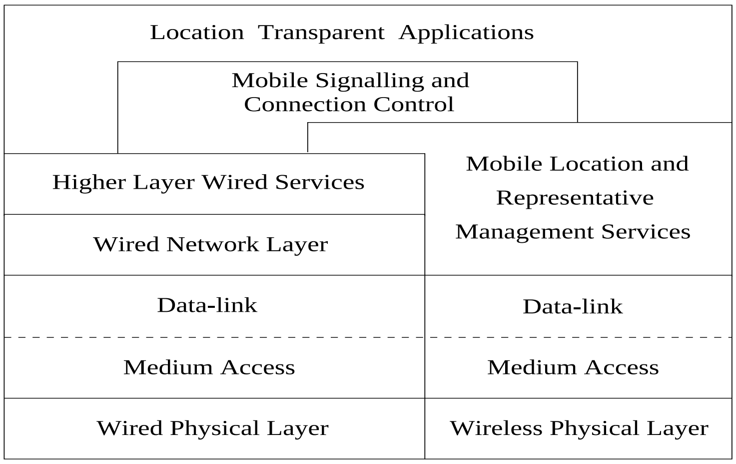

Figure 2: Layered representation of the network services

of the connection oriented nature of the network, most of the location service functions are required to be kept isolated from the data and the signaling services. Although the signaling module is expected to be the most frequent user of the location services (see Section 5), all the higher layer entities, including user applications, can access the latter for various location related queries. Note that the present mobility management architecture does not assume any specific medium access and data-link layer protocol and thus it could be implemented on top of a wide range of lower layer protocols (e.g. [12, 13]) which support the mentioned *cell overlapping* feature.

3 Mobile Representative

While setting up a connection to an MU, the source and intermediate network entities require to know about the location of the end mobile unit. Once the connection is setup, any change of end-MU location information needs to be disseminated so that the connection can be maintained. This may require a large number of network entities to be aware of the changing location of the mobile unit. Our approach for limiting this location information distribution is to provide a control and data interface between a mobile entity and the rest of the network. Such a software interface, known as *mobile representative* (MR), is employed for partial isolation of mobility from the fixed network. A user, interested in connecting to an MU, sends the connection request to the mobile's representative process (see Figure 6) and the representative completes the connection to its client mobile unit. This connection setup mechanism requires the source to locate the representative and the representative to keep an up-to-date location information about its client MU. These location and representative management protocols are presented later in Section 4.

After a connection is setup through the representative of a mobile, when the MU moves between base stations (e.g. from BS_1 to BS_2 in Figure 3), only the parts of the connection from the mobile terminal to its representative are affected. The rest of it remains unaltered as long as the representative process stays stationary. Since a representative is employed for each MU, by choosing appropriate locations for the MR processes the mobility management load can easily be distributed according to the physical locations of the roaming users.

Also, by preassigning a representative which acts like a handover anchor switch [14, 15], this scheme eliminates any need for finding the anchor point during the actual handover process. It requires only a query to the location server, that too during connection setup, for finding out the current location of the representative process. When compared to the handover anchor switch selection protocol in [15], this feature of the representative can be viewed as an efficient means for reducing the handover delay and complexity.

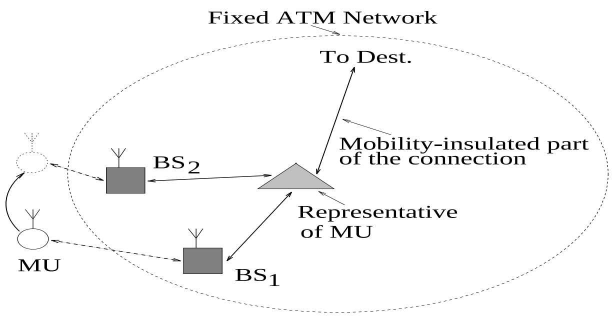

Figure 3: Use of *representative* for mobility insulation

Placement and mobility of these representatives are two crucial design issues. One extreme approach is to permanently nail down a representative process on a stationary entity like a workstation or an ATM switch. Although it makes mobility completely transparent to the fixed network, the scheme suffers from poor utilization of network resources caused by non-optimal connection routing. In the other extreme approach, an MR closely follows its client MU and usually resides within its geographically nearest possible stationary entity. Because of high MR mobility, while yielding a better resource utilization this approach fails to ensure the originally intended *mobility isolation* feature of the representatives. An intermediate approach is adopted in the current implementation [16] where an MR is programmed to be much less mobile than its client MU and a representative migration takes place only when its client MU relocates to a new geographical or administrative domain. More about the representative management functions, including migration protocols, are described in Section 4.3.

4 Representative and Location Management

A Home Location Register (HLR) and Visitor Location Register (VLR) based location server framework [17, 18] is augmented for supporting the representative management functions in the present architecture. An outline of the location server and the details of the representative management protocols, integrated with the location manager, are presented in this section. The details of the location update and query protocols/semantics are out of the scope of this paper and can be found in [19].

4.1 Network Partitioning and Addressing

The network is partitioned into a group of *clusters* and each network entity is a member of a specific cluster. Like other non-mobile entities (e.g. BS), each MU is assigned with a globally unique network layer address which is independent of its geographical location. Also, each mobile terminal is permanently registered with a particular cluster and it shares the same cluster address space with all other entities in that cluster. An MU can visit other clusters without changing their original cluster of registration.

The network address space is partitioned into a number of cluster address spaces, each of which is subdivided into a mobile-cluster address space and a fixed-cluster address space. All stationary entities, including the base stations, belong to the fixed-cluster address space and the MUs belong to the mobile-cluster address space. A four octet address space (like IP) is used where a part of an address represents the cluster-id and the remainder of it represents the host-id.

With this addressing policy, it is possible to resolve the cluster address of an entity from its network layer address. The cluster, which an MU is registered with, is known as its *Home Cluster* and the one, which it is visiting currently, is referred to as its *Visiting Cluster*. Similarly, all MUs which are registered with a cluster are designated as its *Domestic MUs* and the ones which are visiting from outside are known as *Visiting MUs* with respect to that cluster. Note that this static mobile addressing scheme enables the existing name services [20] to operate within the present mobile scenario. For identifying the mobile terminals their *names*, instead of *addresses*, will be used in the rest of this paper.

4.2 Location Service Outline

The location server is designed in a distributed fashion where each cluster maintains a location management entity which is known as *Cluster Location Server* (CLS). A CLS maintains a distributed database containing location information about both mobile units and their representatives. Also, each mobile representative is required to keep (see Section 3) an up-to-date version of its client MU's location information which is provided by the CLS of the local cluster. There are two ways to setup a CLS within a cluster. In the first scenario, each CLS is configured with an ATM anycast address [21] and a client uses that address to send a query to the local CLS without explicitly knowing the identity of the corresponding server. The second option, implemented in the present prototype, is to statically configure a machine as CLS and hard-code that information within all the fixed entities within the cluster.

Depending on the spatial granularity, an MU's location information is classified into the following categories. *i*) High Resolution Location Information (HRLI): contains addresses of the base stations which are seeing the relevant mobile unit and *ii*) Low Resolution Location Information (LRLI): points to the clusters which are currently being seen by the MU. Location of a representative, on the other hand, is represented by the address of the host-entity in which the representative process is currently residing.

The following location information is required to be stored within the CLSs and the mobile representatives. The home CLS of an MU maintains the Low Resolution Location Information (LRLI) of the terminal and the visiting cluster location server holds the High Resolution Location Information (HRLI) of the mobile. The location of the representative of a mobile unit is maintained in both the home and visiting CLSs of the mobile terminal. Note that a representative process also maintains high resolution information about the location of its client mobile unit. Since a representative always resides within the visiting cluster of its client MU and the CLS in that cluster maintains the HRLI of the terminal, the information is simply downloaded from the CLS to the MR once the latter is created within the cluster. Any further change in the mobile's location information is subsequently communicated to the MR by the local cluster location server.

When necessary, a client sends its location queries (for both mobile units and their representatives) to the local cluster location server. Upon reception of a query, the CLS first attempts to resolve it by searching the local databases. It should be able to resolve it locally when the mobile in question is visiting the cluster. If the local search fails, it retrieves the necessary location information from the home CLS of the terminal. Various levels of CLS caching have been implemented for optimizing the query performance of the location server. It is to be noted that the degree of location information caching within a client is completely application dependent and it is outside the realm of the location servers' control. Detailed specification of the update and caching protocols can be found in [19].

4.3 Representative Management

Creation and migration are identified as the two main management operations associated with a representative process. Since both of these heavily depend on the mobility of the client mobile terminals, and the location server maintains that mobility information, all the representative management functions are integrated within the distributed location server.

4.3.1 Creation

The cluster, in which a representative MR_j would reside, is selected by the home CLS of MU_j and the particular representative host-entity is selected by the CLS of the chosen cluster. When MU_j is simultaneously seen by multiple clusters (this happens when a terminal roams around cluster boundaries), the home CLS of MU_j uses a combination of the following rules for choosing an appropriate cluster for the creation or relocation of representative MR_j. *i*) Choose the cluster where MU_j was seen most recently. This is useful when not much information is known about MU_j's *mobility profile*[2]. *ii*) Select the cluster where maximum number of base stations are currently being seen by MU_j. This strategy is useful for minimizing the average distance between MR_j and all the base stations which are seen by MU_j (see Figure 3). Once a cluster is selected, CLS of the chosen cluster uses the following criteria to pickup a fixed host for hosting the MR_j process. First, it is necessary for the host to be located in a fairly central location with respect to the cluster base stations. This helps to keep the length of a mobile connection more or less same, while the end-terminal moves around the cluster. In addition, the chosen machine should have sufficient computing and communication resources (e.g. cpu cycles and buffer) left for hosting the representative process.

4.3.2 Migration

As explained in Section 3, an MR process migrates only when its client terminal moves across a cluster boundary. On relocation of a mobile unit MU_j to a new cluster, its home CLS_h may decide to change the host-entity of representative MR_j and, if decided, a set of heuristics is used by the visiting cluster location server for choosing a new host for MR_j. Since each mobile connection passes through the involving MR (see Section 3), an online representative migration would require the MR to move with all the existing connections passing through it. While providing seamlessness, this requires additional complexity of performing *representative handovers* and that is in addition to the standard *base station handovers* of the mobile connections.

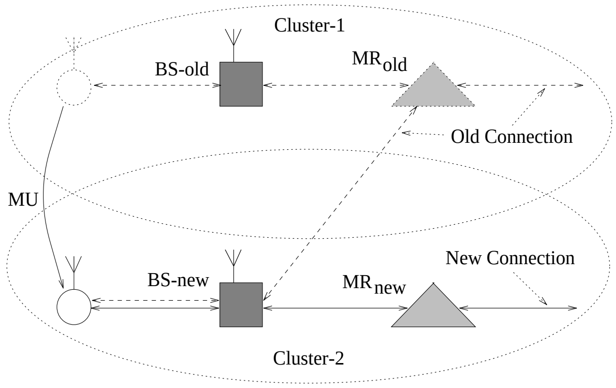

Figure 4: Representative duplication during migration

Instead, we use a *transient representative duplication* mechanism in which a new representative MR_{new} is created (see Figure 4) soon after a migration decision is taken by the relevant home CLS. While all the old connections continue to operate through the old representative MR_{old}, the new connections are setup through MR_{new}. From this point onwards the

[2]Mobility profile is specified [22] as the temporal and spatial locality information about the movements of a terminal.

entire network, including the location servers, considers MR_{new} as the active representative process and MR_{old} becomes defunct except for the old connections. Once all the old connections are closed, MR_{old} terminates itself automatically.

5 Mobility Support Signaling

This section presents the representative assisted signaling protocols which are required for handling ATM connections in a mobile environment.

5.1 Connection Segmentation

A mobile connection can be seen as a set of wired and wireless *segments* which are put together to achieve end-to-end connectivity (see Figure 5). While the wireless segments are single-hop, the number of hops in a wired segment depends on the relative physical locations of the base stations and the host-entities of the end mobile units' representative processes. Any combination of segments which provides an end-to-end connectivity constitutes a *path*. At any point of time there exists only one *active path* which carries traffic through a connection. For example, MU_j-BS_p-MR_j-MR_k-BS_t-MU_k in Figure 5 indicates an *active path* for connection MU_j-MU_k.

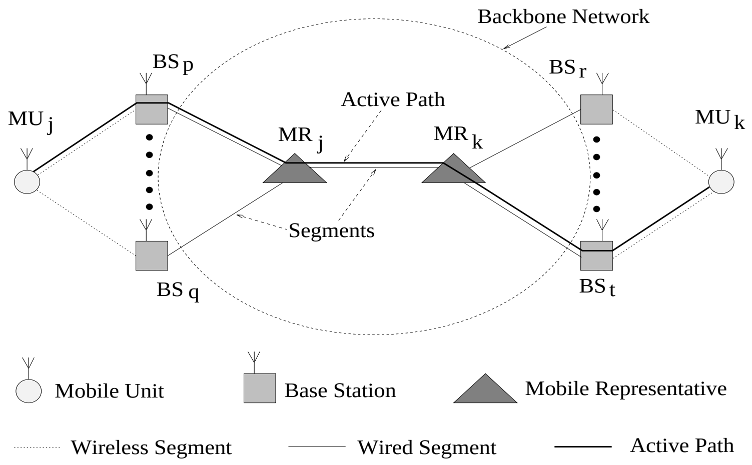

Figure 5: Mobile ATM connection structure

Multiple paths are used for performing handovers. Note that the multiplicity of the edge segments (MU-BS and BS-MR) and the paths in a mobile connection depend on the number of base stations which are simultaneously seen by the end MUs. During a handover, those edge segments are switched in and out of a connection by MUs and their representatives. These switching operations are coordinated according to the inter-base terminal relocation.

5.2 Connection Setup

During the creation of a new mobile connection, only the segments necessary for constructing an *initial path* are setup and data flow starts through it; if necessary, other feasible segments of the connection are created afterwards. An example connection and segment setup procedure is pictured in Figure 6 and the corresponding signaling sequence is outlined in Figure 7.

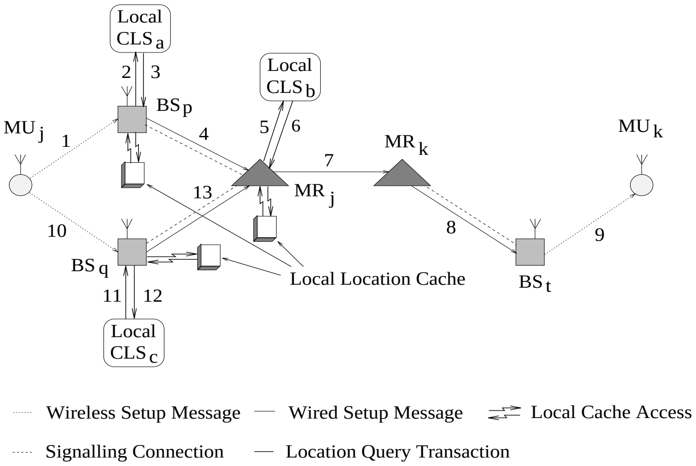

Figure 6: Connection and segment setup

Each connection setup process involves at least one location query to locate the relevant representative process. Two such queries are needed when both the end entities are mobile. Messages '2', '3', '5' and '6' represent such location query transactions during the example setup procedure. To reduce this per-connection-setup query overhead, a local location cache is maintained within each BS and MR (See Figure 6).

A combination of *source* and *hop-by-hop* approach is used for routing a mobile ATM connection. Source routing is implemented in a per-segment basis whereas within a segment, a connection is dynamically routed in a hop-by-hop fashion. This requires base station BS_p, for instance, to put the network layer address of MR_j-host in the connection setup message (labeled '4'), for short term source routing. Segment routing from BS_p to MR_j, however, is hop-by-hop and is dynamically performed by using standard wired-ATM connection routing protocols. Note that the one-hop wireless setup ('1' and '9') and reply messages do not need any explicit routing to be performed.

5.3 Segment Maintenance

While data flow continues through the *initial path*, additional segments are created and destroyed following the end terminal's mobility pattern. The segment, connecting two mobile representatives, remains unaltered during the lifetime of a connection; only the ones with an MU or a BS as one of the end entities are created and terminated dynamically. An off-line segment setup procedure, triggered by MU_j's relocation into BS_q's pico-cell (see Figure 6), is shown in Figure 7; corresponding messages are labeled from '10' through '13'. Other than when a mobile connection is terminated, a segment through base station BS_p is deleted when MU_j goes out of BS_P's pico-cell. A mobile connection may close-down either because of an explicit termination by an end user or due to a fault developed during an MU relocation. In either case, all associated segments need to be cleared while closing the connection.

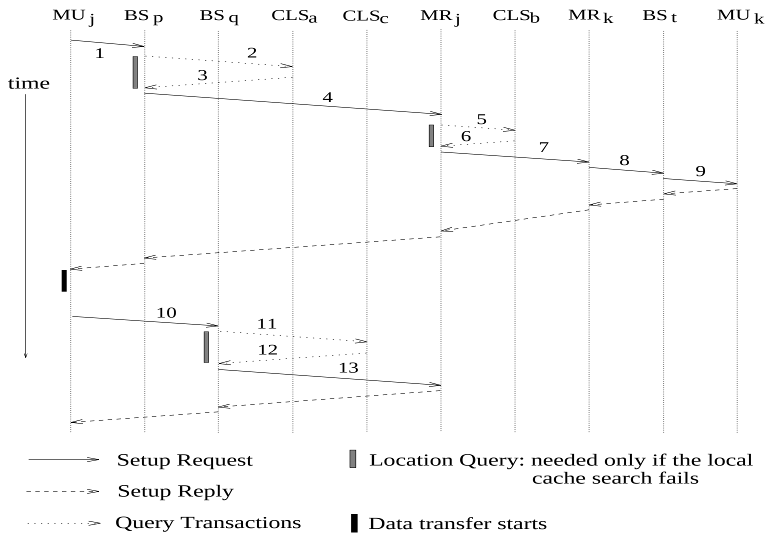

Figure 7: Signaling for connection and segment setup

5.4 Handover Signaling

In order to reduce the handover latency, a segment caching scheme is introduced where the necessary segments of a connection are created before a handover takes place. Once the handover is initiated, the relevant segments are switched in and out of the connection to maintain an end-to-end *path*. Although the present network layer handover protocol attempts to prevent cell loss and out-of-ordering, currently no mechanism is provided for recovery from those error situations. Depending on the specific QoS requirements of the connections, the higher layer protocols will have to recover from these. A carefully designed signaling sequence is used for reducing cell loss and out-of-ordering probability during the handovers.

For expediting the handovers, signaling between a base and a representative is carried out through a pre-established signaling connection. If there exists at least one mobile connection segment between a BS and an MR, the BS is required to create and maintain a handover signaling connection with the MR.

Sudden failure of a radio link, constituting a segment of the *active path* of a mobile connection, triggers a *without hint* handover [2]. On the other hand, when quality of the radio link between an MU and a BS degrades gradually, the MU may like to handover its connections to another BS with better radio link with it. The latter is referred to as a *with hint* handover.

5.4.1 With Hint

Segment switching and the corresponding signaling sequence during a *with hint* handover, initiated by the mobile MU_j, is pictured in Figure 8. It is assumed that the current *active path* is passing through BS_p and segments *j* and *l*, involving BS_q, are already setup during an off-line segment setup procedure. A gradual and consistent improvement of the radio link quality of segment *j*, compared to that of *i*, triggers MU_j to initiate the handover.

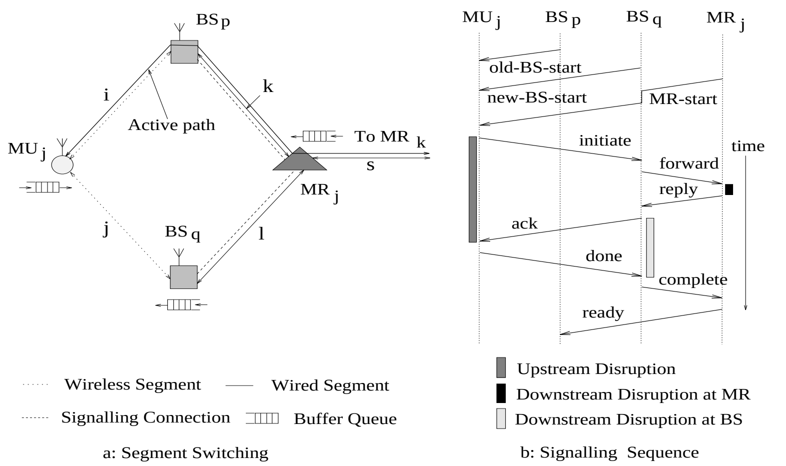

Figure 8: Signaling for a *with hint* handover

Immediately after MU_j decides to initiate a handover it stops sending cells through segment *i* and puts the cells in a temporary buffer; however, it continues cell reception through segment *i*. MU_j also sends an *initiate* signaling message to the newly chosen base BS_q. When BS_q receives this message it puts a buffer in between segments *l* and *j*. Until an explicit *done* signal is received, data stored in this buffer is not sent through segment *j*. Once the buffer is set, BS_q sends a *forward* signal to representative MR_j through a previously established handover signaling connection. Upon reception of the *forward* signal, MR_j breaks the existing path by disconnecting segments *s* and *k* and constructs a new one by connecting segments *l* and *s*. Immediately after this segment switching is over, MR_j replies to BS_q with a *reply* signal. BS_q simply forwards this to MU_j as an *ack* message. BS_q is currently buffering all cells received through segment *l* but forwarding the cells (if there is any) received through segment *j*. On reception of the *ack* signal, MU_j first stops receiving cells through segment *i*; then it clears the buffered cells by sending them through segment

j and starts transmitting and receiving regularly through this segment. MU_j declares the completion of handover operation by sending a *done* signal to BS_q. When BS_q receives this *done* message it clears the queued cells through segment j and starts regular cell forwarding from segment l to j. BS_q also sends a *complete* message to the representative MR_j, informing the latter about a successful completion of the handover procedure. Finally, MR_j sends a *ready* message to the old base station BS_p so that the base station, depending on the quality of its radio link with MU_j, can either delete segments i and k or it can prepare for future handovers.

When a *with hint* handover has to be initiated because of network load balancing or quality of service readjustment [23, 24], it is more likely that a wired network entity, instead of the mobile terminal, would initiate it. In such situations, messages *old-BS-start*, *new-BS-start* or *MR-start* are used by the respective entities in order to instruct the end mobile unit to perform a handover, as described above.

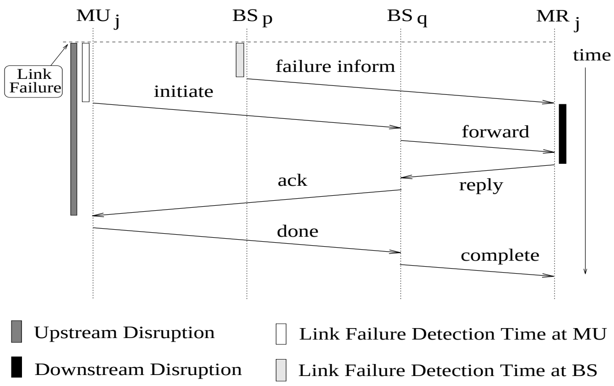

Figure 9: Signaling for a *without hint* handover

5.4.2 Without Hint

The following additional signaling steps are necessary for handling a *without hint* handover which can be triggered by a sudden radio link failure. When a failure of the radio link with MU_j is detected by the base station BS_p, it sends a *failure inform* signal to MR_j (see Figure 9). In addition to this, the base station initiates a deletion procedure for segments i and k. On reception of the *failure inform* signal, MR_j stops forwarding traffic through segment k and the corresponding cells are buffered within a local queue. However, the cells received through segment k are continued to be forwarded through segment s. From this point onwards MR_j acts very much like as in the *with hint* situation. The only difference is that when MR_j receives a *forward* message from BS_q, along with the other usual segment switching activities it flushes all buffered cells and forwards them through segment l.

On the other side, upon detection of the link failure mobile MU_j starts a handover by sending an *initiate* signal to the newly chosen base station BS_q. Unlike for the *with hint* scenario, MU_j stops both transmission and reception through segment i and immediately starts reception on segment j. BS_q handles the *initiate*, *forward*, *reply* and *ack* signals in the same way as described in the *with hint* situation. The only difference to be noted is that BS_q does not have to provide any buffering for MR_j-to-MU_j cells. This is because of the immediate reception switching of MU_j from segment i to segment j, as mentioned in the last step.

During a *without hint* handover, although the newly chosen BS does not contribute towards *downstream*[3] disruption, the MR counterpart of it is quite high because of the link failure detection delay and the *failure inform* message transit time. Upstream latency is also affected by the link failure detection delay. There exists an extreme situation when a radio link on the *active path* fails and no previously setup segment is ready for immediate switching. Considering a very low occurrence probability of such events under the present framework of dense transmission cell overlapping and segment caching, currently no signaling provisions are made for recovering from those.

5.4.3 Performance Issues

Since the handover signaling at both the ends of a mobile ATM connection are completely isolated from each other, the length of the connection does have very little impact on the handover disruption figures; this makes the present handover scheme quite scalable. Also, the representative migration policy (see Section 4.3.2) ensures that an MR is not terminated as long as at least one connection is active through it; that ensures that the handover signaling is never affected by a non-stationary representative process. Note that MU_j, in Figures 8 and 9, exchanges all the handover signaling messages only with the new base station BS_q. Because of a better statistical radio link quality of segment j compared to that of segment i (possibly this is why the handover was initiated), the handover can be executed quite reliably. It is worth noting that the present protocol does not distinguish between an *inter* and an *intra* cluster handover; the only difference, seen by the applications, is increased latency caused by larger geographical separations between the signaling entities in an inter-cluster scenario.

When the mobile terminal MU_j sees both base stations BS_p and BS_q (see Figure 5) with almost equal clarity, there exists a potential problem of oscillatory initiations of handovers which are triggered by intermittent changes of the wireless link quality. This problem has been tackled by using a *skeptic* [25] based initiation strategy where a handover decision is based not only on the current radio link qualities but also on their statistical performance history. A multi-threshold handover initiation mechanism, based on hysteresis, can also be used for alleviating this problem of oscillatory initiation.

[3] Represents the stream from the backbone network to a mobile terminal. The reverse direction is referred to as *upstream*.

6 Prototype Implementation

6.1 System Environment

An ARM-60 RISC processor based ATM switch[4] was used for constructing the ATM backbone of an experimental network. The switches which were connected by 100 Mbits/s co-axial links (running taxi [26] full-duplex link protocol) provided a generic hardware platform for experimenting with the proposed mobility signaling protocols. Due to the unavailability[5] of any working radio link, the co-axial cables were also used for emulating the radio links between the base and the mobile terminals. Also note that the same switches were used for emulating both the base and the mobile entities. A *tunneling* [20] approach was taken for emulating a radio *link layer* over a co-axial cable, connecting two radio ports. Whenever two emulated radio ports are physically connected by a co-axial cable, an ATM connection is created and subsequently maintained over the relevant ports. This ATM connection (tunnel) is used for exchanging radio data-link layer PDUs between the two end entities. The end entities (BS and MU) of an emulated radio link also maintain two state machines which represent the changing quality of the corresponding radio link and this link quality variation is used for triggering appropriate connection handovers. The dynamics of link quality state, including disconnection, is controlled by a pre-loaded model [13] which was obtained from a mobility experiment carried out in an in-building environment. Co-sighting of an MU by multiple base stations was emulated by connecting several BS-switches to the MU-switch over its multiple emulated radio ports. From now onwards, this underlying emulation model will be assumed without any explicit reference to it.

A locally developed threaded micro-kernel, called WANDA [27], was used as a generic software platform for implementing the present signaling architecture. Multi-Service Network Architecture[6] (MSNA) [28] served as an ATM protocol suit for the present backbone and wireless network.

6.2 Mobility Interface and Datapath Modules

A layer of *mobility interface* (see Figure 10) is implemented at the mobile terminal where this interface handles most of the mobility related activities including segment management and handover. Since this layer is in between the applications and the existing MSNA sockets, a user space application remains completely insulated from the mobility related activities. This interfacing approach has enabled us to have a *soft* and quick implementation without any substantial modification of the WANDA kernel code. The routines, constituting this new interface, are listed in Table 1. While most of these routines are direct derivations from Berkeley UNIX socket abstraction, the incoming connections are handled in two slightly

[4] A bus-based 4x4 ATM switch, designed and developed at Olivetti Research Ltd., Cambridge.

[5] Although a radio link was under investigation at Olivetti Research Ltd., it could not be used because of its experimental nature at the time of this implementation.

[6] MSNA was developed in Cambridge Computer Laboratory and an operational MSNA internet currently exists within Olivetti Research Ltd. and Computer Laboratory.

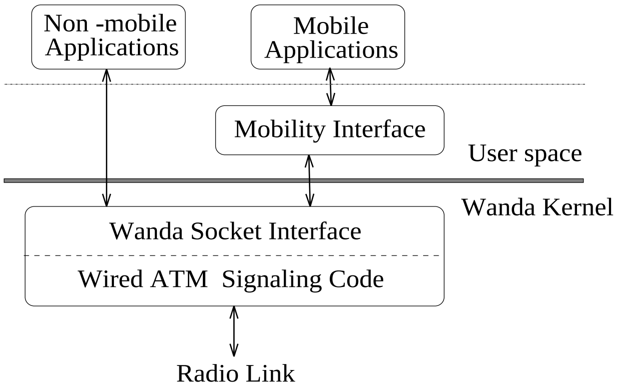

Figure 10: Data and signaling interface at mobile terminal

different ways. In the first approach, an application registers its interest in receiving an incoming connection by invoking the *MobileOffer* routine. When a matching call request arrives, the application thread can check the details of the remote end point including its peer address; based on this information the thread may or may not *Accept* the connection. *MobileListen*, on the other hand, forces an invoking thread to *Accept* a matching incoming connection without any such checking provisions.

buffer: WandaIOBuffer
mobile_socket: MobileConnectionDescriptor
address: WandaIPCAddress
mobile_socket = GetMobileSocket(PROTOCOL_TYPE)
MobileBind(mobile_socket, address)
MobileConnect(mobile_socket, address, timeout)
MobileOffer(mobile_socket, timeout)
MobileAccept(mobile_socket)
MobileListen(mobile_socket, timeout)
MobilePeerName(mobile_socket, &address)
MobileClose(mobile_socket)
MobileSend(mobile_socket, buffer)
buffer = MobileRecv(mobile_socket, timeout)

Table 1: Library routines for mobility interface

An invocation of the *GetMobileSocket* routine at an MU creates a collection of resources which is referred to as a *mobile module* (see Figure 11). A handle to the module, known as a *mobile_socket*, is returned to the invoking thread and the thread uses this *mobile_socket* as a descriptor for the created mobile connection. A mobile module maintains multiple sockets at the network side (see the socket list in Figure 11:a) where each such socket corresponds to a segment of the mobile connection. For each socket, a pair of data *pumping threads* are used for sending and receiving I/O buffers through the relevant connection segment. At any point of time only one segment should have access to the queues and to ensure that, each socket needs to maintain a set of status and other thread control information associated with it. For each mobile module a dedicated manager thread performs the segment and handover management operations, as outlined in Section 5. Any significant change of the radio link status causes the manager thread to be activated; if the link status changes too frequently, the generated events are queued within the *radio link event buffer* and subsequently filtered using the mentioned *skeptic* [25] mechanism.

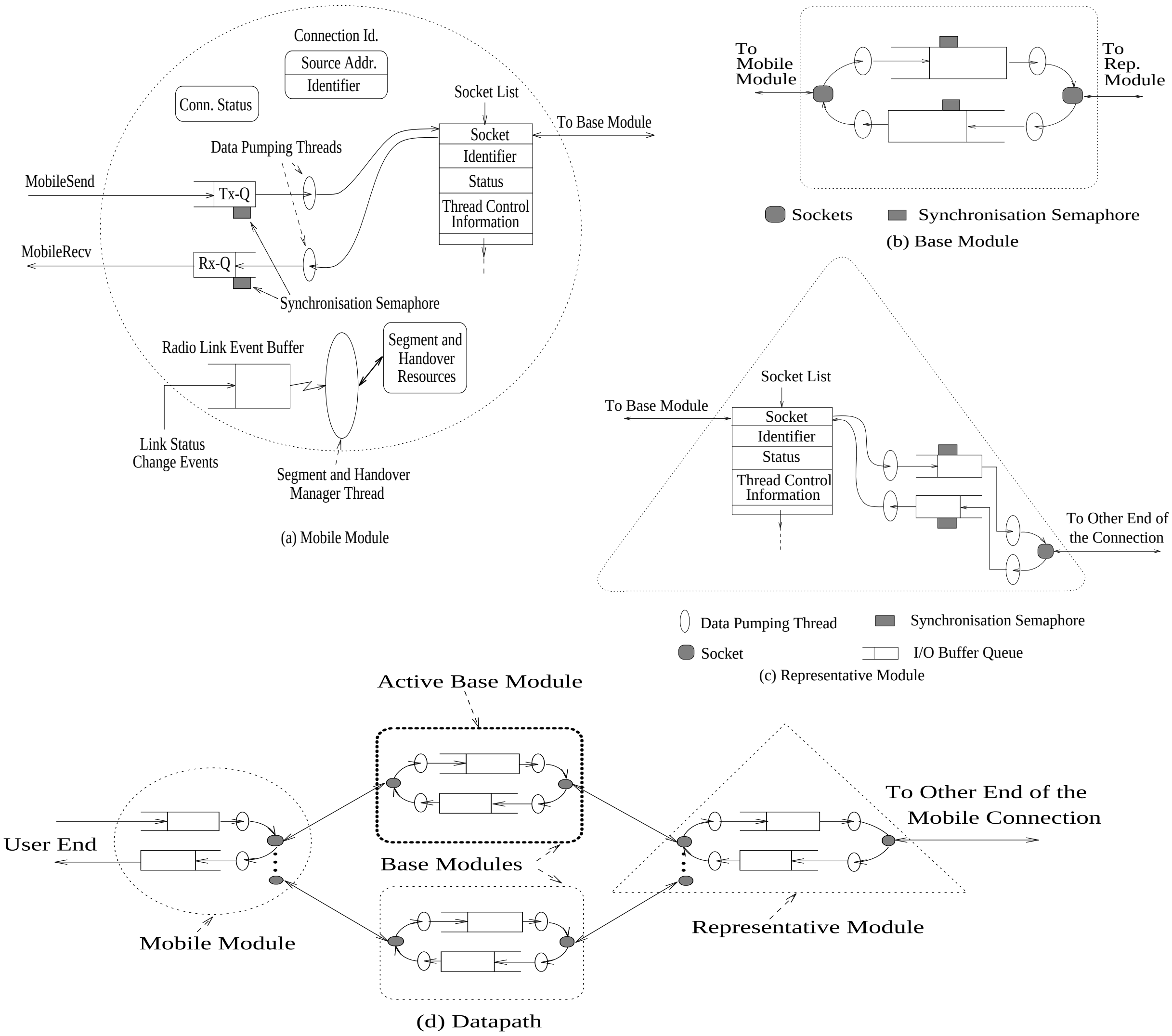

Figure 11: Datapath modules of a mobile connection

When a base station receives a mobile connection or a segment setup request (see Section 5.2), it creates a user space *base module* as shown in Figure 11. Once the necessary segments are established (identified by the two sockets), the received data is forwarded in both directions using two pumping threads; buffer access of these threads are synchronized using two user level WANDA semaphores.

The components of a *representative module*, created during a connection setup, are also shown in Figure 11. Similar to the mobile modules, multiple sockets (implementing segments) can exist simultaneously at the base station side of a representative module. A segment switching involves activation and de-activation of the relevant data pumping threads and the required transient buffering is done at the WANDA I/O buffer queues; these are same as the handover buffers, mentioned in Section 5.4.

A mobile connection, comprising various datapath modules, is also illustrated in Figure 11. Note that at any point of time only one base module is active in transferring data between the mobile module and the representative module. It has been experienced that the deletion of a module needs careful consideration regarding the release of its occupied resources. Since a module operation involves multiple threads, they have to agree about whose responsibility it is to free the resources associated with the relevant user level module. A general approach was taken where the threads in a module exit sequentially after updating their corresponding status and the last survivor thread releases the module occupied resources before exiting.

6.3 Experimental Results

The experiments reported in this section were carried out with the connection topologies as shown in Figure 12. In this single cluster environment a workstation is statically configured as the cluster location server (CLS) and the CLS is connected to the network through an ATM switch. In both the topologies, the mobile representative of the MU is lodged within a switch and the MR remains stationary during the course of these experiments. Two sets of performance measurements are taken. The first one measures signaling and handover performance of the proposed scheme and the second set studies the behavior of various datapath modules.

6.3.1 Signaling and Handover Performance

Mobile connection and segment setup latencies are presented in Table 2. The initial connection-path (see Section 5.1) is setup from the mobile terminal to the destination workstation through base station BS_1, and the corresponding latency is reported as the end-to-end setup delay in Table 2. For the same connection, the combined setup time for segments MU-BS_2 and BS_2-MR is also measured; note that these segments are setup in the presence of an *active path* through BS_1. For each table entry, 1000 such connections and segments are setup with an interval of 1 second and the corresponding average values are computed.

While creating the first connection, base station BS_1 requires to setup a location query

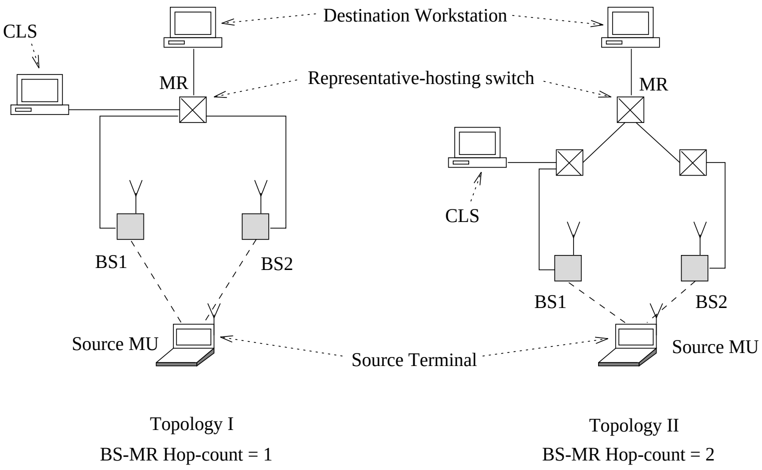

Figure 12: Mobile connections with experimental topologies

connection with the CLS for downloading the location information about the mobile representative. Corresponding setup and location information access delays are also included in Table 2. The last column indicates the time required by the base station to setup a handover signaling connection with the representative-hosting switch. Note that the presented figures for end-to-end connection and segment setup delays do not include any of the BS-CLS connection setup time, location access delay or BS-MR signaling connection setup latency. Since location information caching within base stations is enabled during these experiments, all the location related latencies contribute to the setup delays only for the first instance of connection and segment setup through BS_1 or BS_2; the same applies to the BS-MR signaling connection setup latency.

Topology	End-to-end Setup (ms)	Segment Setup (ms)	BS to CLS Setup (ms)	Location Access (ms)	Signaling Conn. Setup (ms)
I	18.16	12.09	11.98	2.12	9.29
II	23.63	18.13	12.01	2.13	14.02

Table 2: Connection and segment setup latencies

In topology-I when the mobile terminal moves from BS_1 to BS_2, the segments MU-BS_2 and BS_2-MR are created within a time period of 12.09 ms. When compared to the *initial path* setup latency of 18.16 ms, this lower segment setup delay indicates the performance gain of segment setting compared to end-to-end connection setup during handovers. Since the segment setup time depends on the base-to-representative hop-count, this difference (6.07 ms in topology-I) between the connection and the segment setup delays is expected to be more pronounced for connections with larger representative-to-destination hop-count (which is unity in the present setup).

Topology	BS to MR Hop-Count	Upstream at MU (ms)	Downstream at MR (ms)	Downstream at BS (ms)
I	1	6.82	0.73	2.87
II	2	7.01	0.72	2.27

Table 3: Effects of base-to-representative hop-count on handover disruption

In order to study the asymmetric upstream and downstream disruptions during handovers, two bidirectional CBR video streams[7] are studied. For measuring the disruption at the mobile module, two time-stamps are put on each block of data; once during its insertion into the outgoing I/O buffer queue (of mobile module; see Figure 11) and then while taking it out from the same queue. The difference between those two stamps indicates the time spent by a data block within the buffer. In the absence of any handover, this latency is measured initially for 50000 blocks and the maximum value is recorded. After the handovers are activated, any such latency which is larger than the previously recorded maximum is interpreted as the upstream handover disruption. The same procedure is used for measuring the downstream disruptions at the base and the representative modules. The corresponding service disruption figures are presented in Table 3, where each entry corresponds to the mean computed from measurements taken for 5000 *with hint* handovers which were arranged to occur with an average interval of one second.

Buffer Statistics	Upstream at MU (Kbits)	Downstream at MR (Kbits)	Downstream at BS (Kbits)
Mean	2.614	0.276	1.098
Maximum	5.698	0.307	1.495
95th Percentile	4.891	0.277	1.294

Table 4: Handover buffer requirement in various datapath modules

In topology-I, a maximum upstream disruption of 6.82 ms indicates a 62% reduction of latency when compared to the schemes where an end-to-end connection requires to be setup during each handover (see the *full re-establishment* scheme in [2]). The degradation of video quality, caused by this low disruption, was found to be more than acceptable. It is interesting to note that the downstream latencies (both within the base and the representative) do not have any specific dependency on the hop-count between the base and the representative. This can be understood from the handover signaling sequence, illustrated in Figure 8; during a handover no signaling which is responsible for the downstream disruptions takes place between the base and the representative entities. This makes the downstream han-

[7] Each 25 frames/s video stream [29] is comprised of 48x40 octet data blocks, transmitted at an interval of 40 ms. These data blocks represent Multi Service Segmentation and Reassembly (MSSAR) [28] layer PDUs where each such PDU is equivalent to an AAL5 block.

dover latencies independent of the base-to-representative distance. However, the upstream disruption increases with the hop-count between the representative and the relevant base station. In the current setup, a single hop increment (from 1 to 2) causes an additional upstream disruption of 190 μs. Additional signaling through the extra level of switches in topology-II accounts for this small additional delay.

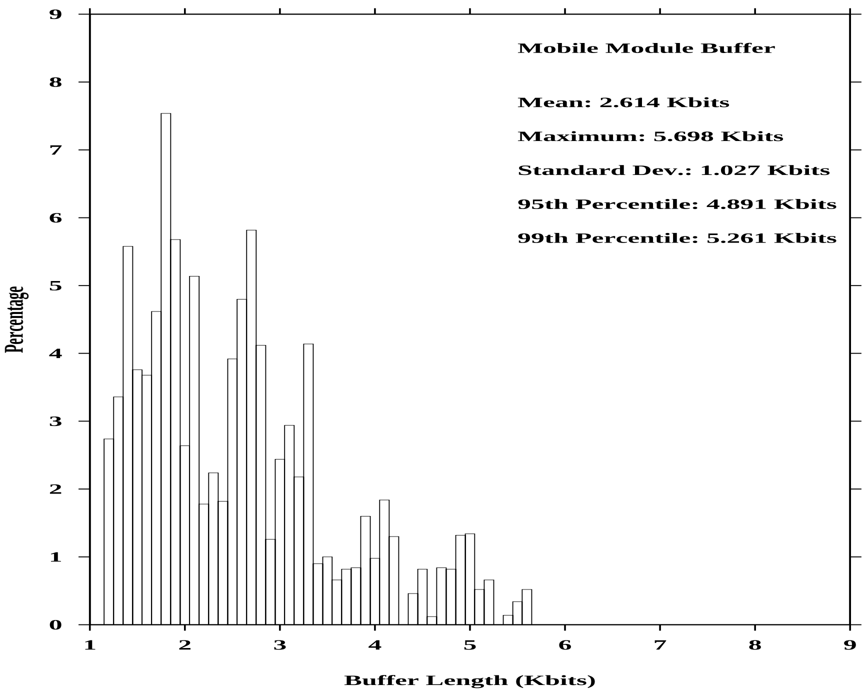

Figure 13: Upstream buffer requirement in mobile module

Handover buffering within different datapath modules are measured during 5000 separate handovers and the related statistics for topology-I are presented in Table 4. Buffer length (in Kbits) distributions for a CBR stream with a datarate of 384 Kbits/s[8] are also illustrated in Figures 13 and 14. When investigated, the primary reason for handover buffering variation turned out to be thread scheduling jitter [27] within WANDA kernel. Careful thread scheduling priority tuning was required [19] to minimize this variation.

As expected, the amount of buffering needed within the mobile module is much more than its base and representative counterparts. Although the observed maximum is almost 5.7 Kbits, a mobile terminal can handle 95% of the handovers with a buffer size of 4.9 Kbits. This gives an indirect estimate of data loss rate for a given buffer size and a specified stream data rate. Since the required buffer size depends only on the handover latency and stream datarate, relative positions of BS and MR modules affect only the mobile module buffering. Absence of any signaling overhead within a representative module (see Figure 8) explains its low downstream disruption and small buffer requirement; disruptions within an MR

[8]Data blocks of 48 octets, transmitted once in every millisecond.

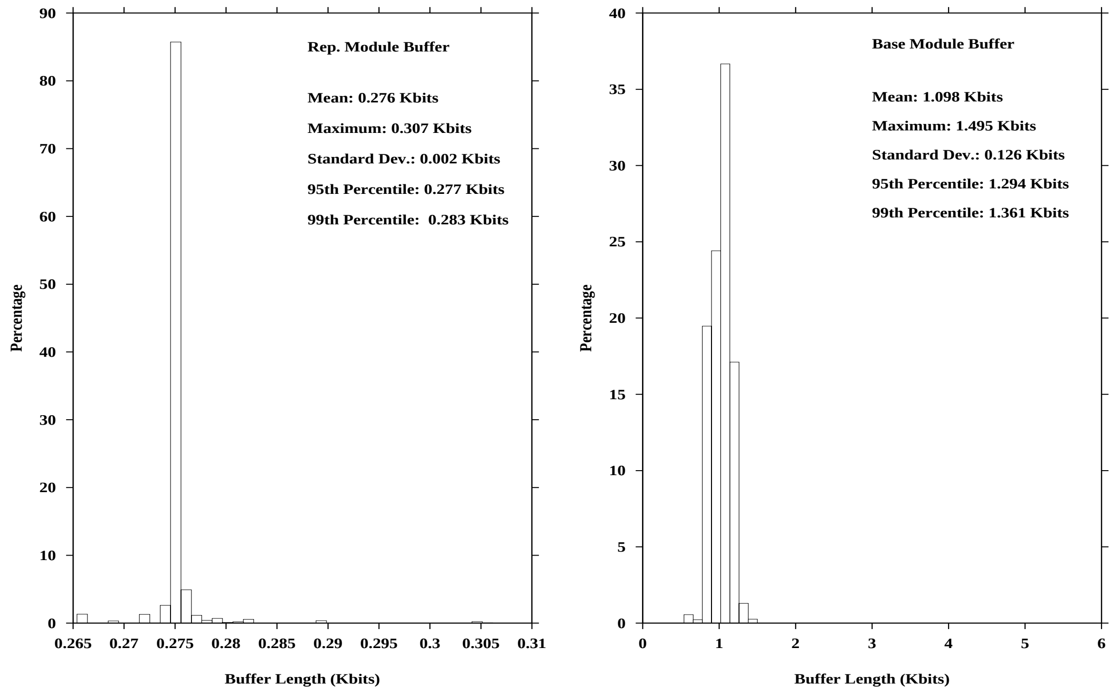

Figure 14: Downstream buffer requirement in representative and base modules

occur only during the segment switching operation. Within the MUs and the BSs, on the other hand, both disruption and buffer length figures are higher because of heavier handover signaling load.

6.3.2 Datapath Performance

The following experiment is done for measuring the performance of the datapath modules of a mobile connection. The mobile terminal sends 96 octet data blocks once in every 40 ms and the destination workstation turns it back to the terminal, immediately after reception. This procedure is repeated for 50000 blocks and the corresponding average round trip delay (in topology-I) is presented in the first column of Table 5. Time spent by the blocks in different datapath modules are also reported in Table 5. Note that the delay experienced by a data block within base and representative modules are greater than that within a mobile module. Larger number of data forwarding threads within those modules account for this difference. It is also observed that the bigger size blocks experience larger delay within all the datapath modules. This is partially due to higher MSSAR layer segmentation and reassembly overhead for larger blocks. The user space datapath modules also require to copy each data block from the kernel to the user space and back; larger blocks incur higher copying delay and that in turn increases the delay offered by the datapath modules.

Stream jitter, measured as inter-block arrival time distribution, on a topology-I connection

End-to-end Round trip (μsecs)	Mobile Module (μsecs)	Base Module (μsecs)	Rep. Module (μsecs)
2326	240.23	393.84	368.99

Table 5: Delay offered by the datapath modules

is shown in Figure 15. The mobile terminal sends MSSAR blocks once in every 40 ms and the corresponding inter-block arrival time is measured at the destination workstation. This inter-arrival delay is sampled for 50000 blocks and the corresponding distribution is plotted as a histogram with a group-interval of 40 μs. The main reason for stream jitter was found to be frequent context switching of the data pumping threads within various datapath modules. CPU contention, caused by a set of background location and topology reconfiguration threads, also contribute to this large variation of inter-arrival time. The results show that an increment of block size causes the 99th centile figure of the inter-arrival time distribution to rise. This is understood as follows: within a datapath module, once a pumping thread is scheduled for execution all the other threads have to wait till a block is completely forwarded by the currently active thread. With larger blocks, this effect becomes more prominent and as a result the stream jitter increases.

6.3.3 Handover Performance Summary

All the presented handover performance figures correspond to a *with hint* scenario where the necessary connection segments are created before an actual handover procedure is started by the mobile terminal. In a *without hint* situation, on the other hand, a mobile is forced to start a handover (see Section 5.4) when the active radio link fails suddenly. This requires the mobile terminal to wait for link failure detection before it can start the handover signaling procedure. This delay in link failure detection gives rise to the difference between *with* and *without* hint handover performance figures which are highlighted in Figure 16. Measurements of these *without hint* handover performance are planned as a future extension of this work.

The data loss expressions in Figure 16 correspond to an infinite buffer situation and therefore do not include the losses caused by buffer overflow. In a *with hint* case, the upstream mobile and downstream representative data losses are present only when the quality of radio link between the mobile unit and the old base station deteriorates soon after the handover initiation. In a *without hint* handover, no disruption is present within the *new* base station and that is why no base station handover buffering is required. This also eliminates the possibility of any packet loss within a base module.

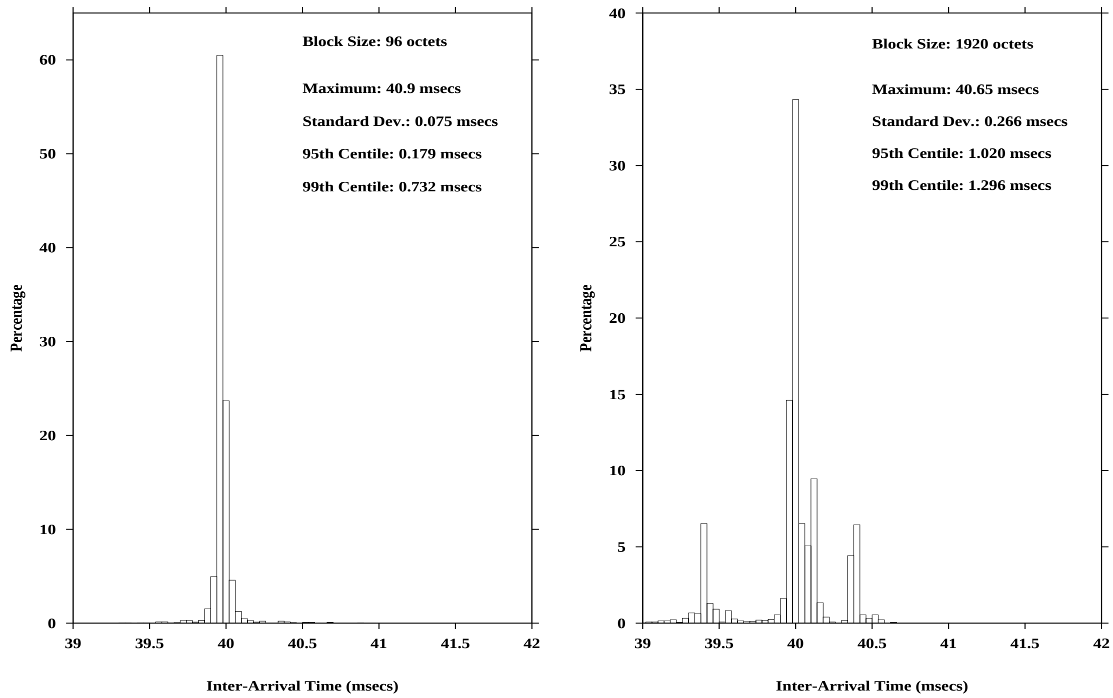

Figure 15: Effects of block size on stream jitter

7 Concluding Remarks

This paper presented a signaling architecture for supporting mobility in an ATM based radio network. A new concept of mobile representative has been introduced where these per-mobile software agents are used for concealing mobility from the fixed network entities. It has been shown that the representatives can efficiently migrate over the fixed backbone network for distributing mobility processing load. The use of the representatives as pre-assigned handover anchor points also eliminates the need for an online anchor selection procedure during the handovers. The details of management protocols for representative creation and migration were presented. A segment caching scheme was also proposed for designing scalable and low-latency ATM connection handovers. A prototype implementation and its experimental results were reported for illustrating the feasibility of the presented architecture. It was shown that the present scheme of representative mediated segment caching can support quite low latency handovers with acceptable buffer requirements in the mobile and the fixed network entities. Although this paper does not address the Quality of Service (QoS) issues, it is possible to tailor the architecture for supporting per-stream QoS guarantee. The QoS and handover cell loss/out-of-ordering issues are currently being investigated.

	With Hint Handover	Without Hint Handover
Upstream Disruption at MU	T_{mu} (Measured)	$T_{muw} = T_{mu} +$ Link failure detection delay at MU
Downstream Disruption at BS	T_{bs} (Measured)	Not Present
Downstream Disruption at MR	T_{mr} (Measured)	T_{mrw}: Starts: On Rx. of 'failure inform' Ends: On Tx. of 'reply'
Upstream Buffering at MU	T_{mu} x Datarate (Measured)	T_{muw} x Datarate
Downstream Buffering at BS	T_{bs} x Datarate (Measured)	Not Present
Downstream Buffering at MR	T_{mr} x Datarate (Measured)	T_{mrw} x Datarate
Upstream Loss at MU	Normally Not Present	Link failure detection delay at MU x Datarate
Downstream Loss at BS	Not Present	Not Present
Downstream Loss at MR	Normally Not Present	(Link failure detection delay at old BS + Transit time of message 'failure inform') x Datarate

Figure 16: Performance expressions for *with* and *without* hint handovers

8 Acknowledgements

The authors would like to acknowledge useful discussions on the topic with John Porter of Olivetti Research Ltd. Cambridge and Simon Crosby of Computer Laboratory. During this work S. K. Biswas was supported by a Cambridge Nehru Scholarship, granted by the Cambridge Commonwealth Trust.

References

[1] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Solutions and Challenges in Data Management," *Technical report*, DataMan project, Rutgers University, April 1990.

[2] K. Keeton, B. A. Mah, S. Seshan, R. H. Katz and D. Ferrari, "Providing Connection-Oriented Network Services to Mobile Hosts," *Proceedings of the USENIX symposium on Mobile and Location-Independent Computing,* Cambridge, Massachusetts, August 1993, pp. 83-102.

[3] I. M. Leslie, D. R. McAuley and D. L. Tennenhouse, "ATM Everywhere?" *IEEE Network Magazine,* vol. 7, no. 2, March/April 1993, pp. 40-46.

[4] K. S. Meier-Hellstern, G. P. Pollini and D. J. Goodman, "Network Protocols for the Cellular Packet Switch," *To be appeared in the IEEE Transactions on Communications.*

[5] A. S. Acampora and M. Naghshineh, "An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks," *IEEE Journal on Selected Areas of Communications,* vol. 12, No. 8, October 1994, pp. 1365-1375.

[6] L. V. Hauwemeiren, L. Vercauteren, A Saidi and T. V. Landegem, "Requirements for Mobility Support in ATM," *Proceedings of IEEE GLOBECOMM,* San Francisco, December 1994, pp. 1691-1694.

[7] V. C. M. Leung, N. Qian, A. D. Malyan and R. W. Donaldson, "Call Control and Traffic Transport for Connection-Oriented High Speed Wireless Personal Communications Over Metropolitan Area Networks," *IEEE Journal on Selected Areas of Communications,* vol. 12, No. 8, October 1994, pp. 1376-1388.

[8] B. Jabbari, G. Colombo, A. Nakajima and J. Kulkarni, "Network Issues for Wireless Communications," *IEEE Communication Magazine,* vol. 33, no. 1, January 1995, pp. 88-98.

[9] E. Hyden, J. Trotter, P. Krzyzanowski, M. Srivastava and P. Agrawal, "SWAN: An Indoor Wireless ATM Network," *Proceedings of ICUPC,* Tokyo, Japan, November 1995, pp. 853-857.

[10] M. Karol, M. Veeraraghavan and K. Eng, "Mobility-Management and Media-Access Issues in the BAHAMA Wireless ATM LAN," *Proceedings of ICUPC,* Tokyo, Japan, November 1995, pp. 758-762.

[11] H. Hashemi, "The Indoor Radio Propagation Channel," *Proceedings of IEEE,* vol. 81, no. 7, July 1993, pp. 943-968.

[12] S. K. Biswas, J. D. Porter and A. Hopper, "Performance of a Multiple Access Protocol for an ATM Based Pico-Cellular Radio LAN," *Proceedings of the Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* Boston, Massachusetts, October 1992, pp. 5.2.1-5.2.6.

[13] J. D. Porter and A. Hopper, "An ATM Based Protocol for Wireless LANs," *Olivetti Research Ltd. Technical Report,* no. 94-2, April 1994.

[14] R. Yuan, S. K. Biswas, L. J. French, J. Lee and D. Raychaudhuri, "A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks," *To appear in the NOMAD special issue on Wireless ATM, in early 1997.*

[15] Chai-Keong Toh, "Performance Evaluation of Crossover Switch Discovery Algorithms for Wireless ATM LANs," *To Appear in IEEE INFOCOM,* San Francisco, March 1996.

[16] S. K. Biswas and A. Hopper, "A Representative Based Architecture for Handling Mobility in Connection Oriented Radio Networks," *Proceedings of ICUPC,* Tokyo, Japan, November 1995, pp. 848-852.

[17] CCITT Blue Book Recommendations Q.1000 Series, "Public Land Mobile Network Interworking with ISDN and PSTN," *CCITT Draft Recommendation,* November 1988.

[18] EIA/TIA IS-41.3, "Cellular Radio Telecommunication Intersystem Operation: Automatic Roaming," *Working Group Draft,* Rev. B, July 1991.

[19] S. K. Biswas, "Handling Realtime Traffic in Mobile Networks," *Ph.D. Thesis, Cambridge University Computer Laboratory,* August 1994.

[20] D. E. Comer and D. L. Stevens, "Internetworking with TCP/IP, vol. I," *Prentice-Hall Inc., Englewood Cliffs,* New Jersey, 1991.

[21] C. Partridge, T. Mendez and W. Milliken, " Host Anycasting Service," *RFC 1546,* DDN Network Information Center, November 1993.

[22] B. R. Badrinath, T. Imielinski and A. Virmani, "Locating Strategies for Personal Communication Networks," *Proceedings of the Workshop on Networking of Personal Communication Applications,* Florida, December 1992, pp. II.1-II.9.

[23] D. Raychaudhuri and N. Wilson, "ATM Based Transport Architecture for Multiservices Wireless Personal Communication Networks," *IEEE Journal of Selected Areas of Communications,* vol. 12, no. 8, October 1994, pp. 1401-1414.

[24] M. Naghshineh and A. S. Acampora, "QOS Provisioning in Micro-Cellular Networks Supporting Multimedia Traffic," *Proceedings of the IEEE INFOCOM,* vol. 3, Boston, April 1993, pp. 1075-1084.

[25] M. D. Schroeder, A. D. Birrel, M. Burrows, H. Murry, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite and C. P. Thacker, "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links" *IEEE Journal on Selected Areas in Communications,* vol. 9, no. 8, October 1991, pp. 1319-1335.

[26] "Data Sheets for Am7968/Am7969-175 TAXIchip Integrated Circuits," *Technical Manual, AMD Ltd.,* 1992.

[27] M. J. Dixon, "System Support for Multi-Service Traffic," *Ph.D. Thesis, Cambridge University Computer Laboratory,* September 1991.

[28] D. R. McAuley, "Protocol Design for High Speed Networks," *Ph.D. Thesis, Cambridge University Computer Laboratory,* January 1990.

[29] C. J. Sreenan, "Synchronization Services for Digital Continuous Media," *Ph.D. Thesis, Cambridge University Computer Laboratory,* March 1993.