Mobility Management in Integrated Wireless-ATM Networks

Bala Rajagopalan
AT&T Bell Laboratories
101 Crawfords Corner Rd., Rm 1F-427
Holmdel, NJ 07733
(Email: *braja@qsun.ho.att.com*)

ABSTRACT

There is emerging interest in integrating mobile wireless communication with broadband ISDN based on the ATM technology. Many issues arise when such integration is attempted. This paper addresses the mobility management problems of tracking the current ATM addresses of mobile terminals and sustaining active ATM connections as mobiles move. The paper presents some architectural options for integrating wireless access to ATM networks and highlights an architecture based on interworking devices to provide transparent mobility support in existing ATM networks. Location management and handoff solutions for this architecture are presented. Also, how procedures being developed for multiprotocol transport over ATM networks may be adopted to perform location management is described.

1. Introduction

There is emerging interest in integrating mobile wireless communication with broadband ISDN based on the ATM technology [1, 4, 20]. It is believed that such an integration facilitates the offering of advanced telecommunication services, encompassing voice, video and data, to mobile users. Accordingly, both the European initiative on Universal Mobile Telecommunications System (UMTS) and the ITU's study of Future Public Land Mobile Telecommunication System (FPLMTS) are considering BISDN-based transport for wireless access networks [9].

The generic architecture of such a network is shown in Figure 1. Here, the BISDN serves both fixed and mobile users. Mobile terminals (MT) communicate with base stations (BS) over a wireless interface. Base stations are assumed to be controlled by mobile switching centers (MSCs) that have fixed connectivity to the ATM BISDN. Communication between MSCs is via the BISDN, which may have a hierarchical topology, with smaller capacity switches in the lower levels and larger backbone switches [20]. Intelligent network (IN) services, in support of mobility, are provided by add-on servers, perhaps co-located with MSCs. Connectivity to external networks is also provided.

Other than facilitating multiservice personal communication, this type of integration has some practical advantages. The ATM network may not only carry user traffic, but also the signaling messages required to support mobility. Also, ubiquitously deployed private and public ATM networks could provide readily available local and wide area connectivity for wireless access networks.

Many issues, however, arise in integrating wireless access networks with BISDN (for example, see [13, 14, 18, 21]). Our interest in this paper is on supporting mobile endpoints in ATM networks. Specifically, we are interested in exploring architectural options for adding mobility support to ATM networks *externally*, and in detailing solutions to the following *mobility management* problems:

—*Location management*: The tracking of MTs as they move between MSCs attached to different ATM interfaces, and

—*Handoff*: The retention of active ATM connections when MTs move from one MSC to another attached to a different ATM interface.

Our emphasis differs from some earlier proposals for mobility support in ATM networks that require additional ATM-level protocols or switch hardware. While standards could be developed to support such features eventually, solutions external to the ATM network are worth considering for initial deployment, for adding value to networks that do not have embedded mobility support, and for insulating core ATM networks from the additional processing and communication overheads imposed by user mobility.

In the next section, we describe an architecture for implementing mobility management solutions in ATM networks and outline the generic functions that need to be incorporated. In Sections 3 and 4, we present solutions to the connection handoff and location management problems in ATM networks, respectively. We consider the particularly difficult case of handoff during mobile-to-mobile communication. Related work is described in Section 5, and summary and conclusions are presented in Section 6.

2. A Mobility Management Architecture

2.1 Overview

We consider each MT to be a mobile ATM endpoint, with a permanent id, called the Mobile Identification Number (MIN). The MIN of an MT is used as the destination identifier when a connection is to be set up to the MT. But to be able to route a call to the MT, the MIN must be associated with the ATM address of the MSC under whose control the MT currently is. This procedure, called location management, is typically implemented by requiring the MT to register its presence with each MSC it attaches to, and by updating one or more network databases of the current MIN-to-MSC-address association. A separate procedure, called handoff, is needed to sustain active connections when the MT changes its ATM point of attachment.

MOBICOM 95 Berkeley CA USA
© 1995 ACM 0-89791-814-2/95/10..$3.50

In the absence of specific support for mobility in the ATM network, such functionality may be built external to the network, using *interworking devices (IDs)*. IDs attach to the ATM network at various points, and implement mobility management functions. Under this model, a call originating from a fixed host, or an MT, to a mobile is first signaled to an ID. The ID, using IN services, performs address discovery and bridges the call between the source and the mobile (Figure 2). Our desire is to make this process completely transparent to fixed hosts and ATM switches. Such transparency can be achieved through a combination of ATM addressing and routing, as described below.

2.2 Addressing

Consider the ATM *NSAP* (Network Service Access Point) address format, shown in Figure 3 [12]. An NSAP address indicates a unique ATM end system. Using a reachability protocol, ATM switches in a network determine the route to a destination, given its NSAP address. The NSAP address consists of an Initial Domain Part (IDP) and a Domain Specific Part (DSP), as shown in the figure. The IDP has sufficient information to indicate the structure of the rest of the address. The values 39, 47 and 45 have been assigned to the AFI field to indicate three distinct address formats, *DCC, ICD* and *E.164,* respectively. Now, suppose we create a *MIN-format* NSAP address class, with a unique AFI value. In Figure 3, an arbitrary value of 83 is used for this AFI. As described below, ATM switches can use the unique NSAP prefix to route call set-up messages to an ID.

Now, IDs, as a class, may be identified by a *generic* NSAP address, obtained from this space. In particular, Figure 3 illustrates an address, obtained by setting the ESI field to zero. This address, therefore, is not assigned to any MT. It is generic in the sense that a call set-up message with this address is routed to *some* ID. Thus, when an MSC has to select an ID to effect connection handoff or other control functions, it can launch a call set-up with this address. The message will be routed to the nearest ID. The ID, seeing the generic address, will know that the message is directed to itself, and not to an NSAP-addressed mobile.

It is assumed that MSCs are fixed ATM endpoints, and therefore have regular NSAP addresses, in a format other than MIN. Finally, for inter-ID communication, described in the next two sections, each ID must be assigned a unique identifier, say, a regular NSAP address.

2.3 Routing

We consider a model where IDs are connected to ATM switches over a private network-to-network interface (PNNI). Assume that the PNNI routing scheme [11] is used in the network. Briefly, this scheme allows each ID to interact with directly attached ATM switches as a routing peer. In particular, IDs may advertise the reachability of MIN-format NSAP addresses to all ATM switches in a network. Switches route calls with MIN addresses to the nearest ID. Thus, when a host generates a call set-up message to an MT, the message is routed automatically to an ID, without any special action on the part of ATM switches.

In the virtual path overlay architecture described in Section 2.5, each ID is required to know the addresses of "nearby" fixed hosts and MSCs. An ID may acquire this information either via configuration, or by receiving some approximate routing information from ATM switches across the PNNI interface [11].

The PNNI-based solution is robust, since a failure of an ID will result in a switch selecting a new ID to forward call set-ups addressed to MINs. Under this scheme, IDs begin to look like ATM switches with mobility-related enhancements.

The routing requirement at the ID, however, can be eliminated if ATM switches are forced to route MIN-addressed call set-ups towards an ID via manual configuration, and IDs are configured with the NSAP prefix of hosts and MSCs reachable nearby. This simplifies ID operation, but it may not be a desirable solution due to the need for manual labor. Furthermore, if a designated ID is unreachable due to a failure, the corresponding switches cannot complete calls to any MT.

Finally, communication between MSCs and IDs is required for mobility management. Using the generic ID address, an MSC may establish an SVC, whenever necessary, to an ID. Similarly, an ID may set up a connection to an MSC by directly using the latter's ATM NSAP address. SVCs for mobility-related control messages, however, impose some signaling load on ATM switches. This overhead can be eliminated by setting up PVCs between each MSC and a designated ID. Figure 2 illustrates the interconnection of various components. In summary,

- MTs are assigned NSAP addresses with an AFI prefix different from those already defined for the three NSAP classes. The new format is called the MIN format.
- IDs recognize a generic address in MIN format as denoting themselves. This address may be used by MSCs to set-up connections to the nearest IDs.
- MSCs are assigned regular ATM addresses.
- IDs may attach to the ATM network via PNNI interfaces. This completely automates call forwarding between MTs and fixed hosts. Or, manual configuration in ATM switches may be used to direct set-up messages appropriately.

2.4 The End-to-End Protocol Model

Figure 4 illustrates one possible end-to-end protocol model for communication between a fixed host and an MT, using IDs. Here, it is assumed that the ATM connection from the source may terminate at the MSC, and a separate, *mobile ATM (MATM)* layer may be used for the connection extended to the MT from the MSC. The MATM layer may use cell sizes and header information optimized for the radio medium and may offer a radio UNI that is different from the conventional ATM UNI. In this figure, PHY and RPHY indicate ATM physical layers, wired and radio, respectively. Also, the fixed and mobile hosts implement the appropriate ATM adaptation layer (AAL) functionality.

2.5 The Virtual Path Overlay Network Architecture

A drawback of using only switched VCs for communication with mobiles is that it exposes the ATM network to increased call set up activity when MTs move. Briefly, when an MT moves from one MSC to another, the VCs associated with an active connection must be either augmented or replaced with new VCs, depending on the handoff mechanism (Section 3). For instance, assume that the MT in Figure 2 moves from MSC A to MSC C. One handoff solution is to let ID B clear the VC to MSC A and set up a new VC to MSC C. This activity imposes additional call set up load on the ATM switches enroute. Moreover, this type of handoff could involve many switches when the ID B and MSC C are far apart. In such situations, the handoff delay also increases. Ideally, handoffs should be handled by an ID closest to the mobile's location, but when we allow only single ID in the connection path, there is not much flexibility in terms of the handoff schemes that can be implemented.

One way to minimize the call set-up load on the network is to allow multiple IDs in the connection path and connect IDs with permanent virtual paths (VPs). The *VP overlay network* interconnection architecture is shown in Figure 5. The overlay network can be used for carrying both control messages between IDs, and virtual circuits originating from, or terminating at, MTs. Such VCs are signaled transparently to the ATM network and impose no additional load on the switches. The control messages may flow on low bandwidth SVCs (or PVCs) between each pair of IDs, over the corresponding VP. The capacity of the VP network can be efficiently engineered based on pair-wise traffic load expected between the IDs (for example, as described in [6]). The flexibility of ATM networks allows new VPs to be quickly set up and capacity to be added or removed easily from existing VPs. The VP network allows efficient handoff schemes to be implemented, without switch participation, as described in the next section.

The VP overlay network may provide complete interconnection among the IDs[1]. In general, however, the interconnection may be based on an arbitrary topology, as shown in Figure 5, according to the projected traffic demands between ID pairs and the cost of VPs. In this case, each ID must either take part in a routing protocol to determine how to reach other IDs, or be administratively configured with this information. To allow the processing of handoffs by an ID close to a mobile, we let multiple IDs in the connection path. MSCs work together with nearby IDs to implement mobility management functions. IDs become aware of the ATM addresses of the MSCs they serve, either through configuration or via the PNNI routing protocol. Similarly, each MSC may be configured with a designated ID[2]. In any case, the NSAP addresses of MSCs and fixed hosts served by an ID are made available to all IDs via the routing protocol. Thus, when an ID receives a connection set-up request destined for an MT, it does the following:

1. Determine the ATM address of the MSC to which the MT is attached, and the destination ID serving the MSC (Section 4).
2. Using the routing information, determine the next ID in the path to the destination ID, if the current ID is the not the destination.
3. Reserve resources for a VC on the VP to the next ID, as per the QOS requested by the source. Signal the connection set-up to the next ID, with the QOS information, destination ID address, MIN, etc.
4. Establish a cross-connect between the incoming and outgoing VCs that preserves the QOS of all connections carried.

The message finally reaches the MSC, which sets up an MATM connection to the mobile. A connection completed message is then returned by the mobile to the source, in the reverse path. Figure 5 illustrates the MSC-ID associations. Here, MSCs are connected to IDs via PVCs that carry the control traffic.

In a large network, the number of IDs could be large. In such networks, a hierarchical organization of ID interconnection may be efficient, especially for routing purposes. This is shown in Figure 6. Here, IDs within a region are interconnected by VPs according to some topology derived from capacity requirements. A *backbone* VP topology interconnects IDs between regions. A call from a source ID to a destination is bridged by multiple intermediate IDs, as shown. Each segment of the call is a separate VC. The connection handoff and location management procedures described in Sections 3 and 4 are the same whether or not IDs are organized in a hierarchical fashion.

In summary, the following functions must be implemented at an ID to support mobile terminals:

1. *Address discovery*: Given a MIN, determine the ATM address of the destination MSC, and the destination ID serving the MSC. This may involve querying one or more servers attached to the ATM network. The ID and the server must implement a reliable protocol for querying and responding.
2. *Location update*: At the request of an MSC, update the MIN-to-ATM-addresses association maintained by one or more servers.
3. *Connection identification:* When a connection is set up to an MT, assign a unique id to the connection, to facilitate mobile-initiated handoffs. In Section 3, it is shown that to implement handoffs under certain schemes, this is the only piece of information that an MT needs to maintain.
4. *Connection bridging*: Establish local cross-connects to patch incoming and outgoing ATM network connections that are part of a single end-to-end connection to an MT. If more than one ID is in the path of a connection, each ID does connection bridging. The bridging scheme must maintain the QOS characteristics of all connections being bridged.
5. *Handoff*: Reliably communicate with peer IDs to terminate existing ATM connections and establish new ones, in order to effect an handoff.
6. *Routing*: Optionally, participate in a routing protocol to determine the path to all remote IDs, so that incoming set-up requests may be forwarded appropriately. Also, advertise MIN reachability to directly attached ATM switches.

2.6 Hybrid Architectures and Others

It is possible to combine the VP overlay network architecture with switched VC interconnection of IDs. For example, considering Figure 6, switched VC interconnection may be used within regions, while the backbone network employs VP interconnection. This may be useful, if the network load imposed by handoffs within the local region is tolerable. IDs within a region thus communicate directly with all fixed hosts and MSCs in the region. When a connection is

[1] There may be more than one VP between a pair of IDs, depending on the number of simultaneous VCs expected.

[2] This is certainly the case when PVCs are used between MSCs and IDs.

to be established to a host or MT in another region, the set-up is first signaled to a designated backbone ID. The set-up then propagates to an ID in the destination region, over the VP network. This ID sets up a VC directly to the destination. For this scheme to work, identifiers of IDs must be hierarchically structured to indicate which region an ID belongs to. Or, a *region id* may be returned with location queries when ID identifiers are returned.

This type of hybrid architecture still allows regional control of handoffs. When an MT moves from one region to another, the handoff may be efficiently handled by the IDs in the backbone network, using the partial reestablishment procedure described in Section 3.

Finally, we mention two other options to implement mobility management functions in an ATM network. First, some minimal set of functions may be implemented in the ATM switch itself, but others may be off-loaded to external processors. This may be a viable option in networks where IN functions, unrelated to mobility, are implemented using a combination of external and embedded processing. For example, switches may merely have the ability to recognize calls to MINs and shift control to an IN processor off-board. The IN processor, after completing address discovery, would interact with the switch to set up a call to the destination MSC. Connection handoffs can be handled by the external processors, with minimal involvement from the switches. For instance, switches may recognize handoff related signaling and hand control over to the designated IN processor. Second, connection handoff related functions may be associated with the switches, while keeping location management related activities off-board. This is the model considered for UMTS [4]. In this paper, however, we concentrate on the ID-based VP overlay architecture for mobility management. In the next two sections, we describe how the ID is used to implement connection handoff and location management.

3. Connection Handoff

Consider the call set-up from a fixed host to an MT, as in Figure 2. There are many ways of implementing connection handoff. These require the cooperation of the source and destination IDs, and the MT. In the following, we look at these. We consider two cases, with an MT being the destination in each case:

I. *The communication is originated by a fixed host*:

1. The MT is initially under the control of MSC_A, which is associated with ID_A.
2. A fixed host, via ID_B, originates a connection for the MT. The MT receives and accepts the connection request from the ID.
3. After the connection is set up, the MT moves to MSC_C, which is associated with ID_C.

II. *The communication is originated by* MT_s. *The destination is* MT_d.

1. MT_s and MT_d are under the control of MSCs, MSC_A and MSC_B, respectively. The associated IDs are ID_A and ID_B, respectively. MT_s originates a connection set-up request and MT_d accepts it.
2. When the connection is still active, MT_s moves to MSC_C. Independently, MT_d moves to MSC_D. The IDs associated with these MSCs are ID_C and ID_D, respectively.

The second case is most general. The first case, however, illustrates the activities involved in handoff in a simple manner. The communication originated by an MT to a fixed host is similar to Case I.

In the following, we assume that when a connection is originated by a fixed host, the first ID in the connection path, say, ID_s assigns a unique id to the connection, consisting of two parts, $<ID_s, y>$, where ID_s is the NSAP address of the ID and y is a suffix that does not repeat within the lifetime of a connection. The connection id is associated with resources allocated at each ID in the connection path, and the MSC. Similarly, when a MT originates a connection, the ID associated with the MSC assigns a connection id of the form $<ID_x, y>$ as above. Finally, the VP topology interconnecting IDs is assumed to be arbitrary, with multiple hops between the source and destination IDs (e.g., the hierarchical topology).

3.1 VC Reestablishment

A straightforward way to implement a handoff is to clear the VCs in the original connection path from the source ID to the old MSC, and establish new VCs in the path from the source to the new MSC. We look at Case I and assume that a connection has been initially established between the fixed host and MT, as given in Section 2.5.

When the MT moves to MSC_C:

1. After registration, the MT informs MSC_C of the id, $<ID_B, x>$, of the open connection.
2. MSC_C sends a message to ID_C requesting the handoff of the connection, $<ID_B, x>$.
3. ID_C sends a control message to ID_B, requesting a handoff of the connection $<ID_B, x>$. This control message may be sent on a specific VC on each VP, and routed to the destination, much like a datagram.
4. ID_B sets up a new VC to ID_C, as given in Section 2.5. To clear the VCs on the original path, it also sends a clear message to the next ID in the path. The clear message propagates to MSC_A, each ID enroute releasing the VCs on both directions. MSC_A clears the MATM VC.

5. The MT, as well as ID_B, make the appropriate associations such that data queued up for the old VC are switched to the new one.

The handoff process, as described above, is expected to be quick. Thus, it is not likely that an MT may move again before the handoff is accomplished. But if the MT moves, handoff is again attempted to the current destination ID, and it will eventually succeed. On the other hand, an MT may move during connection setup, and the set up might not succeed. In this case, ID_B must determine the new location of the MT and attempt another set-up.

The handoff procedure when the MT originates a call that is terminated by a fixed host is similar. Thus, when the MT is under MSC_A and sets up a call to the fixed host, ID_A first determines the ID closest to the fixed host, say ID_B. It then assigns an id to the connection, of the form, $<ID_B, x>$, which is conveyed to MSC_A. When the MT moves from MSC_A to MSC_C, the new MSC sends a message to ID_C, requesting the handoff of the connection,$<ID_B, x>$.ID_C then sends a message to ID_B to clear the connection to MSC_A and set up a connection to MSC_C.

3.2 Partial Reestablishment

When there are multiple IDs in the path, clearing and reestablishing all the VC segments may not be necessary. Because of the spatial locality in movement, it is very likely that the reestablished path from the source ID to the new destination ID shares most of the IDs and VPs in the old path. Handoff could be more efficient if most of the VC segments in the old path could be reused in the new path, instead of being cleared. Connection handoff works as follows for Case I:

When the MT moves to MSC_C:

1. After registration, the MT informs MSC_C of the id, $<ID_B,$ $x>$, of the open connection.
2. MSC_C sends a message to ID_C requesting the handoff of the connection, $<ID_B, x>$.
3. ID_C sends a control message to ID_B, requesting a handoff of the connection $<ID_B, x>$. This message traverses the ID path from ID_C to ID_B.
4. Each ID enroute examines this request. Say ID_D finds that it has an active connection identified by the connection id. It sets up a new connection to ID_C, as per the procedure given in Section 2.5, and clears the connection on the old path. This preserves the path from ID_B to ID_D. ID_D makes the appropriate cross-connects to bridge the incoming VC on the segment from ID_B to ID_D and the outgoing VC from ID_D to ID_C.
5. The proper VC associations are made at the MT, after the new connection is established.

Considering the network in Figure 6, Figure 7 shows the partial reestablishment procedure when the MT moves to MSC-C from MSC-A. Here, the part of the connection from ID-B to ID-D is preserved and only the segment from ID-D to ID-C is newly established.

The procedure above works correctly if MTs do not move to new MSCs before the reestablishment is complete. This is a reasonable assumption, considering the speed of signaling, compared to the velocity of mobiles. But if such movement is possible, handoff may not be correctly accomplished. In particular, multiple reestablishment attempts may be simultaneously in progress. Correct handoff, however, may be ensured as follows. Suppose each MT keeps a sequence number that counts the number of handoffs initiated for a given connection. Let the handoff control messages sent by IDs in Step 3 above carry this number. Suppose in Step 4, ID_D first sends a clear message (with the sequence number) on the old path towards ID_A, gets an acknowledgment, and then sets up a new path to ID_C. If there is another reestablishment in progress, involving another ID enroute, the reestablishment with the higher sequence number may preempt the lower.

The partial route establishment results in optimal paths (as achieved by the routing protocol used by IDs) from the source to the destination IDs after handoff. This procedure is similar to that presented in [19], for a switch-associated handoff procedure. But unlike the procedure in [19] we do not require ATM switches to implement mobility-related functions. Thus, the advantage of the VP overlay network is that it allows local reconfiguration of the connection as the mobile moves, without involving the ATM switches.

Partial reestablishment, in the case of a call originated by an MT to a fixed host, is identical. To facilitate this, It is required that ID_A assign a connection id of the form $<ID_B, x>$, as described at the end of Section 3.1.

3.3 VC Extension

Another option for implementing handoffs is VC extension. Under this scheme, the destination ID in the connection path to the first MSC reestablishes the connection to each new MSC the MT moves to. This works as follows in our reference Case I:

1. When the MT accepts the call set up request from the fixed host, it keeps track of the connection id, as well the id of ID_A that was associated with MSC_A.
2. When the MT moves to MSC_C, it passes on the connection id and identifies ID_A as the terminating ID.
3. ID_C sends a request to ID_A to redirect the connection to ID_C. This request has the connection id, from which ID_A may determine the connection particulars.

4. ID_A then sets up a VC to ID_C and bridges the connection segment from ID_B to ID_A, and the new connection segment.

If the MT moves from MSC_C to a new MSC, MSC_D, the procedure above would be identical, except that ID_A would clear the VC between itself and ID_C after step 4.

This procedure is different from the *VC chaining* procedure described in [19] where the VC is continuously extended from each new MSC to the last MSC visited by the MT. That procedure is more complex and requires procedures for preventing the formation of loops.

VC extension could potentially lead to fast handoffs. Also, due to the spatial locality in the movement of an MT, this procedure may work well initially. But the partial VC reestablishment procedure could be faster, and more optimal, as the MT moves farther away from the initial MSC. VC extension, however, has application in implementing handoffs during mobile-to-mobile communication, as described next.

The VC extension procedure when the MT originates a call to a fixed host is similar, except the VC is extended on the side of the source ID, as the mobile moves. Also, it is easy to see that VC extension works correctly even if an MT moves before the completion of the process. Finally, a procedure may be incorporated in the extension process to determine when it may be more advantageous to reestablish the connection directly from the source to the destination, without involving the intermediate ID [10].

3.4 Mobile-to-Mobile Communication

When VC extension is used as a handoff technique, MT-to-MT communication can be handled very much like the fixed-host-to-mobile case above. Let us consider our reference Case II. When MT_s establishes the connection to MT_d, both ID_A and ID_B are required to be in the connection path for the duration of the connection. When MT_s moves to MSC_C, the connection is extended from ID_A to ID_C. When MT_d moves to MSC_D, the connection is extended from ID_C to ID_D. As each MT moves, the procedure described in Section 3.3 is used to extend the connection from the corresponding ID.

The paths obtained using VC extension may not be optimal. The situation improves, if partial reestablishment is used. But handling MT-to-MT communication with partial reestablishment is more involved. In Section 3.2, handoff was accomplished by partially rebuilding the connection to the MT. For this, the destination ID had to send a reroute message towards the source. If the source were also simultaneously mobile, such a technique would not work, since the reroute message may be sent towards an old location. One solution is to let one side, say, the source MT_s, use VC extension and let MT_d use partial reestablishment. As long as ID_A is in the connection path due to VC extension, this works correctly.

Partial reestablishment on one side may provide some improvement over VC extension on both sides. However, complete reconstruction of optimal paths between the MTs is more desirable. But the procedure for optimizing the connection path after every move of an MT may be complex, with undesirable overheads. In the following, we describe a compromise. Here, VC extension is used on both sides. Initially, ID_A and ID_B serve as anchors for the VC extension on their respective sides. But after each VC extension procedure, the anchor evaluates whether the new ID to which the VC is being extended is farther than a predetermined distance. If so, it requests the new ID to directly set up a connection to the other anchor. The new ID also becomes the anchor on this side. This procedure requires each MT to maintain the address of the current anchors on both sides, in addition to the connection id. Thus,

---

When MT_s moves to MSC_C:

1. MSC_C sends a handoff request to ID_C, with the connection id, $<ID_A, x>$, and the address of the current anchor, ID_A.
2. ID_C sends a control message to ID_A to extend the connection, $<ID_A, x>$.
3. ID_A evaluates if ID_C is within a predetermined distance:
   - — If so, it sets up a VC to ID_C and bridges it with the existing VC between ID_A and ID_B.
   - — Otherwise, it sends the address of ID_C to both the source and destination MTs reliably. It then requests ID_C to become the new anchor and establish the connection directly to ID_B. ID_A clears the connection to ID_B.
4. ID_A clears the old connection to MSC_A.

In Step 3, the anchor transfers the new anchor address to each MT. This control message has to pass through the remote anchor. Thus, simultaneous execution of anchor transfers on both sides can be arbitrated, by letting one side back off and resort to regular VC extension temporarily.

Figure 8 illustrates the procedure above. In this example, MT_s and MT_d are initially under ID_2 and ID_3, respectively. Then,

1. MT_s moves to ID_1. VC is extended from ID_2 to ID_1.
2. MT_d, in the meantime, moves to ID_4. ID_3 applies the distance criterion and makes ID_4 the anchor. ID_3 conveys its address to both MTs. A new VC is set up from ID_4 to ID_2 and the old VC from ID_3 to ID_2 is cleared.
3. MT_s moves to ID_5. ID_5 is made the new anchor and its address is conveyed to both MTs. A VC is set up from ID_5 to ID_4 and the old VC from ID_2 to ID_1 is cleared.

3.5 Summary

The handoff problem is challenging, especially for the mobile-to-mobile communication case. We outlined several handoff schemes, in particular, a solution for the MT-to-MT case.

An issue we did not consider was the performance impact of handoffs, in terms of ATM cells lost. When an MT moves to a new MSC, some cells received by the old MSC may not have been delivered to the MT, leading to cell loss. One way to avoid cell loss is to buffer cells at the old MSC and transfer them to the new MSC. This solution is briefly described in [22]. Minimizing cell loss may be an explicit goal for handoff schemes, in addition to the construction of optimal paths.

4. Location Management

Location management consists of a registration and a look-up parts, both involving a *location server*. Registration is invoked when an MT moves from a BS under the control of one MSC to a BS under the control of another MSC. In this case, the location server is updated of the current association between the MIN of the MT and the ATM address of the new MSC. The registration process may additionally involve authentication, user profile updating at the MSC, etc. [10]. The look-up procedure allows an ID to supply the MIN of an MT to the location server and obtain the ATM address of the MSC. Our interest in this section is to consider how procedures being proposed for address resolution when an ATM network is used to carry a higher-layer protocol traffic can be adopted to perform location management.

4.1 Multiprotocol over ATM

An ATM network could be used to interconnect hosts that use a higher layer protocol, such as IP, for communication [5]. In this case, the higher layer protocol may consider the ATM layer simply as a link layer, similar to that of, say, an IEEE 802-type local network. One of the main multiprotocol over ATM issues is that of associating higher layer addresses with ATM addresses, when forwarding data over ATM networks. The solutions to this problem was first considered by the IP-over-ATM working group at the IETF [7]. Since then the multiprotocol-over-ATM (MPOA) working group at the ATM Forum has also begun working on the issue. For concreteness, we consider an IP-over-ATM solution below.

This solution [17] allows the dynamic resolution of the ATM address of an IP device (i.e., a host or a router) attached to an ATM network, using a protocol similar to ARP [8] used in broadcast LANs. Under this *ATM ARP* scheme, a router forwarding an IP datagram over an ATM network could send a query to an *ARP server* available at a well-known ATM address. The server supplies the ATM address corresponding to the IP address, provided the target IP host (or router) had previously registered its address with it. Clearly, these procedures resemble the registration and look-up procedures described above for supporting mobility. In particular, the following exchanges are supported:

- *Register (IP address, ATM address)*: Allows an IP client (e.g., a router) to register its current ATM address (using the inverse-ARP protocol [3]).
- *Look-up (IP address)*: A client may send an ARP_REQUEST message to the server to determine the ATM address associated with an IP address.
- *Reply (IP address, ATM address)*: The server may send an ARP_REPLY message, giving the association requested.
- *Negative Reply (IP address)*: The server may indicate a lack of known association, by sending an ARP_NAK message.

The MPOA work will perhaps lead to a server interface that provides at least the above functionality, but for multiple higher layer protocols. Treating mobile addresses simply as another higher layer protocol addresses, the same interface could be used to integrate location management with MPOA service.

MPOA, however, differs from location management in that the latter may require frequent server updates compared to the former. This may be the case even though MPOA servers may require periodic refreshing of client registrations, as specified in [17]. In general, the issue of scale is important in location management, but this has not been resolved in MPOA. The generic architecture in Figure 1 allows for some minimal degree of scaling, by not requiring server updates for MT movements among BSs connected to the same MSC. The update load on the server can be further reduced by installing multiple servers and spreading the load among them, as described next.

4.2 Scaling and NHRP

A common method to achieve scaling is hierarchical aggregation, shown in Figure 9a. Here, the ATM network is divided into logical areas (represented by ellipses), with a server per area (shaded squares). Areas are aggregated into larger, super-areas, with a server per super-area. This aggregation continues until there is exactly one large area covering the whole network. The servers may thus be considered as nodes in a logical tree. Here, the MSCs are attached to the leaves of the tree. Each MSC is assigned to a leaf server and up to l_k level k servers are assigned to a level $k+1$ server. Location information about an MT are kept by all servers in the path from the corresponding MSC to the root of the tree. At the leaf server, this information consists of the MIN-to-ATM address association. At other servers, the information indicates the branch along which a search should proceed to locate the MT. This is shown in Figure 9b, using dashed arrows. When the MT moves from MSC A to another MSC B, all the servers in the branch from MSC B to the first common ancestor, S3, of A and B are updated of the new location and the old information is deleted from those servers along the branch from S3 to A. This is shown in Figure 9c. A query to determine the location of an MT is first sent to a leaf server. If this server does not have any information on the MT, the query is propagated to the ancestor of the server. The query is thus propagated towards the root of the tree, until it is received by a server that has information. Then, the query propagates towards a leaf, ultimately being received by a server that has the MIN-to-ATM association. This server directly responds to the source of the query. This is shown in Figure 9d.

Assuming locality in movement of MTs, this arrangement spreads the query and update loads on many servers. Furthermore, by using broadcast queries and caching, the necessity for a single root server, containing information about all MTs in the network, can be eliminated. Minimization of overheads in hierarchical server arrangements has been studied (e.g., [2]). The hierarchical arrangement, however, requires an extension to the basic registration and look-up protocol. But such an extension is similar to the NHRP (Non-broadcast multi-access network next-Hop Resolution Protocol) proposal [15] currently under study by IP-over-ATM and

MPOA groups.

Under the classical IP-over-ATM model described in the last section, hosts and routers directly attached to the ATM network, may communicate directly with each at the ATM level only if they belong to the same logical IP subnet (LIS). Communication between hosts (or routers) in different LISs is necessarily via intermediate routers. NHRP recognizes that in a large ATM network, several disjoint, LISs may exist. To allow direct ATM level communication between hosts (or routers) in different subnets, NHRP defines a protocol for a source to determine the ATM address of a remote destination across multiple subnets. Under NHRP, each LIS has a Next Hop Server (NHS) that keeps track of the ATM address of routers and hosts that belong to that LIS. NHSs in adjacent LISs, where the adjacency is defined by administrative configuration, exchange information about IP destinations they serve, using a route distribution protocol.

NHRP defines the means for hosts and routers to register with the NHS in their LIS. When a host, or router, in an LIS has to send a packet to a remote IP destination reachable over the ATM network, it resolves the ATM address to forward the packet by sending a query to the NHS in the LIS. The NHS, if it serves the destination, returns the ATM address. Otherwise, the query is forwarded towards the NHS serving the destination, as learned through the route distribution protocol. Thus, the query may travel across multiple LISs, until it reaches an NHS that serves the destination. At this point, the NHS may return the ATM address of the destination (or an egress router) to the source, either along the same path the query traversed or directly via an ATM connection. The details of NHRP can be found in [15].

The hierarchical server organization seen in Figure 9 can use NHRP as the address resolution protocol. Figure 10 illustrates this. Here, an LS (logical subnet) can be taken as a level $k+1$ server, along with all its level k direct descendents. Each leaf server, together with all the MSCs served by it, form an LS. Each non-root server is thus part of two LSs and each server is a "border" server (corresponding to the next hop NHS in NHRP) to a different LS. The route distribution topology is the tree defined by the hierarchical structure. The edges of the tree could be either PVCs or SVCs. Each interior node in the tree either has no information about a given MIN, or it has a *<MIN, next hop>* association, where the *next hop* indicates a descendent in the tree. The leaf servers, either have no entries for the MIN, or have an entry, *<MIN, ATM address>*. When an MT registers with an MSC, a registration message, with the MIN and the ATM address, is propagated towards the root of the tree. Each server enroute does the following:

1. If the server is the leaf in the tree:
   - If there is an entry, *<MIN, ATM address>*, update it with the new information. Do not propagate the registration any further.
   - If there is no entry for the MIN, add an entry, *<MIN, ATM address>*. Send the registration to the ancestor in the tree.
2. If the server is an interior node in the tree:
   - If there is an entry, *<MIN, next hop>*, first send a deregister message to the *next hop.* Do not propagate the registration any further. The deregister message propagates towards the leaf server which previously served the MT. Each node enroute, and the leaf, removes the entry for MIN. Next, update the entry with the new *next hop* information, where *next hop* points to the node from which the registration was received.
   - If there is no entry for the MIN, add an entry, *<MIN, next hop>*, where *next hop* is as before. Propagate the registration message to the ancestor in the tree.

This protocol is slightly different from conventional route distribution protocols, since it sometimes propagates updates only towards the root. But NHRP does not prevent this type of distribution. By treating MINs as network layer addresses and by assigning a unique protocol id for location management, the NHRP specification can be used as given [15].

5. Related Work

External architectures for integrating wireless and ATM networks has not been looked into in detail before. Also, how location management in the ATM environment relates to other problems, such as multiprotocol transport over ATM, has not been explored. Connection handoff in this environment, however, has been studied.

In [19], the authors describe a VC extension solution that builds a chain of VCs as a mobile moves to different MSCs. The VC chain includes each MSC and a mechanism is included for loop detection. This procedure is more complex than the VC extension described in this paper. The VC rebuild approach described in Section 3.1 is similar to that presented in[19], except the latter requires ATM switches to be capable of route rebuild, whereas our method implements it external to the ATM network. Also, mobile-to-mobile communication is referred to only briefly in [19].

In [17], some solutions are presented for connection handoff, for communication between a fixed host and a mobile. The solutions are full reestablishment of the connection and partial rebuild. Here too, switch level capability is assumed. Also, the solution to mobile-to-mobile communication is not considered in [17].

Finally, in [1] a route rebuild approach, along with reserved bandwidth to minimize call processing, is presented. This approach requires special switch hardware, as well as protocol support in the ATM network.

6. Summary and Conclusions

In this paper, some architectural options for integrated wireless-ATM networks were presented. The ID-based architecture was identified as the one that provides transparent mobility support in existing ATM networks. Connection handoff and location management solutions for this architecture was presented. Furthermore, how the location management problem relates to address resolution for multiprotocol transport over ATM networks was outlined. Performance aspects of handoff protocols were not considered in this paper. This is an area for further study.

References

1. A. Acampora and M. Naghshineh, "An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks," pp. 1365-1375, *IEEE JSAC*, October, 1994.

2. B. Awerbuch and D. Peleg, "Concurrent Online Tracking of Mobile Users," *Proc. SIGCOMM "93*, 1993.

3. T. Bradely and C. Brown, "Inverse Address Resolution Protocol," *RFC 1293*, January, 1992.

4. E. Buitenwerf, G. Colombo, H. Mitts, and P. Wright, "UMTS: Fixed Network Issues and Design Options," *IEEE Personal Communications Magazine*, pp. 30-37, Febrauary, 1995.

5. H. J. Chao, D. Ghosal, D. Saha, and S. K. Tripathi, "IP on ATM LANs," *IEEE Communications Mag.*, pp. 52-59, August, 1994.

6. I. Chlamtac, A. Farago, and T. Zhang, "Optimizing the System of Virtual Paths," *IEEE/ACM Trans. Networking*, 581-587, December, 1994.

7. R. Cole and D. Shur, "IP over ATM: A Framework Document," Internet Draft, draft-ietf-atm-framework-doc-01.txt, Febraury, 1995.

8. D. Comer, *Internetworking with TCP/IP*, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1988.

9. J. S. Dasilva and B. E. Fernandes, "The European Research Program for Advanced Mobile Systems," *IEEE Personal Communications Magazine*, pp 12-17, Febrauary, 1995.

10. EIA/TIA, "IS-41.3-C Cellular Radio Telecommunications Intersystem Operations: Automatic Roaming Information Flows," December, 1994.

11. ATM Forum, *Draft PNNI Specifications, 94-0471R10*, July, 1995.

12. ATM Forum, *UNI 3.1 Specifications*, January, 1995.

13. L. Van Hauwermeiren, L. Vercauteran, A. Saidi, and T. Van Landegem, "Requirements for Mobility Support in ATM," *Proc. Globecom "94*, pp. 1691-1695, 1994.

14. Special Issue on the European Path Towards UMTS, *IEEE Personal Communications Magazine*, Febrauary, 1995.

15. D. Katz and D. Piscitello, "NBMA Next Hop Resolution Protocol (NHRP)," Internet Draft, draft-ietf-rolc-nhrp-03.txt, November, 1994.

16. K. Keeton, B. Mah, S. Seshan, R. H. Katz, and D. Ferrari, "Providing Connection-Oriented Network Services to Mobile Hosts," *Proc. USENIX Mobile and Location Independent Computing Symposium*, pp. 83-102, 1993.

17. M. Laubach, "Classical IP over ATM," RFC 1577, January, 1994.

18. M. J. McTiffin, A. P. Hulbert, T. J. Ketseoglou, W. Heimsch, and G. Crisp, "Mobile Access to an ATM Network Using a CDMA Air Interface," *IEEE JSAC*, 900-908, June, 1994.

19. P. Mishra and M. Srivastava, " Call Establishment and Rerouting in Mobile Computing Networks," AT&T Technical Memo, 11384-940906-13TM, June, 1994. Available from *mbs@research.att.com.*

20. D. Raychoudhuri and N. D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless PCNs," *IEEE JSAC*, pp. 1401-1414, October, 1994.

21. P. F. M. Smulders, "Error Control in ATM-Based Indoor Radio LANs," *Proc. IEEE PIMRC "94*, pp. 237-241, 1994.

22. R. Yuan, S. Biswa, and D. Raychoudhury, "Mobility Support in a Wireless ATM Network," *Proc. WINLAB Workshop on 3rd Generation Wireless Systems*, pp. 335-345, 1995.

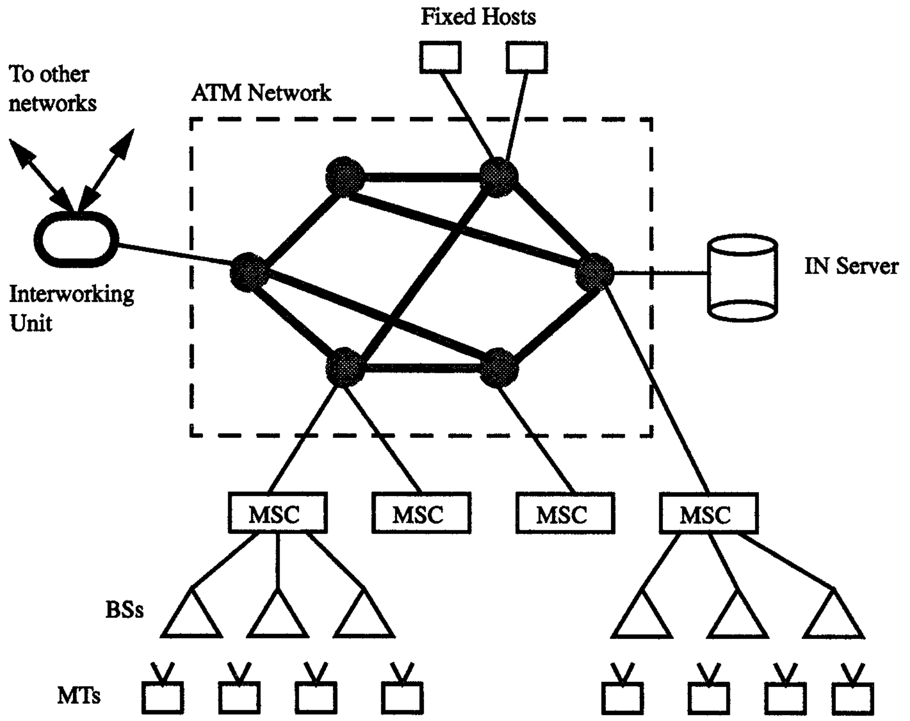
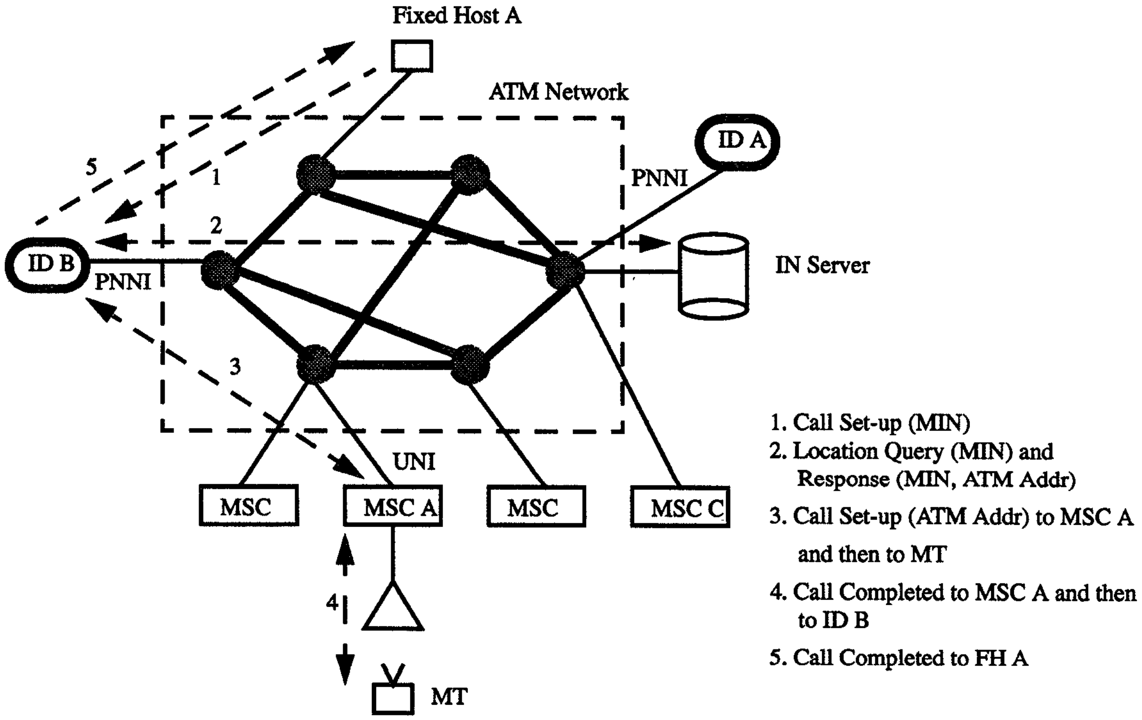
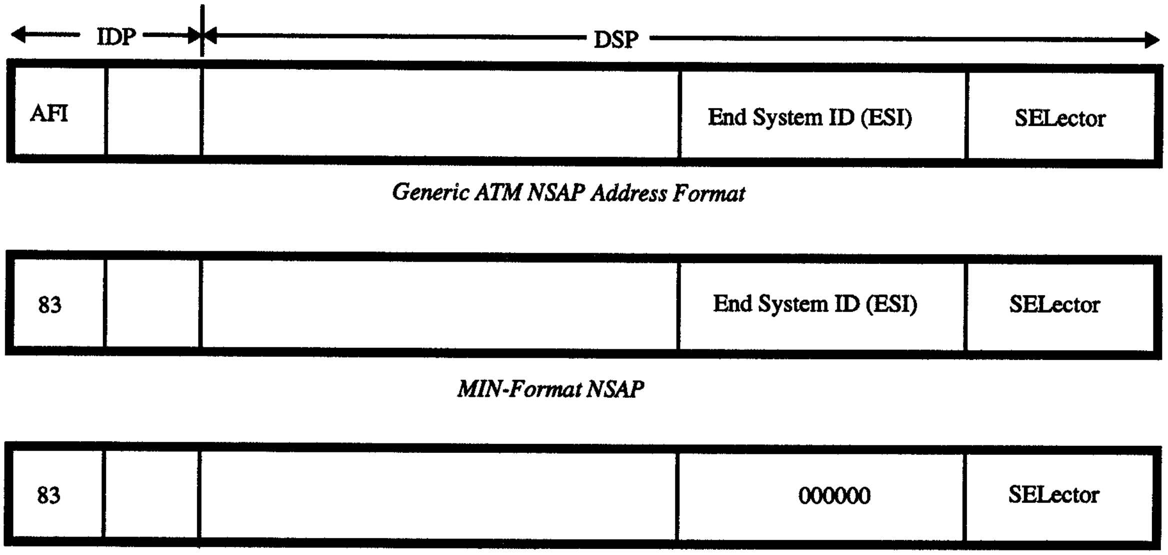


Figure 1: Generic Integrated Network Architecture



1. Call Set-up (MIN)
2. Location Query (MIN) and Response (MIN, ATM Addr)
3. Call Set-up (ATM Addr) to MSC A and then to MT
4. Call Completed to MSC A and then to ID B
5. Call Completed to FH A

Figure 2: IDs for Mobility-Related Functions



Generic ATM NSAP Address Format

MIN-Format NSAP

Generic ID Address

Figure 3: NSAP Addresses for Transparent Routing

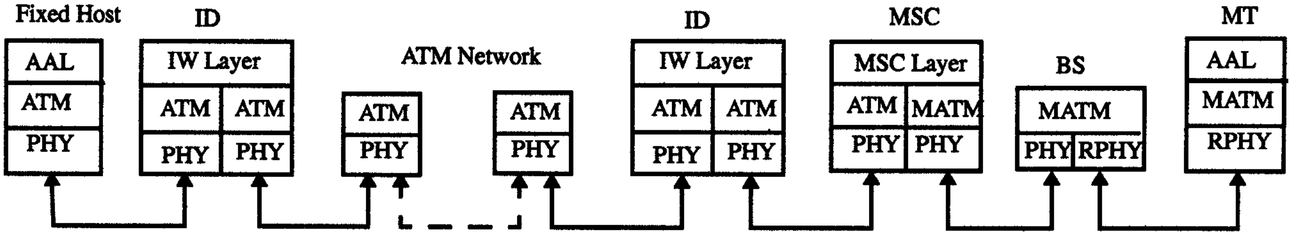


Figure 4: End-to-End Model

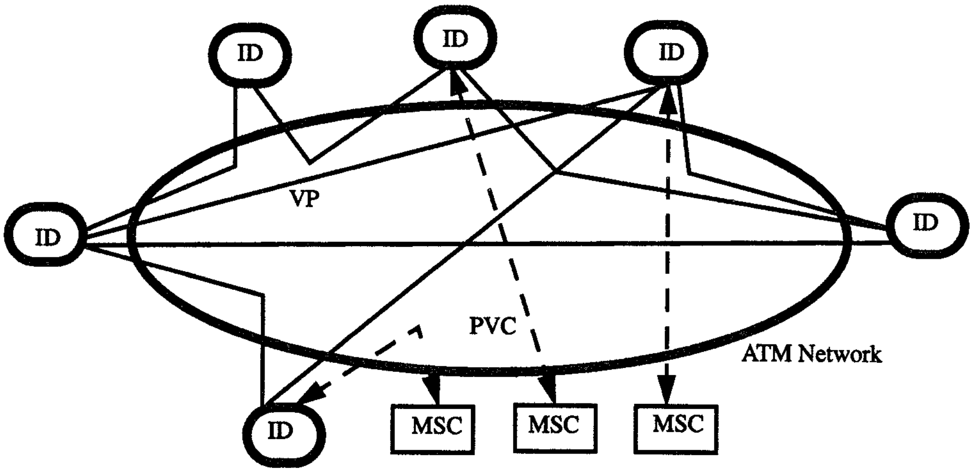


Figure 5: VP Overlay Network

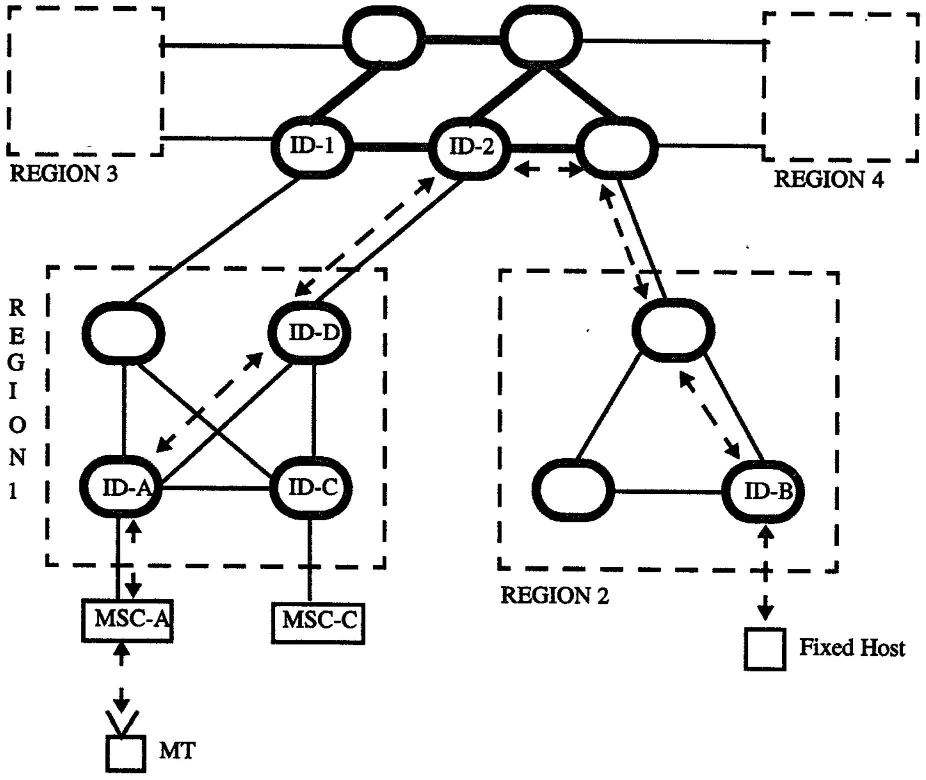


Figure 6: Hierarchical Overlay Network

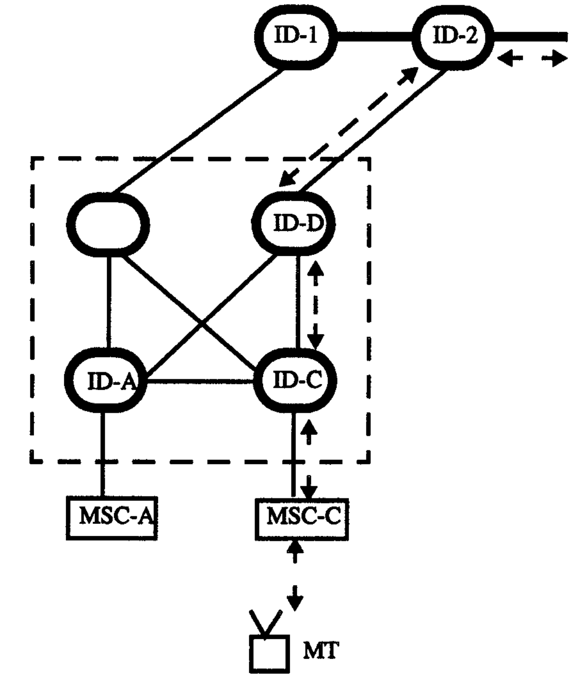


Figure 7: Partial Reestablishment

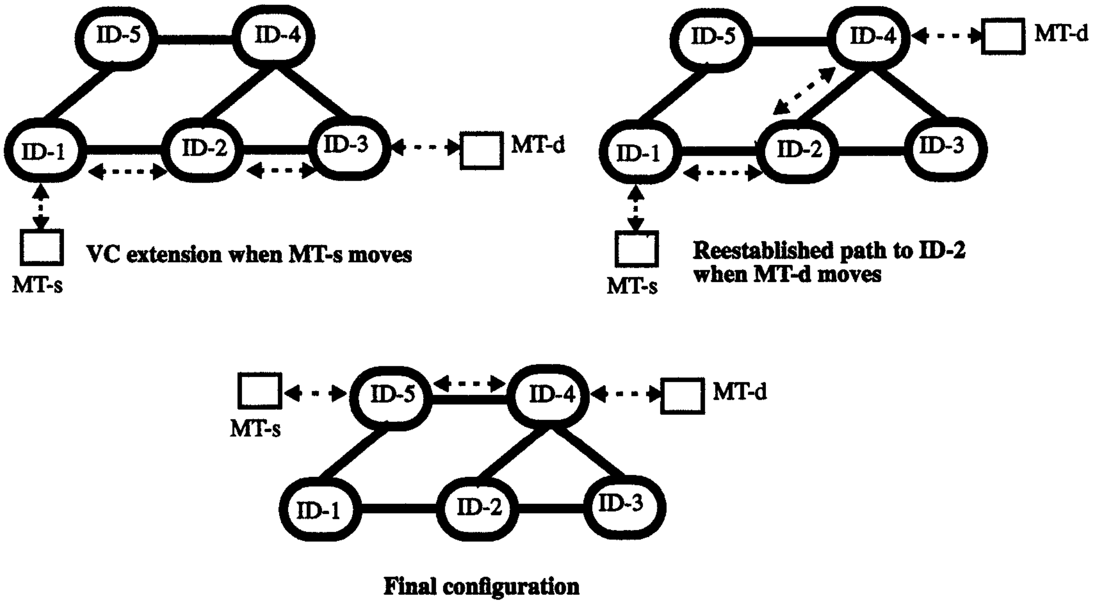


Figure 8: MT-to-MT Communication

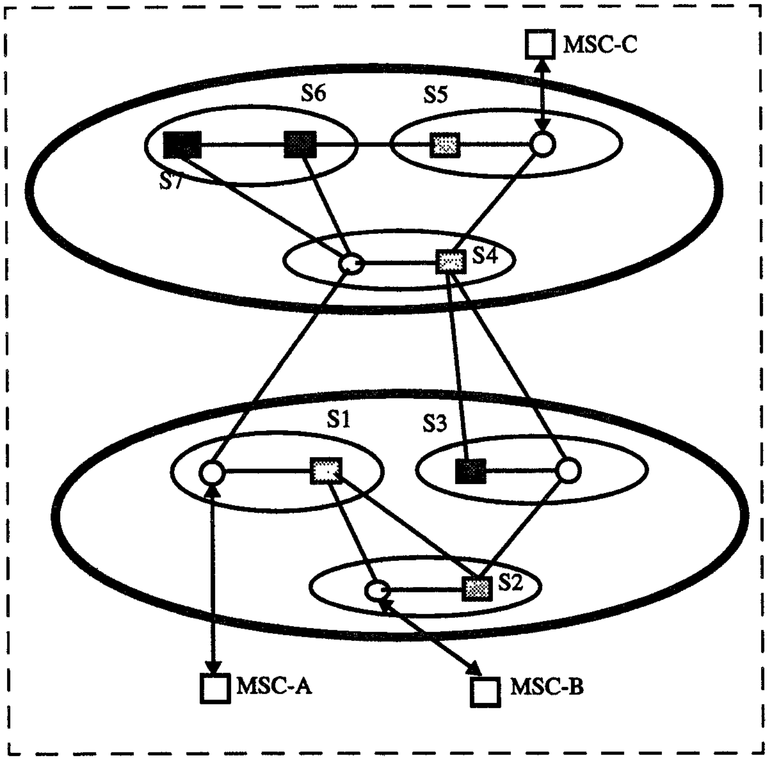


Figure 9a: Hierarchically Organized ATM Network

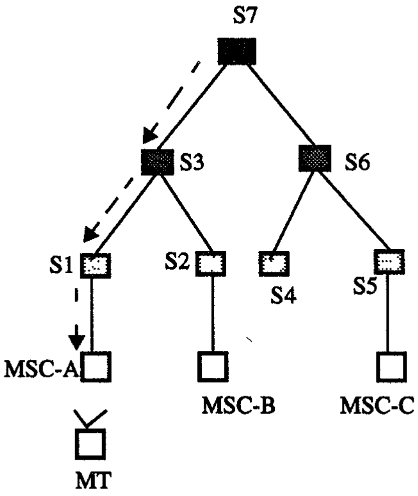


Figure 9b: Logical Server Connectivity

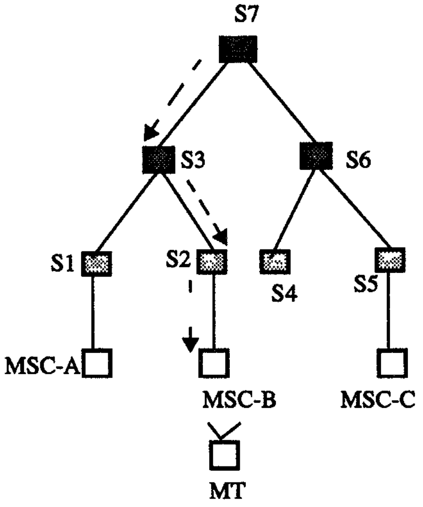


Figure 9c: When the MT moves from MSC-A to MSC-B

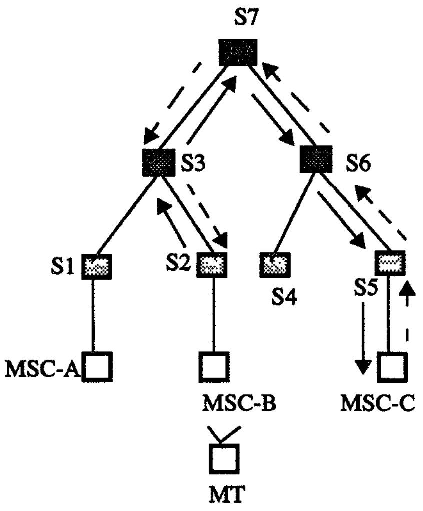


Figure 9d: Query from MSC-C and reply

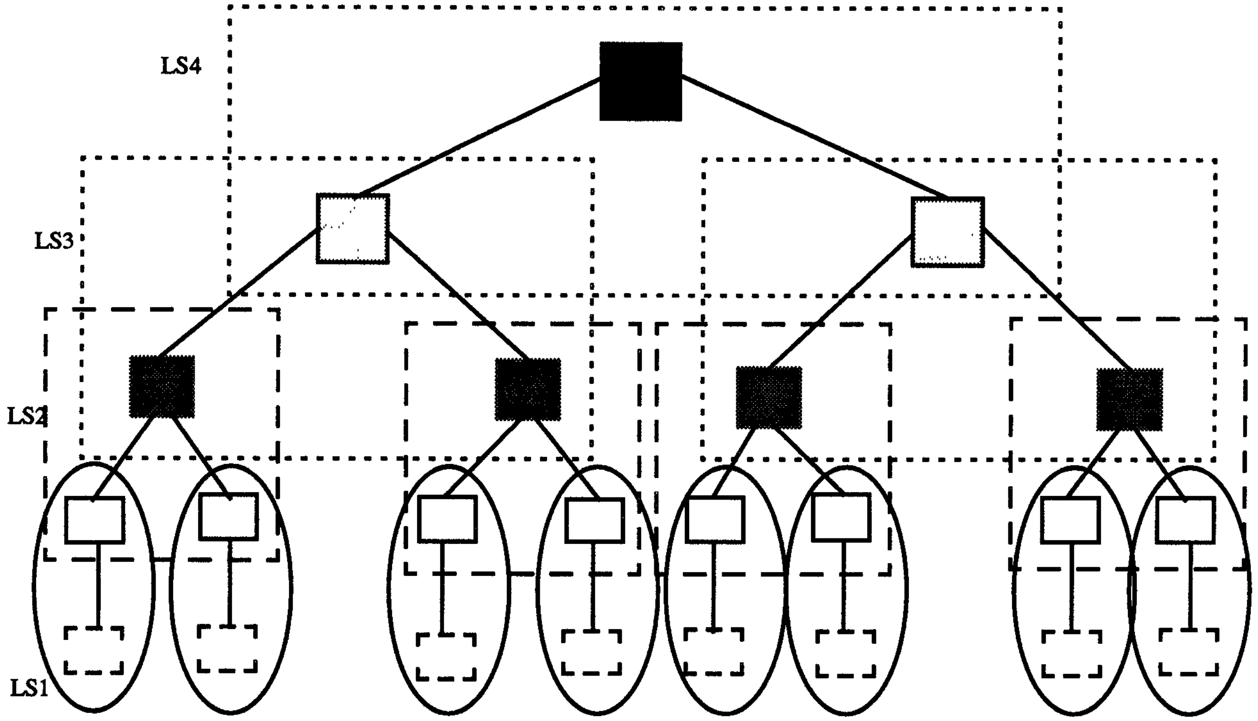


Figure 10: Server Hierarchy and Logical Subnets