Computer Communications 25 (2002) 56–73

computer communications

www.elsevier.com/locate/comcom

MCoRe: an adaptive scheme for rerouting multicast connections in mobile ATM networks ☆

Kwan-Wu Chin[a], Mohan Kumar[b,*]

[a]Communications and Networks Lab, Motorola Australia Research Center, Level 1, 12 Lord Street, Botany, NSW, 2019 Australia
[b]Department of Computer Science and Engineering, The University of Texas at Arlington, P.O. Box 19015, Arlington, TX 96019-0015, USA

Received 17 February 2000; revised 23 February 2001; accepted 4 April 2001

Abstract

In this paper we propose multicast connection rerouting (MCoRe), a novel scheme for efficient multicasting in mobile asynchronous transfer mode (ATM) networks. We investigate and evaluate McoRe's performance in mobile ATM networks in the presence of mobile source fand/or receivers. In the past, multicasting from and to mobile hosts in mobile ATM networks has not received much attention from researchers. MCoRe is based on active network (AN) technologies, where the computation required for rerouting connections is performed at switches. Due to intra-network processing, MCoRe is efficient, requires minimal buffer requirements, has low handoff latency and high reuse, and incurs low signaling overheads. Apart from that, MCoRe preserves cell ordering, prevents cell duplication and minimizes cell loss during source or receiver migration. We show how shortest path tree adapts to source migration without incurring high signaling overheads. With MCoRe, source migration is transparent to the multicast group members and does not entail reconstruction of existing multicast tree. We also show how receiver migration is achieved and how cell ordering is preserved, when the receiver rejoins the multicast session.

Keywords: Multicast; Connection rerouting; Mobile networks; ATM

1. Introduction

With the proliferation of mobile computers, one of the main research issues has been the adaptation of current protocols to support movement of end-hosts. For example, the internet protocol (IP) is augmented to cope with a mobile host (MH) through the use of mobile IP [1]. When the source is mobile, the multicast tree would need to be reconstructed, which results in cell loss and disruptions to traffic flow. On the other hand, when a receiver migrates and rejoins a multicast session it may receive duplicate cells. In addition, if the foreign network does not support multicast or no members exist at the foreign network, then the mobile receiver will experience disruptions. In ATM networks, the problem of unicast connection rerouting in the event of host migration is well studied [2]. Rerouting multicast connections in the presence of mobile sources and receivers is an area that has not received adequate attention by researchers in the past. Existing multicast protocols [3–5] for ATM networks assume fixed end hosts. The solution presented in [6] suffers from sub-optimal path and assumes a local area network (LAN).

To overcome the aforementioned problems, we propose a solution called multicast connection rerouting (MCoRe). MCoRe provides adaptivity to a source rooted tree (SRT) and is designed for rerouting multicast connections in Wide Area Networks (WANs). MCoRe is based on active networks (ANs) [7] where switches/routers are programmable, and programs are injected into them to extend their basic functionalities. The programs injected into routers/switches use the computational power to facilitate control and management of traffic passing through. In other words, computation is pushed into the network and there is no need to probe the network for relevant data for computation at end-hosts. As a result, signaling messages are significantly reduced. Hence an efficient and simpler solution, which takes advantage of the information within the network, can be realized. In this paper, the objective is to provide a solution to rerouting multicast connections in the event of mobility. That is, given a source-rooted tree, we show how

☆ This work was done while the authors were at Curtin University of Technology, Western Australia.

* Corresponding author.
E-mail addresses: kwchin@arc.corp.mot.com (K.-W. Chin), kumar@cse.uta.edu (M. Kumar).

PII: S0140-3664(02)00336-X

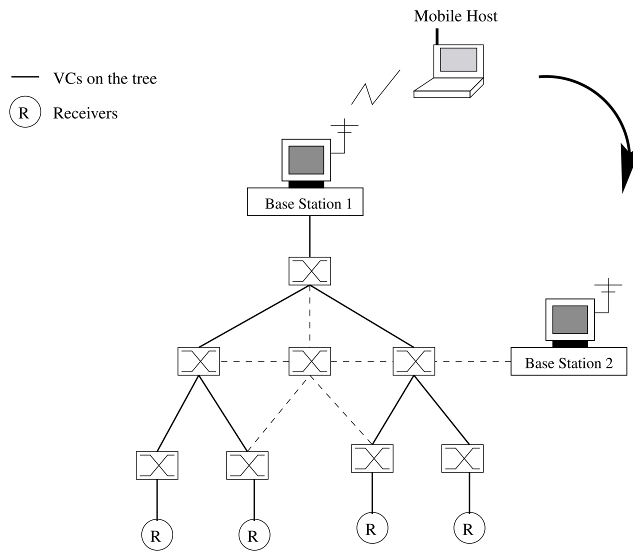


Fig. 1. An example showing the effects of handoff in connection oriented networks.

the tree can be updated to reflect the source's migration. Apart from that we show how cell ordering is preserved when a receiver migrates and rejoins its multicast session at a new location.

MCoRe uses information within the switches for rerouting multicast connections. We show how MCoRe utilizes intra-network processing to reroute multicast connections dynamically after source/receiver(s) migration. The MCoRe protocol has low signaling overheads, maintains cell ordering, incurs minimal cell loss, low handoff latency, low buffering requirements and requires minimal state (amount of information recorded) at switches. MCoRe is efficient and simple and does not employ complex algorithms in order to update the multicast tree after each migration. The most important aspect of MCoRe is that the existing multicast tree does not have to be reconstructed after source migration. Moreover, a majority of the states allocated before migration are reused.

This paper has two main contributions. First, we present a simple solution to the problem of multicast connections rerouting. So far there has been only one reported work [6] in this area. Secondly, with respect to research in ANs, the application of AN technologies in solving routing problems in mobile communications, specifically rerouting of connections have not been investigated. In this work, we demonstrate the application of ANs in solving the aforementioned problems.

1.1. The problem

The multicast approach taken in this paper is based on the SRT or shortest path tree because SRT is ideal for delay sensitive applications. It is crucial that the migration problem for SRT is addressed. The SRT in this paper is independent of the underlying unicast routing algorithm and is receiver initiated. In SRT approaches, traffic concentration at the core or rendezvous point (RP) and sub-optimal path are avoided. Traditionally, the main drawback of SRT is scalability [8]. This is due to the fact that a separate tree has to be constructed for each host interested in transmitting to the multicast session. Thus the amount of state at routers/switches is $O(S \times R)$, where S and R are the number of sources and receivers respectively. As a sub-goal of this paper we will discuss how SRT can be made scalable and accommodate multiple sources. Unlike the unidirectional tree of UNI 3.0 [3], the connections in our tree are bi-directional. Hence receivers are able to send feedback messages back to the source.

The problems addressed in this paper are illustrated in Fig. 1, where a MH, the source of the multicast migrates from BS_1 to BS_2. All receivers interested in joining the multicast session send a request to the session source. The multicast tree shown in Fig. 1 is receiver initiated and classified as a SRT. When the MH migrates to BS_2, some connections of the multicast tree are no longer valid and some multicast cells from BS_1 maybe lost. As we can see, rebuilding the tree would prove to be highly disruptive due to the time taken to setup up all the necessary connections. Furthermore, the rebuilding and tearing of connections would be costly in terms of signaling overheads.

In receiver handoff, when receiver migrates to a different tree branch, no cell duplications may occur. At the new location, the receiver may receive cells already received before migrating. The problem is aggravated because ATMs do not incorporate the notion of sequence number in the cell structure. Therefore, a synchronization process is required to determine whether a given receiver has received a particular cell. Moreover, the synchronization process must be generic in that it must be independent of higher layer protocols. Another problem with receiver migration is that the network in which the receiver migrates to may not have multicast capabilities or does not have group member(s). If no multicast capabilities exist then the receiver's session will be interrupted until it moves out of the current network. In networks, which support multicast but currently have no group members, the receiver experiences a high disruption time due to the rejoining process. On the other hand, if predictive handoff is used, a rejoin can be initiated in advance thereby minimizing disruption time.

This paper is organized as follows. In the next section we will review existing multicast protocols and address their short-comings in handling mobile hosts. Then in Section 3 a generic design specification for routing multicast connections is presented. After that, the features of MCoRe are discussed in Section 4. Section 5 describes the simulation methodology used to evaluate the effectiveness of MCoRe. This is then followed by results from our simulations, which are presented in Section 6. Then we look at issues that require further investigation. Finally conclusions are drawn in Section 8.

2. Background

2.1. Multicasting in ATM networks

In UNI 3.0 [3] the multicast model employed for ATMs is source controlled and unidirectional with no support for multicast group addressing. In source controlled models the source has to be notified of join requests. The source explicitly informs the network to add/delete receivers from the multicast tree. When a receiver decides to leave, the source is notified. Unidirectional approach is employed due to the problem of cell interleaving [9]. Cell interleaving occurs when cells from multiple sources arrive at a merge point of the multicast tree. As a result, these cells are intermixed and receivers are unable to distinguish the sources of cells. Readers interested in solutions to the cell interleaving problem are referred to Ref. [10]. Clearly the model proposed in UNI 3.0 [3] does not consider the possibility of the source/receiver being mobile. When the source is mobile, a new multicast tree will have to be rebuilt from the source's new location. As a result, a high signaling overhead is incurred and worsens when handoff rate is high. Furthermore, subsequent join requests will be directed to the source's previous location.

Currently native mode multicasting protocol in ATM networks assumes a point to multipoint model (PTM). To emulate multipoint to multipoint (MTM) connection (i.e. many-to-many connections), two methods were proposed: VC mesh and VC tree. In VC mesh, a PTM tree is built from each member to every other member of the multicast session. As a result the connections incident on a particular member increases with group size. Furthermore any changes such as add/delete in group membership will have to be notified. Therefore the VC mesh scheme becomes infeasible in mobile environments, especially when large number of signaling messages are transmitted in high mobility cases.

The second MTM model is the VC tree. This is realized with the use of multicast server (MCS). The MCS's job is to terminate VCs from sources, reassemble AAL_SDU and create another PTM to session members [5]. MCS suffers from single point of failure and congestion at the root.[1] Moreover, quality of service (QoS) is determined by the MCS instead of receivers/sources. Apart from that, end-to-end latency increases due to sub-optimal path and AAL_SDU processing [12]. The advantages of the VC tree model are: reduced receiver resource consumption, host connectivity independent of group size and one point of control for bandwidth [12]. Further, unlike the VC mesh model less signaling is incurred since joining/leaving operations do not affect sources/receivers due to the abstraction provided by the MCS. The MCS is dependent on the multicast address resolution server (MARS) [5] for multicast group membership management. MARS was introduced to provide multicast group address abstraction, which enables IP multicast to work over ATM networks. The main objective of MARS is to maintain a mapping between IP/ATM addresses and multicast group address. Receivers inform the MARS server when they want to join a particular multicast group. The receivers addresses are relayed to the sender (MCS address if VC tree model is used) and receivers added onto the multicast tree. To date, no support for mobility has been added to the VC tree model. One disadvantage is that the MCS must be present after migration since without a MCS in the current network the receiver maybe forced to connect to its previous MCS which maybe far away.

Another method currently under investigation is the use of a single shared VC for both receiving and sending [13]. The main problem with this approach is cell interleaving from multiple sources at merge point. This means that a shared tree approach (e.g. CBT [14]) can be employed in ATM networks where multicast group members only need to setup a connection to the core for sending/receiving data. The downside of this approach is that the core is chosen when the multicast session first starts up. After each migration by a source or receiver the path to the core becomes sub-optimal, hence increasing end-to-end lantecies. Recomputing the core's position after handoff is computationally expensive.

In UNI 4.0 [4] the leave initiated join (LIJ) was introduced. LIJ is issued by receivers interested in joining a multicast session. The network then responses with a leaf setup request. The root/sender does not need to know of receiver's existence. If no node on the tree is encountered, the leaf setup request is processed at the source and a network is notified to add the corresponding receiver. In Ref. [15], Tedijanto et al. investigate suitability of private network-network interface (PNNI) [16] in supporting LIJ. Due to the inability of a node to determine which nodes downstream are already on the tree, a snapshot of the tree is required to determine whether a loop will form if a LIJ is initiated. The problem is heightened if LIJ is handled by a node downstream and all nodes upstream from that node do not know the existence of the newly added receiver, therefore the new branch may cause a loop if subsequent connection is generated from a node higher upstream. To overcome the problem of looping, Tedijanto et al. [15] defined the notion of LIJ domain. A LIJ domain contains a proxy root that is responsible for setting up branches. Each proxy has a view of sub-tree (nodes subscribed to the multicast session) emanating from it. Hence it can safely determine whether any loops will form given a join request. Nodes that have more than one receiver inform upstream nodes about receivers that are downstream. The signaling required to obtain a snapshot of the multicast tree and relaying membership information to upstream nodes entails a high overhead in mobile environments. Due to the dynamic nature of members, signaling messages would need to be sent constantly in order to keep an updated

[1] A proposal to address these limitations are presented in [11].

snapshot of the multicast tree and also group memberships downstream.

2.2. Multicasting in mobile ATM networks

Toh [6] extended his unicast connection rerouting methods [17] to multicasting in ATM. In Toh's model, the MH signals to the previous foreign agent (FA) prior to handoff and the connection rerouting process starts even as the MH establishes a new connection. The connection rerouting algorithm employed is the prior path knowledge (PPK) algorithm [18]. PPK works as follows, during connection rerouting a query is made to the connection server. In response to the query, the server returns the nodes that are on the multicast tree. The MH then uses the map of the tree to calculate a crossover switch. To locate the CX, firstly a distribution point is determined. The distribution point is found by traversing from the old BS up to a point where the degree of the node is more than two. The CX is any node between the old BS and the distribution point which satisfies a given performance metric. Toh's scheme [6] considers migration within LAN only and as a result suffers from packet loss in the case of hard handoff and sub-optimal path. Furthermore Toh's scheme requires a snapshot of the multicast tree, which is infeasible with frequent change of group membership.

3. Design criteria and assumptions

In the design phase of the MCoRe protocol we attempt to satisfy a set of requirements. We believe these requirements are generic as they can be applied to subsequent work in this area. The requirements are:

- Low handoff latency
- High Reuse
- Low signaling overheads
- QoS
- Low cell loss and ordering
- Minimal state at switches
- Transparency
- Loop avoidance

Handoff latency should be minimal to ensure smooth handoff. High reuse of existing connections conserves resources because new connections entail overheads in terms of additional states and signaling. With regards to signaling, we believe a scalable solution should not require the probing of switches for information, for instance to obtain a map of the multicast tree. Probing the network incurs high signaling overheads. By reducing signaling overheads, handoff latency due to delay in probing and computation of data collected can be minimized. Besides that, the QoS established in the original tree must be maintained across handoff. Therefore during connection rerouting or when the multicast tree is rebuilt the QoS agreed upon must be satisfied.

After handoff, cells that are in-transit must be salvaged. This depends on whether the application can tolerate losses. But depending on the speed of transmission and size of the multicast tree there maybe a significant number of cells in transit. Tearing down connections without salvaging cells in-transit may significantly affect QoS observed at receivers, hence resulting in high disruption time. The salvaging operation must preserve cell ordering. After handoff, once a new tree or connection(s) has been established, the solution must ensure that salvaged cells are forwarded first before new cells are forwarded to the receivers. In the case of receiver migration, when a receiver rejoins the multicast session it may receive duplicate cells or the next expected cell may have already passed. Therefore a multicast protocol should have a mechanism to overcome the aforementioned problems when receivers rejoin the multicast session after handoff.

Switches that are part of a multicast session should maintain minimal state information regarding the session. Moreover the state should scale as the multicast group size increases. Furthermore rerouting algorithm should make use of available information within the switches, for example information acquired from the underlying routing protocol. Moreover, the rerouting process must be transparent to the source and receivers of the multicast group. This means that receivers and sources are not required to rejoin or rebuild the multicast tree after migration. Transparency is attained by having the network adapt to host movement without requiring the intervention of end-hosts.

The last considerations for any rerouting scheme must not introduce loops. Loops cause cell misordering and cell duplication. Loops arise because a given switch does not know which downstream/upstream switches are already subscribed to the multicast session. Therefore merging of connections without prior knowledge of switch subscriptions results in the occurrence of loops.

In our scheme switches within the network are programmable and there exist standard sets of interfaces for manipulating VCs related to the current multicast session. In our approach, the programs injected into ASs reroute VC(s) or VP(s) pertaining to a given multicast session. The interfaces at ASs include functions for tearing and setting up connections, updating of translation table and control over which VCs are multicast. Note that in this paper issues such as security (in ANs) and QoS adaptation will not be addressed. These issues require further investigation and are beyond the scope of this paper. Besides the above, the work presented here does not attempt to map IP multicast models over ATM networks.

4. The McoRe protocol

In this section we outline the MCoRe protocol. The main

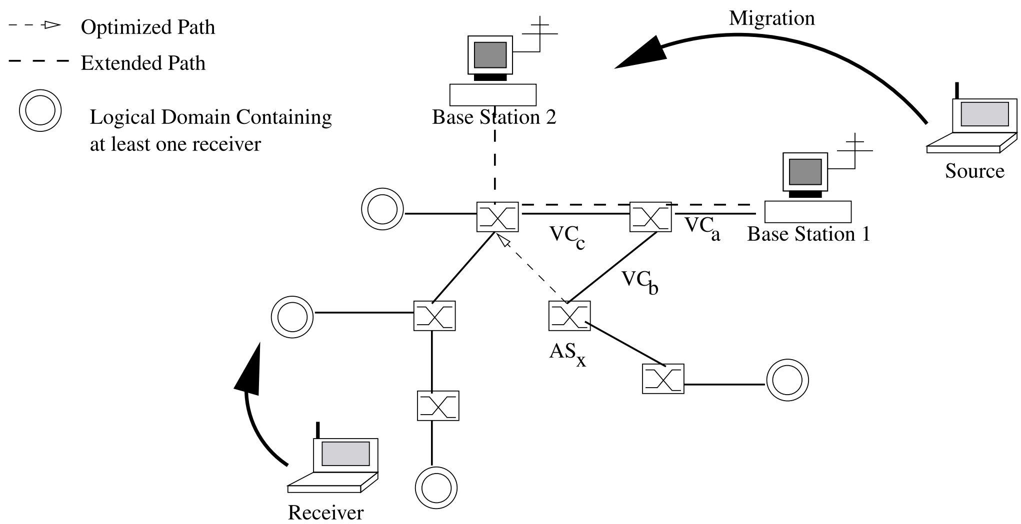


Fig. 2. An Overview of the MCoRe Protocol.

objectives of MCoRe are to satisfy the design requirements introduced in Section 3 and show how programmability within switches can be used to reduce signaling overheads involved during rerouting. We show how after source migration, the multicast tree can be updated without disruption to traffic flow, thus allowing for a seamless multicast session.

An overview of the MCoRe protocol, is illustrated with the help of Fig. 2. In Fig. 2 we see that a multicast session is rooted at BS_1. The receivers are represented as logical domains (LDs). Basically a LD represents a subnet and it contains at least one receiver. The role of a LD will be elaborated further in the next section. After handoff, the MH sets up a connection back to BS_1. In other words, a path extension process is performed from BS_1 to BS_2. Therefore after handoff the MH is able to multicast and receives packets through the extended path. After handoff MCoRe invokes an optimization process that updates the multicast tree. The MH notifies all LDs of its new address. In response, the proxy root (PR) of each LD sends out a tree update message back to the mobile source. As the tree update message traverses, the network to the mobile source's current location, each AS on the path checks whether the next hop being forwarded is the parent node. In Fig. 2, AS_x determines that the update message will not be forwarded to its parent node. Hence a new connection is setup as shown in Fig. 2. Once the update message arrives at the source, the extended connection and connections that are invalid are torn down. In Fig. 2, these connections are labeled VC_a to VC_c. Before tearing down any connections

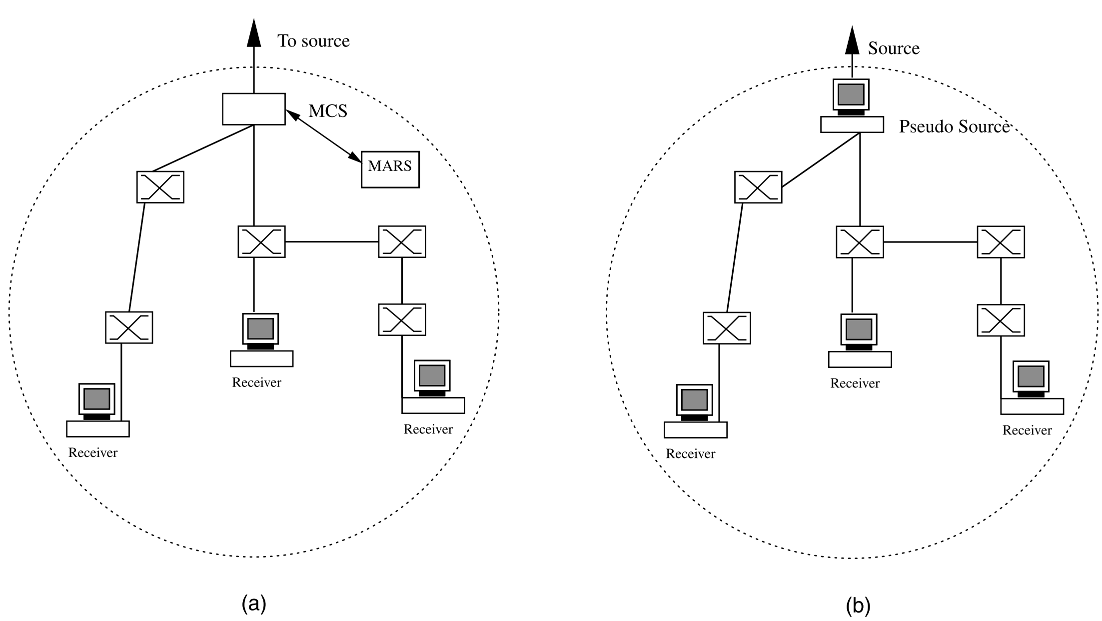


Fig. 3. Logical domain using (a) MCS approach and (b) Source-rooted

Table 1
Data Structures Maintained by AS for a given multicast session. The variables VCI/VPI/port/flag/counter are one byte in size. Multicast address and QoS contract are 32 bytes in size.

Data	Description
ParentVC	The data structure containing information about parent node. *(VCI/VPI, multicast address, port and QoS contract).*
RD_i	This list contains information for each subscriber downstream. *(VCI/VPI, QoS contract and prune flag),*
GrpID	The multicast group identifier. *(Multicast address)*
SAddr	The source's current care-of-address.
Nconn	The VCs corresponding to new connections created or received during the optimization process. *(VCI/VPI, address, port and QoS contract)*
ExConn	The VC information of the extended path during handoff. *(VCI/VPI, address, port and QoS contract)*
LastRM	The sequence number from the last resource management (RM) cell seen.
CellCount	The *k*th cell after the RM cell has passed. The counter is reinitialized to zero each time a RM cell is processed.

we need to ensure that cells are drained properly to preserve cell ordering. This is done by injecting a marker cell along the old tree which signals to core ASs to update their multicast state and start forwarding cells from newly created connections. Core ASs has more than one receiver subscribed to them.

Receiver migration also involves path extension. Other than the advantages such as fast handoff, path extension enables a receiver to continuously receive cells in networks that do not have multicast service. The next stage of receiver migration involves a rejoin of the multicast tree at the current network. Therefore the extended path is maintained for a short period of time and is torn down after the MH has rejoined the multicast tree at the foreign network. The main problem that needs to be addressed is cell synchronization to ensure that the receiver obtains the next expected cell after it has rejoined the multicast session.

4.1. Logical domain

As mentioned in the previous section MCoRe assumes the notion of LD. LD is a logical grouping of receivers and can have its own multicast protocol. For example all receivers belonging to a peer group can be classified as a LD. By introducing LD, migration within a LD can be handled using algorithms such as [6] and reduce handoff rate considerably. Another advantage of having LD is that connection-rerouting process is transparent to those receivers within the LD since they are not required to participate in the rerouting process. In Fig. 3(a) and (b) we see two LDs running two different multicast protocols. MCoRe assumes the existence of a PR within each LD. The PR of a LD can be the MCS or core (or rendezvous point) if shared tree is used. The role of the PR will be explained in the next section.

4.1.1. Proxy root

The PR handles all join requests on behalf of receivers within the domain. Further, the PR keeps track of receivers within its domain and administers the setting up and tearing down of connections. PR is programmable and responds to signaling messages sent as part of the MCoRe protocol. Depending on the multicast used, the PR may be setup dynamically or administratively. For example a PR might be set to the MCS, which is handling the given multicast session. The process by which a PR is chosen within a LD is beyond the scope of this paper.

The PR must have the following functionalities in order to support the MCoRe protocol:

- Send ADD_PARTY message: to initiate a join to a multicast session. Source address of the multicast session is obtained from a session directory.
- Process join requests from receivers: new join requests are not forwarded beyond the LD if the PR is already subscribed.
- Tear down connections: if no receivers are subscribed within the domain, a prune message is sent to its parent node. This means the PR has to keep track of receiver(s) within its domain since a MH may migrate without explicitly unsubscribing from the session.
- Respond to HO_UPDATE message: upon receipt of this message the PR updates its mobility binding by recording the source's current care-of-address. Once the mobility binding is updated, the PR then sends a TREE_UPDATE message back to the source. If the PR is an AS then the operations outlined in Section 4.4 are performed. Otherwise the TREE_UPDATE message is simply forwarded on the VC allocated (leading to the parent node) for the multicast session.

4.2. Tree creation and maintenance

In this section we outline how PRs join the multicast tree. In MCoRe, the SRT is bi-directional and a single VC is used for receiving and sending. Hence receivers are not required to setup connections back to the source in order to send acknowledgements for received data. Each connection on the multicast tree is constructed using a single VC. This conserves resources and reduces the number of connections involved during connection rerouting. One of the main problems with the use of a single VC is cell interleaving, which occurs when cells from receivers are interleaved with cells from other receivers/source(s) at merge points (i.e node degree greater than one). The problem of cell interleaving has been addressed in Refs. [19,10].

Each switch subscribed to the multicast session maintains the information shown in Table 1. Basically each switch knows which of its immediate downstream ASs are subscribed to the multicast session. Note that the number

Table 2
Explanation of variables used in Algorithm 1

Variable	Details
opRM_seq	The RM sequence number received from the optimized path
OpCellCounter	The number of cells received from the optimized path
extRM_seq	The RM sequence number received from the extended path
RMs_EQUAL	Flag to indicate whether RM sequence number between extended and optimized path has been achieved.
ExtCellCounter	Number of cells received from the extended path

of subscribers maintained for a given AS is minimal. This is because states for each receiver belonging to a multicast group are distributed among ASs in the network. From the viewpoint of an AS, it only knows a given downstream AS has subscribers and it does not need to store information of all subscribers at the downstream AS. During optimization, ASs distinguishes between newly created VC and old VC on the old tree to prevent cells from being forwarded on the optimized VC. VCs, which belong to new and old trees, must be distinguishable in order to prevent loop(s) from occurring. We will show later in Section 4.4 how the optimization process takes advantage of the *NConn* variable and how different VCs are differentiated.

4.2.1. Initiating a multicast session

To start a multicast session, the source registers its intention with the PR of its LD by sending a packet with the following tuple *{source permanent address, multicast address, services}*. Note that the source's permanent address (i.e. home address) is used. When receivers interested in joining the multicast session transmit join requests, the source's permanent address is returned during multicast address resolution. Therefore the receiver requests will be directed to the source's home network. If the source has left its LD then a care-of-address is returned in the release message. The corresponding receiver then uses the care-of-address in a subsequent join request. The problem of location management has been addressed by Acharya et al. [20]. The approach taken by Acharya et al. [20] is similar

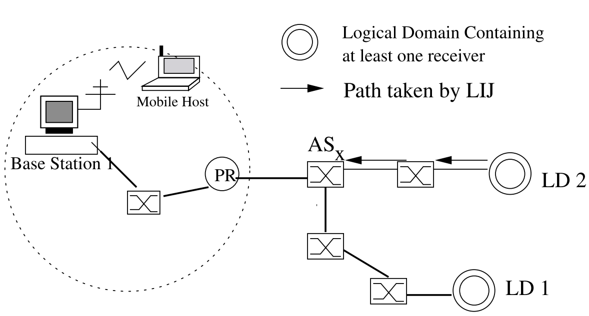


Fig. 4. Illustration of Join Process.

to that in Mobile IP [1] where a home agent manages all MHs that have migrated from its domain. The difference here is that a MARS service is required to provide a mapping between multicast address and source's home address before Archarya et al. [20] scheme can be employed.

Upon receipt, the PR relays the registration to a multicast directory server (MDS). Here MDS refers to server that maintains a mapping between multicast address and the source's ATM address, for example MARS. Here, we assume multiple MDSs exist which provides distributed mappings of multicast addresses. When MDS confirms the registration, an acknowledgement is sent to the initiating PR, which is relayed to the source. Unlike MCS [5], the PRs in MCoRe are not required to maintain a connection with the MDS for any information regarding group membership. Therefore in our MCoRe model, the source is neither aware of the multicast size nor is it required to add receivers one at a time. This is because the management of multicast group members has been implemented in ASs and PRs are notified of the source's care-of-address.

After registration, the source's PR waits for join request(s) from receiver(s). When a join request arrives, the PR sets up a new connection to the source if one does not exist. Then an active setup message augmented with an active tag is sent to the receiver. The active tag is used by each AS to load the MCoRe program. The MCoRe program then creates state information as shown in Table 2. The MCoRe program controls VCs, signals and maintains states pertaining to the given multicast session. After the connection is setup the source starts transmitting. Readers interested in AS architecture are referred to Refs. [21,22].

4.2.2. Join rocess

The signaling process in which receivers join the multicast session is adapted from ATM's native multicast mechanism, specified in UNI 3.0 [3] and UNI 4.0 [4]. Note that due to the limitations of the native ATM multicast mechanism, modifications to the signaling messages are required. These modifications allow the programming of ASs, which enable bi-directional connections and installation of necessary states for connection rerouting.

A PR interested in joining a multicast session first makes a query to the MDS. In response, the MDS returns the multicast session source's ATM address. A LIJ message is then sent to the source containing the multicast group identification. Each AS checks whether it is subscribed to the multicast group specified in the LIJ message. If so a setup message is returned, which results in a new connection (branch) to the LD initiating the join request.

In Fig. 4, we see that LD_1 has joined the multicast session. When the join request from LD_2 is intercepted by AS_x, a setup message is sent to the PR of LD_2. At AS_x, a new entry is then added to the multicast table (see Table 1) reflecting

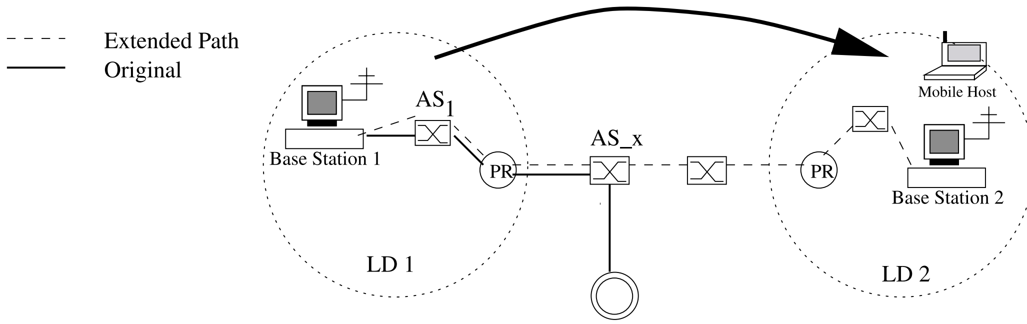


Fig. 5. Handoff with partial optimization.

the new connection to LD_2. Once added, incoming cells from the ParentVC are multicast accordingly. Thus the latency of join process is the time for the join request to encounter an AS subscribed to the multicast session plus the connection setup time.

In a LD where no proxy root is assigned, the adjacent AS (programmed with MCoRe) takes the role of the PR. The receiver does not need to be equipped with the MCoRe protocol. This is because the intermediate switch from the receiver is already programmed with the MCoRe program. One of the main advantages of ANs is dynamic augmentation of functionalities. When a receiver migrates to a LD that does not support multicast, ASs within the LD can be programmed with the MCoRe protocol. Subsequent receivers migrating into the LD will be able to utilize the programmed ASs. In this scenario, one of the ASs programmed within the LD will have to be selected as a PR. For simplicity the designated PR is the AS located at the boundary of the peer group.

An exception in the joining process is that join messages originating from receivers in the same LD as the source must not graft a new branch on the connection that connects the source to the PR. When the source migrates, the connection from the source's previous location to the PR will be torn down. As a result all receivers grafted on the branch will be dropped from the session. Alternatively we could have the PR perform an update on the source's old connection after migration. But since we assumed that the rerouting process within LD is independent of MCoRe, the update of connections is left to the rerouting protocol employed in the domain.

4.2.3. Leave process

The leave operation is performed by sending a prune message. When the PR detects that it has no receivers subscribed to a given multicast session it sends a release message on the outgoing VC for the multicast session (i.e the connection leading to the parent node). At the merge point, upon receipt of the release message, the AS removes states allocated for the given subscriber.

Here we assumed that the PR has a mechanism for detecting whether it has any receivers subscribed within its LD. This is crucial when receivers are mobile. A MH may migrate out of the current LD without explicitly tearing down its connection to the PR. A solution to this may involve a soft-state VC approach. Each VC is associated with a timer, which is refreshed each time cells are received. Once the timer expires, the corresponding VC is torn down. The advantage of using soft state is that no additional signaling needs to be invoked to tear down connections.

When tearing down connections, care must be taken because unlike connection rerouting in unicast communications, an old connection cannot be torn down without first checking whether there are receivers relying on the connection for data.

4.3. Handoff

The handoff process in MCoRe is a simple path extension process. This means that a new connection is setup from the source's new location to the source's old location. It has been shown in [23] that path extension has a low handoff latency, maintains cell ordering and has minimum cell loss. There are two main disadvantages with path extension: increased end-to-end delay and likely formation of loops. The problem of end-to-end delay will be addressed in the next section. As for preventing the formation of loops an active solution has already been proposed by Chin et al.[24]. Therefore we will adapt their partial optimization scheme into MCoRe with slight modifications to handle multicast connections.

To illustrate the partial path optimization process consider Fig. 5. When the source migrates to LD_2 it sets up a path to BS_1. From Fig. 5 we see that the extended path can be short-circuited at AS_x, given that there are no receivers in LD_1 and on the branch leading to LD_1. If there are receivers in LD_1 then the path is short-circuited at the PR for LD_1. This is the reason why receivers in the same LD as the source are not allowed to create a branch on the connection between the source and PR. The partial optimization process is performed after the extended path has been established. Note that the partial optimization process is performed automatically by ASs without the knowledge of PRs. Moreover no additional signaling is generated to

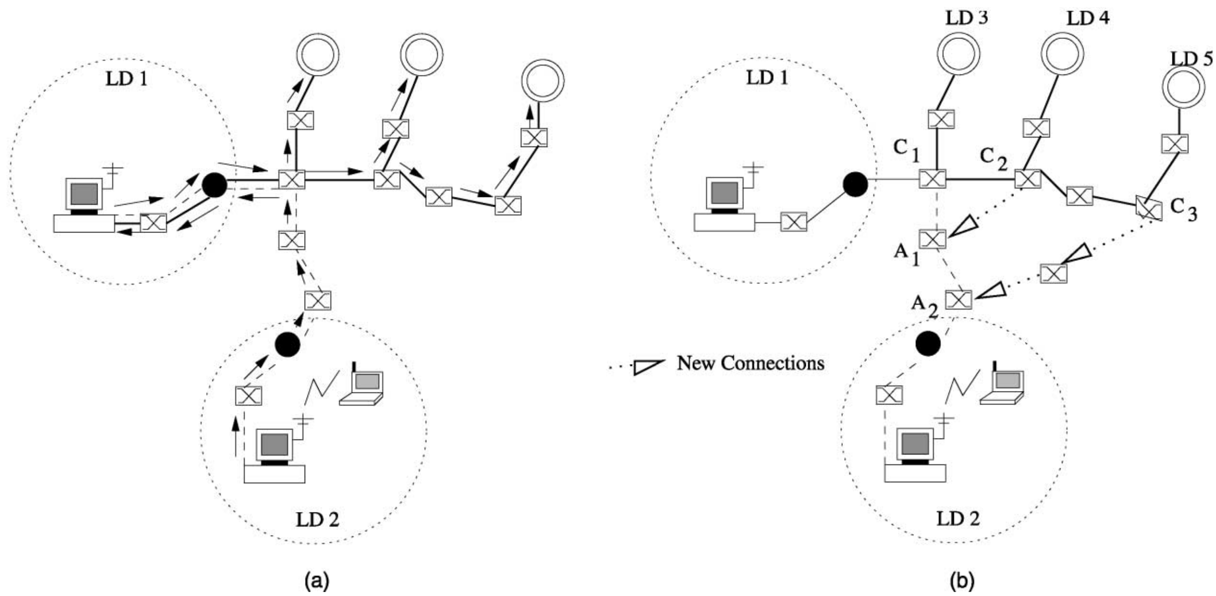


Fig. 6. (a) Shows the flow of HO_UPDATE message and (b) new connections created after transmitting TREE_UPDATE

invoke the optimization process. The only signaling involved are those used for tearing down connections.

The process in which the path is partially optimized is as follows. Firstly a check is performed by PR of LD_1 to determine whether there are any receivers. If there are receivers subscribed then the connection $PR–BS_1–AS_1–PR$ is torn down after cells from the path are drained. Thus the resulting path ranges from BS_2 to PR of LD_1. The tearing down process is similar to those in Refs. [25,24]. If there are no receivers subscribed, a PARTIAL_OP is sent by the PR on the extended path. At each AS, a check is made to determine whether the extended path has diverged from the current AS. From Fig. 5, we see that the extended path has diverged at AS_x, hence AS_x is chosen. In general, the AS with a degree more than one is chosen. This differs from Toh's [6] scheme where a snapshot of the multicast tree is required.

The computation for determining whether the path has diverged is straight forward. Since downstream AS(s) are subscribed to the multicast session we can compute a path to the source's current location. This is similar to PNNIs [16] computation of designated transit list (DTL). Here we compute the DTL of the source's current location. We then check whether the next hop matches one of the entries in RD_i. If the next hop is not the parent node, the AS performs connection rerouting, hence draining and tearing down the original path from the AS to BS_1 and the extended path. In case the AS encountered is a branch point such as an AS with degree greater than one, referred to as AS_{br}, then the partial optimization process is performed by AS_{br}.

4.4. Optimization

The optimization process is initiated by PR of the LD in which the source is in. To optimize the multicast tree a HO_UPDATE message is multicast by the source (see Fig. 6(a)). The HO_UPDATE message contains the source's current care-of-address. The main purpose of the HO_UPDATE message is to notify PRs of the source's care-of-address and begin the optimization process. When PRs receive the HO_UPDATE message, they note the source's care-of-address and transmit a TREE_UPDATE message. The TREE_UPDATE contains the source's address and is used by ASs to compute the corresponding DTL. Upon receipt of the TREE_UPDATE, the AS determines whether there is an optimal route to the source's current location.

In Fig. 6(b) we see new connections created by the TREE_UPDATE message. After a PR has sent the TREE_UPDATE message, each AS computes a DTL using the source's address within the TREE_UPDATE message. The route obtained is then used to determine whether the next hop is the parent AS. If so, then no new connection setup is required. For example, ASs after C_3–LD_4 do not have to create a new connection because the TREE_UPDATE message is being forwarded to the parent AS. If the next hop is not the parent AS then a new connection is setup, for example C_3 in Fig. 7(b). At C_3 the AS determines from the computed DTL that the next hop is not the parent AS. Therefore it sends a setup message to the source with the MCAST_TYPE field set to NEW_TREE.

As a result a new connection is setup[2] towards the source. ASs that create new connections or have new connections incident on them are referred to as core ASs. In Fig. 7(b), we see that both C_2 and C_3 have created new connections to the source's current address. At C_2 and C_3 these new connections are marked as NEW.

This is to prevent cells from being forwarded to any VCs marked as NEW and the occurance of loops. Note that C_1

[2] Given the required QoS requirements are met.

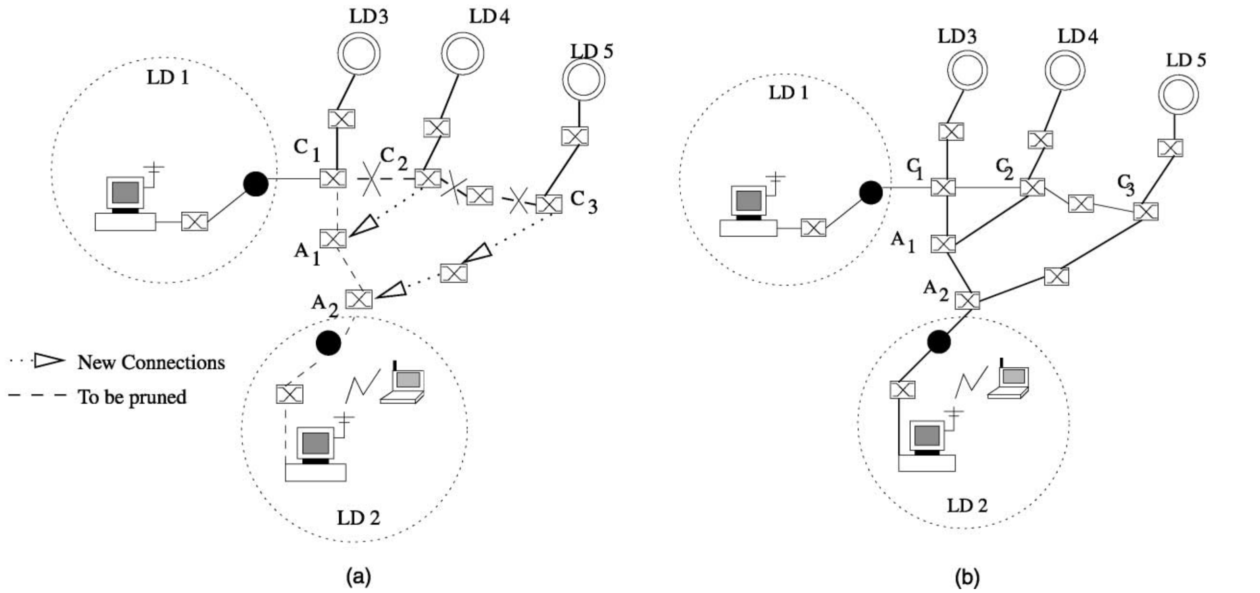


Fig. 7. (a) Shows the connections to be pruned after receiving marker cell. (b) Resulting tree after optimization. (a) Connections pruning, (b) Tree after optimization

needs to update its multicast table during optimization because currently its parent node variable is referred to LD_1. Here a new connection is not required because the extended path is reused.

When the setup message reaches an AS (e.g. AS_1 and AS_2 in Fig. 7(b)) subscribed to the multicast session, the AS proceeds to merge the connection. The connections are merged using the following policy. If the AS contains a state for the extended path, in other words the extended path passes through it, then the connection is merged with the extended path.

The newly formed branch is then marked as NEW. Once the connections are merged, the setup process for the new connection is terminated. A TREE_OP_ACK is then generated and transmitted to the source and the initiating core AS is informed of the successful creation of an optimized path. This confirms the optimization process from the corresponding LD. Note that the TREE_OP_ACK messages are dropped by an AS if the AS has previously processed the TREE_OP_ACK message. This reduces the amount of signaling involved, thus resulting in a scalable solution.

In Fig. 7(b) we see that AS_1 and AS_2 have merged connections from C_2 and C_3 onto the extended path. At this point, cells are still being forwarded on the old multicast tree. Note that these cells are not routed onto the new connections at AS_1, AS_2, C_2 and C_3.

Once a core AS has setup a new connection, it sends a PRUNE_RDY message upstream. This message tells the upstream ASs to mark the VC in which the PRUNE_RDY message arrived on to be torn down after a MIG_CELL is received and sent. In Fig. 7(a), C_3 sends a PRUNE_RDY message upstream. Note that this message is only sent on the link leading to the parent node. All ASs such as C_3 and C_2 that are not designated as core mark the incoming VC as PRUNE. The PRUNE_RDY message from C_3 is terminated at C_2 and that from C_1 is terminated at AS_1.

4.4.1. Tree transition

Once a new connection has been setup, the source can proceed to update the multicast tree. To preserve cell ordering all cells from the old tree must be drained first before new cells arriving on the new path can be forwarded. In order for a smooth update where cell ordering is preserved, MIG_CELL is multicast on the old tree. The MIG_CELL marks the end of cells flowing on the old tree and signals core ASs to update their multicast state. Also, core ASs tear down connections marked PRUNE. Furthermore, MIG_CELL is used to determine whether buffering is required at the core ASs.

As the MIG_CELL traverses the old tree, each core AS receiving the marker cell unmarks VCs that have been previously marked NEW. As a result cells will be forwarded on the unmarked VC(s). Cells arriving on 'new' connections before MIG_CELL are buffered. For example in Fig. 7(a), cells may arrive at C_3 before the marker cell is processed by C_3 since the MIG_CELL has to travese through the sub-optimal path. Once the marker cell is processed, buffered cells are forwarded ensuring preservation of cell order during the transition process.

Upon receipt of MIG_CELL, connections previously marked as PRUNE are torn down. Note that the connections are torn down after the MIG_CELL has been multicast. This enables downstream ASs to tear down any marked connections. The pruning process avoids the problem of introducing duplicate cells during the transition process. After forwarding the MIG_CELL to downstream subscribers, the core checks whether the incoming VC in which, the MIG_CELL was received is marked PRUNE. If so then a release message is sent and the VC is torn down. In Fig. 7(a), we see the corresponding connections that are marked PRUNE and torn down by the corresponding core after processing of MIG_CELL. The resulting tree after pruning down of connections is shown in Fig. 7(b).

In the unlikely case where the MIG_CELL arrives before the "new" connection has been setup, a prune message is sent to the parent node once the "new" connection has been setup and synchronization of cells have been performed. Once the parent node receives the prune request, it removes the subscriber from its multicast subscribers list. Before the old parent connection is removed, synchronization of cells arriving from the "new" and parent connection has to be performed. This is to ensure that receivers downstream do not get duplicate cells. The synchronization process will be explained in Section 4.5.2.

4.4.2. Updating multicast state

Apart from tearing down of connections, the states pertaining to the multicast session are updated. At each AS, new connections marked as "new" are unmarked. Refering to Fig. 7(a), AS_1 and AS_2 have connections marked as "new" incident on them. Upon receiving MIG_CELL, AS_1 and AS_2 update their multicast state and clear any connection marked NEW. As a result, subsequent cells will be multicast along the new link. Consequently C_2 and C_3 may receive cells from the optimized connection before receiving MIG_CELL. In this case the cells from the optimized connection are buffered until the arrival of MIG_CELL. Note that the extended connection is reused.

During the processing of MIG_CELL, a check is necessary to determine whether the MIG_CELL arrived from the extended path or from the parent node. If the MIG_CELL is at an AS with only extended path information then a new multicast state is created (see Table 1). States from the extended connection are then incorporated into the newly created multicast state. Any "new" connections are also incorporated.

At an AS where multicast table exists and the MIG_CELL does not arrive from the parent nodes no update is performed. This is because updating multicast table would cause cell loss and infinite buffering. Furthermore if the ASs has connection marked for pruning, these connections will be prematurely pruned. As a result the MIG_CELL does not get forwarded when it arrives from the parent node since the connections leading to downstream receivers have been pruned. This results in infinite buffering at ASs downstream.

A buffer limit is set at each AS, while waiting for the arrival of MIG_CELL. This is to ensure that infinite buffering does not occur when MIG_CELL is lost. Setting a buffer limit provides for a simple solution since it can be determined based on the available resources at a given AS.

4.4.3. When does the MIG_CELL gets sent?

We have mentioned the role played by the MIG_CELL but have not discussed when tree transition should take place. The transition process can be invoked when the PR of LD_2 receives a TREE_OP_ACK message. The side effect of this is that some connections may not be setup thus causing cell loss. For example, a neighboring TREE_OP_ACK message is bound to arrive sooner than those LDs, which are a few hops away. Alternatively, we could wait t amount of time. Note that the waiting time does not result in cell loss because cells are still being multicast on the old tree. The only side effect is that additional resources (connections) are held up in the network. A more complicated approach involves estimating the number of LDs. Once the TREE_OP_ACK of all LDs have been received, the transition takes place. Naturally this will incur additional signaling overheads.

The approach taken in our implementation is to send a MIG_CELL once the PRUNE_RDY and TREE_OP_ACK messages are received. Since a prune message is only generated once a new connection has been established and the message has to traverse through the elongated path (up to old source's point of contact and extended path), the time taken by the prune message provides a simple way for all core ASs to complete any optimization required before an update is made. By waiting for the PRUNE_RDY and TREE_OP_ACK message, we do not need to keep track of multicast group members. For example informing upstream parent node about the number of subscribers (or core ASs) downstream. This means that periodically signaling messages are flooded to the source of the multicast tree. For these reasons, we chose to make MCoRe scalable by waiting for the PRUNE_RDY and TREE_OP_ACK messages only. We conceed that this method is not reliable in that for some cases both messages may arrive sooner before "new" connections are setup properly. Note that this is not in any way detrimental to the transition process. Because synchronization is done at the core AS if a MIG_CELL is received before a "new" connection is setup. Further, since a prune is only sent after synchronization is performed no cell loss occurs. Also note that buffering is required to ensure cell ordering. We will show later that the buffering required is minimal.

4.5. Receiver migration

So far we have only considered the migration of source. In this section we show how connections to receiver are rerouted. The main problem of receiver migration is that of cell ordering. Assume that cells have sequence numbers incorporated, at the previous LD the MH has received cells with sequence numbers up to i. When the MH migrates to a

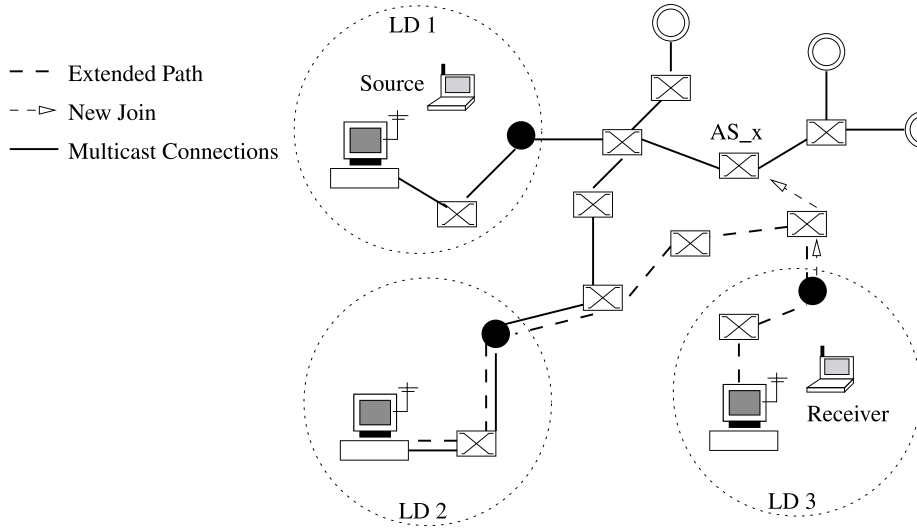


Fig. 8. Shows the handoff process of a mobile receiver.

new LD and rejoins the multicast session, in the absence of a scheme such as MCoRe, the MH may receive cells with sequence numbers less or more than *i*. The problem is made more difficult since ATM cells do not have sequence numbers. Therefore ASs is unable to determine whether a receiver has received a particular cell by looking at the header unless there is a global mechanism for distinguishing cells. To overcome the aforementioned problems the source (or the PR of the source's LD) sends RM cells every *k* cells. Included in each RM cell are: sequence number and *k* (the number cells preceding the RM cell).

4.5.1. Handoff

As in source migration, to avoid cell loss and preserve cell ordering during handoff, a simple path extension is made. This is illustrated in Fig. 8. After migration, the MH sends a rejoin request to the local PR. Once receiving the rejoin request from the receiver, the new PR sets up a connection to the receiver's previous location. Once the connection has been setup, the handoff process is finished. At this point in time, the receiver is receiving cells from the extended path, therefore from the receiver's point of view it is still in multicast session. The next phase is for the PR to send a rejoin message to the multicast session specified in the rejoin request. If the PR has already joined the multicast session then no rejoin is required. In this case the synchronization process outlined in the next section is performed.

The information available in the rejoin message sent by receivers are shown in Table 3. Once the AS receives the rejoin message and the QoS conditions are agreed upon, a new branch is setup to the intiating PR.

Table 3
Information within the rejoin request

Data	Description
McastGrp	The multicast group to rejoin.
RM_seq	The last received RM sequence number at the previous LD.
Old_COF	The previous care-of-address.

4.5.2. Synchronization

As a result of rejoining the multicast session, the PR starts receiving cells from both the newly created branch and the extended connection. Upon receiving cells from the new branch, the PR prepares to tear down the extended connection. To do this it must ensure that there is a seamless transition of cells from the extended and new connection. To illustrate the synchronization process at the PR consider Fig. 9. The PR has two connections incident on it and is receiving cells from both connections, new and extended. The RM cells from both connections are used to determine when the PR should switch connections. When a RM cell is received from the new connection all subsequent cells are buffered. Once a matching RM cell from the extended connection is received, the PR switches connection and starts forwarding the buffered packets. All subsequent

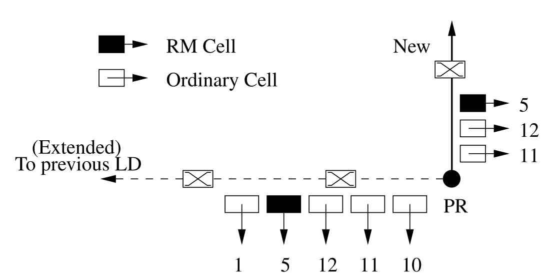


Fig. 9. Illustration of cell synchronization at PR.

cells from the extended connection are discarded and the extended connection is torn down.

In the case where the next RM cell number received from the new connection is less than that of the extended connection, cells from the extended path are forwarded until matching RM cell from the extended path is received. After that, the extended connection is torn down. Cells from the new connection are forwarded only when the RM sequence number is equal to cells from the extended path. An alternative solution may involve in maintaining the extended path but this introduce the possibility of loops. The maintenance of extended path is useful in LDs with no multicast capabilities. If the receiver discovers that the LD does not support multicast then the extended path is maintained until it migrates to another LD, which does. The algorithm for the synchronization process is shown in Algorithm 1.

Algorithm 1: Cell synchronization.

```
IF RM cell THEN
{
  IF RM cell from new link THEN {
    opRM_seq = Get_RM_sequence_number
    opCellCounter = Get_RM_cell_counter
  }
  END
  IF RM cell from Extended link THEN
    extRM_seq = Get_RM_sequence_number
  END
  IF opRM_seq ! = 0 AND extRM_seq ! = 0 THEN
    {
      IF opRM_seq > extRM_seq THEN
        set BUFFER_REQUIRED true
      ELSE IF opRM_seq < extRM_seq THEN {
        Send STOP_SENDING signal to
        previous BS
        set Discard_From_Ext true
      }
      ELSE {
        set RMs_EQUAL true
      }
    }
} ELSE
{
  IF cell from extended path AND NOT
  Discard_From_Ext THEN
    extCellCounter + = extCellCounter
  IF RMs_EQUAL THEN
  {
    IF opCellCounter > extCellCounter
    THEN
      set BUFFER_REQUIRED true
    ELSE IF opCellCounter <
    extCellCounter THEN
      Remove cell from extended path
    ELSE
      Optimization process done
  }
  IF BUFFER_REQUIRED && (opRM_seq ! =
  extRM_seq ı opCellCounter ! =
  extCellCounter) THEN
  {
    IF cell from optimized path THEN
      Buffer cell
    ELSE
      Forward cell to receiver
    }
  ELSE
    Forward buffered cells to receiver
}
```

Apart from the handoff procedure described above, a fast handoff is achieved by including the last RM sequence received as part of the path extension process. When a path extension is initiated, a join message containing the next cell sequence the receiver is expecting is sent to the local PR. If the PR has the next expected cell (assuming the PR is already subscribed) then the PR forwards the cell and the path extension process is terminated. This results in conservation of resources.

As mentioned above, RM cell is used for synchronization. The downside is that if the application decides to define k to be a large value then the synchronization process may take considerably longer before another RM is received. To enable a faster synchronization, AS and PR keep track of a counter call *CellCount*. This counter is incremented each time a normal cell is received and reinitialized to zero when a RM cell arrives. During the synchronization process if both the RM cells received from the newly joined connection and extended path have the same sequence number then the *CellCount* is used. As part of the rejoin acknowledgement, the acknowledgement message contains the *CellCount* value of the AS that intercepted the rejoin request. Once the PR receives the *CellCount* value it compares with the *CellCount* from the extended path (given the RM sequence number is equal). Then the PR proceeds to cell synchronization as in the RM case.

5. Simulation methodology

To investigate the performance of MCoRe, we used the NIST ATM/HFC network simulator [26]. With the NIST ATM simulator, routes between two end-hosts have to be hard-wired, no call setup mechanisms and no mobility support are available. So we augmented the simulator with the link-state routing protocol and a simple call setup protocol was implemented. The call setup protocol does not involve any call admission algorithms. During a call setup, a VPI is allocated for the connection followed by the setup of translation table. If the translation table has already been setup, then it is updated with the connection's

Table 4
Input Parameters

Data	Description
VPI	The virtual path identifier
InPort	Incoming port
OutPort	Outgoing Port
Conn-Type	Connection type (e.g. CBR)

information shown in Table 4. As for mobility support, since we are not concerned with the wireless link characteristics we modeled handoffs by severing a particular link and restoring another. Once a particular link is up, the MH hears a beacon broadcast from the BS and assumes occurrence of a handoff. Here the BS is set to transmit on the link (maybe severed initially) leading to the MH once every second.

A mesh topology (degree of four) consisting of 36 ASs was used. The links interconnecting ASs are set to 155Mbits/s with a distance of 0.5 kilometers. The link connecting the MH is set to 2Mbits/s. The delay to process a cell at switches was set to one millisecond. The source of the multicast session uses the constant bit rate (CBR) service and transmits at m Mbits/s. The value m ranges from 5 to 50 depending on the experiment involved.

Each switch was injected (pre-loaded) with the MCoRe program to facilitate recognition of signaling messages pertaining to the MCoRe protocol and also to maintain necessary states. During the join process, the injected program is activated and after processing each join message the AS is considered subscribed to the multicast session. In our simulations only one multicast session is running at any given time. Furthermore all receivers join the multicast tree at the start of the simulations. In our simulations a total of 10 receivers were used corresponding to 10 LDs. We ran a total of 10 simulation runs for each experiment with the MH migrating randomly. In the experiments we recorded end-to-end latencies of receivers, signaling messages filtered and buffer requirements at the PR and ASs.

We compare the performance of MCoRe with that of CBT [14]. The core is manually positioned at the center of the topology with all receivers having an approximately equal distance to the core. The core remains fixed for the duration of the simulation. When the source migrates, a new connection is setup to the core and core proceeds to multicast any cells received from the new link. The old link is then torn down. In our implementation of CBT no connection rerouting algorithm such as [27] was implemented. Since the connection is reestablished after handoff, the path from the MH to the core is thus optimal although a high disruption time to receivers is observed. A possible solution is to employ connection rerouting methods such as those investigated by Bui et al. [28]. The use of connection rerouting and the adaptation of shared tree approach to handle mobile group members is beyond the scope of this paper and will not be discussed further.

6. Results

6.1. End-to-end latency

Fig. 10 shows the end-to-end latencies observed by each receiver before and after handoff. A significant increase in end-to-end latencies is observed after handoff due to the path extension process. Once the path has been extended, optimization is performed and the resulting end-to-end latencies dropped significantly. For some receivers the observed end-to-end latencies increase because the MH's new location maybe further away from where the MH was

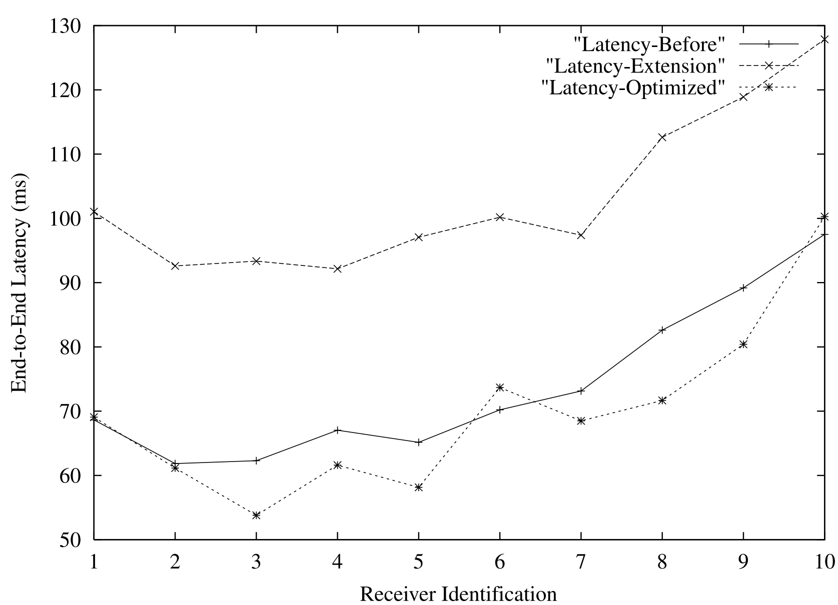


Fig. 10. Average end-to-end latencies.

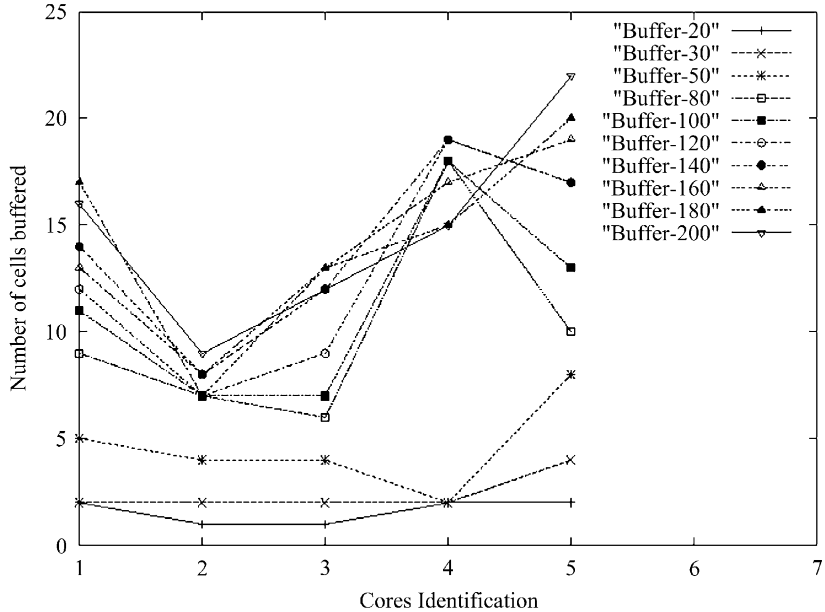


Fig. 11. Average buffering requirements at core Ass, with source transmitting at 10 Mbits/s.

previously. Therefore we see that in Fig. 10, receivers one, six and ten observed a higher end-to-end latency after optimization.

6.2. *Buffer requirements*

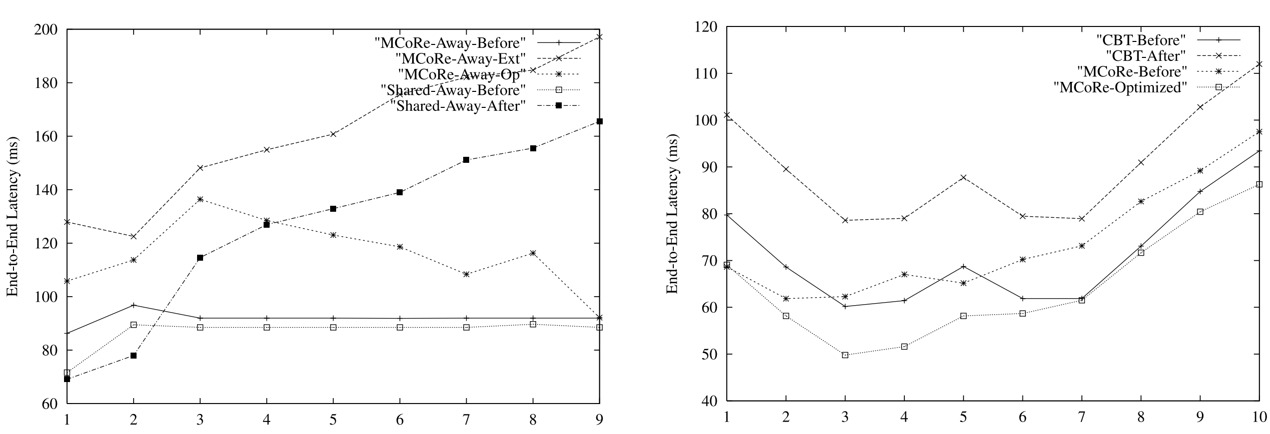
In this experiment we measured the buffering requirements at ASs during the optimization process. We plot the buffering required against different transmission rate at the source.

In Fig. 11 we see the number of cells buffered at each core given varying transmission rate. The buffering required is dependent on two factors: rate and the delay from a given core to the MH's current location. The delay from the core, the MH's current location determines the delay of the MIG_CELL. As a result this delay determines the number of cells buffered. If the source transmits at a higher rate, the number of cells buffered will be higher since the arrival rate of cells from optimal link will be higher. In our simulation environment, the average number of cells buffered at core ASs range from 2 to 23 when the source transmits at 10 Mbits/s.

6.3. *MCoRe vs. shared-tree*

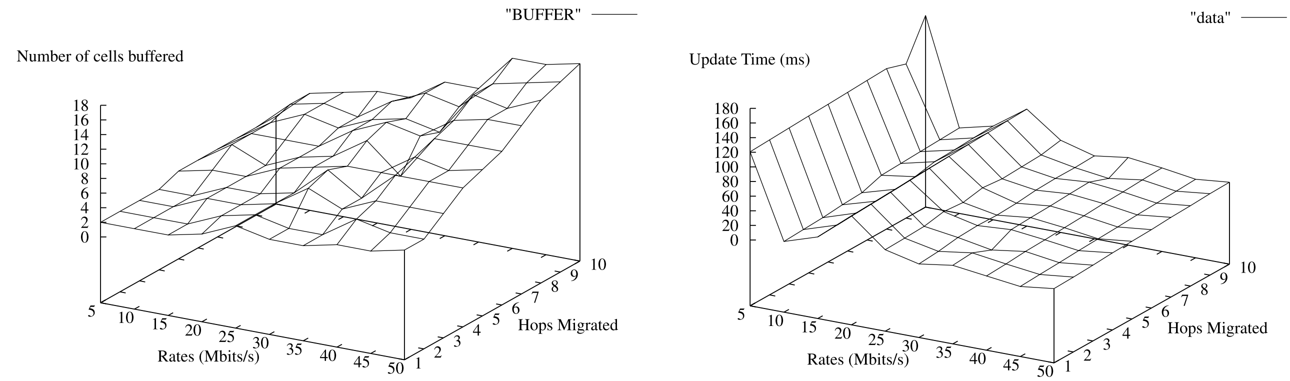
Here we present a comparison between CBT and MCoRe. Two experiments were performed. In the first, the MH is set to migrate increasingly away from its previous LD. The second experiment looks at the end-to-end latency in general for both MCoRe and CBT on varying topologies with receivers attached randomly.

In Fig. 12(a) we see that as the MH migrates further away,



(a) (b)

Fig. 12. (a) Effects on end-to-end latency when MH migrates further away. (b) Average end-to-end latency comparison between shared-tree and MCoRe.



(a) (b)

Fig. 13. (a) Buffer requirements given different rates (b) Tree update time given different rates.

the end-to-end latency increases with each migration. After migration we see that the end-to-end latency dropped significantly. For CBT we see that the end-to-end latency increases in a similar manner to that of MCoRe. Given that the core in CBT does not change after each migration, receivers are unable to improve on the observed end-to-end latency. This means that if CBT is used, the source has to stay within a certain range for the delay to be acceptable. In MCoRe the end-to-end latencies only persist, while the MH is receiving cells from the extended path. After optimization, the end-to-end

latencies dropped dramatically as shown in Fig. 12(a).

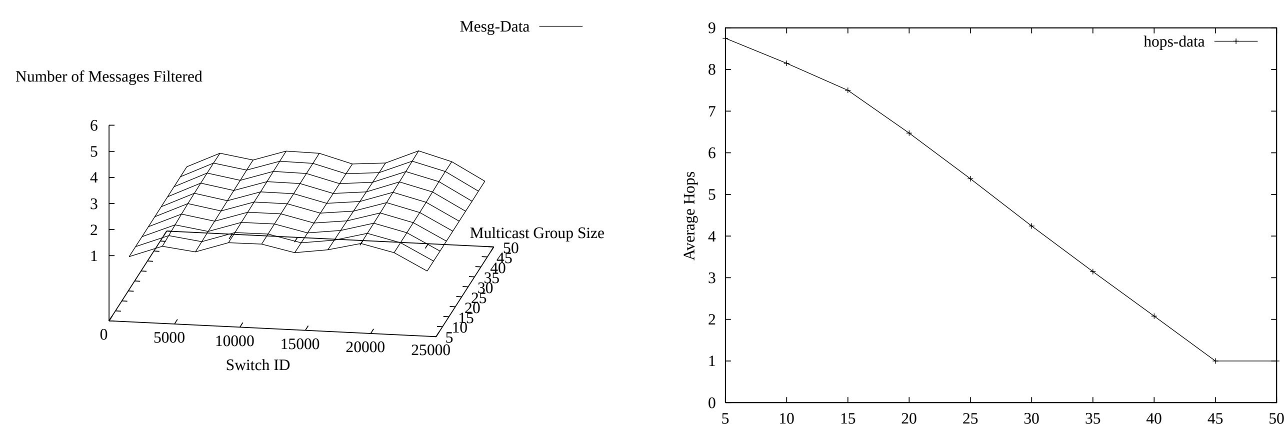
The comparison of the end-to-end latencies of CBT and MCoRe is shown in Fig. 12(b). On average the end-to-end latencies observed by MCoRe are lower than those of CBT due to the sub-optimal path between the source and receivers in CBT. Note that the use of connection rerouting algorithms only reduces the disruption time due to handoff and does not improve end-to-end latencies. This is because the position of the core is the determining factor in the perceived end-to-end latencies by receivers.

6.4. Receiver's migration

The performance under receiver's migration is dictated by the rate at which the multicast source transmits data and also how far the MH has migrated from its previous position. Other than that, the bandwidth available at the LD in which the MH migrated to affects the disruption time.

In Fig. 13(a) we see that with increasing rate the buffering requirement when the MH is 10 hops away is at a maximum (14 cells). This is due to the delay of the extended path where the PR has to buffer all incoming cells from the newly joined connection. The fact that MH migration is usually local only holds true when the migration is within a LD. During inter-LD migration, the extended path maybe traversed through a number of ASs. On the other hand, for local migration, we can expect that the extended path is relatively short. Therefore at hops ranging from one to four the buffering requirements range from zero to six.

We also investigated how long the synchronization process took with increasing rates. In Fig. 13(b) we see that at rate of 5 Mbits/s the update time is at its maximum.



(a) (b)

Fig. 14. (a) Number of messages filtered. (b) Average Hops traversed by signaling messages.

The update time increases when the MH migrates away from its previous LD. From 10 Mbits/s onwards the synchronization time is relatively stable and is only affected by the length of the extended path.

6.5. Scalability

In this section we looked at the role played by ASs in reducing the number of signaling messages involved. We ran experiments with increasing multicast group size (5–50) and recorded the number of TREE_UPDATE messages filtered during the tree optimization process. Further, the number of hops traversed by each message is recorded to determine how far a message has to traverse before being filtered. This means that if a message can be filtered as soon as possible then the traffic induced by signaling is reduced. In Fig. 14(b) we see that when the group size is at 45 or 50, the signaling messages are removed once they arrive at the first switch.

Fig. 14(a) shows the number of messages filtered as the multicast group size increases. Here we show the number of messages filtered by core ASs. For example we see that AS with an identification of 5000 has filtered five messages when multicast group is of size 50. Combining with other ASs (e.g. 5 ASs) assuming all of them have filtered out five messages, the total number of messages filtered is 25. Note that as the multicast group size increases the number of core ASs increases as well. Therefore more signaling messages are likely to be filtered. This is shown in both Figs. 14(a) and (b) where as the multicast group size increases, the number of messages filtered increases. In Fig. 14(b) we see that the average number of hops traversed by signaling messages is inversely proportional to multicast group size. As a result the signaling messages generated during handoff or optimization can be reduced dramatically.

7. Future research

In the following paragraphs we provide areas that requires further investigation.

- Multiple sources: As mentioned previously, currently MCoRe only supports one source. Given the data structure shown in Table 1 it is possible to have more than one source. The MCoRe program within ASs can be modified to multicast cells received from a given connection to all outgoing connections.
- Security: Currently MCoRe does not have security considerations incorporated and it remains to be investigated further. One of the main reasons for dismissing security at this stage is partly due to the immature nature of ANs securities technologies. Interested readers in ANs securities are referred to Ref. [29]. Our main concern is on rerouting multicast connections, the issues concerning securities are beyond the scope of this paper.
- Adaptive Shared-Tree: shared tree approach provides a simple solution to handling mobility but modifications are required to handle mobility. For example the core needs to be mobility aware. This means the core must contain a protocol in which it is able to perform handoff from an old connection leading to the mobile source's previous location onto the new connection to the mobile source's current location and vice–versa for receiver handoff. Furthermore, the core has to distinguish whether a new connection belongs to a source that just migrated or a new source. The handoff process can be improved if connection-rerouting algorithms are incorporated. The incorporation of connection rerouting becomes more difficult when rerouting connections leading to receiver. Due to the dynamic nature of multicast group members, after each source/receiver migration the core's position may cause some receiver's QoS to be violated. To overcome this problem we plan to look at the possibilities of applying active based solution in which a new core is computed given predetermined constraints.
- QoS: The issue of maintaining QoS after each MH migration requires further investigation. Some of the issues involve incorporating MH's QoS into the rerouting process and QoS adaptation. Current work in connection rerouting, both unicast and multicast, has not incorporated QoS policies.

8. Concluion

In this paper, we have discussed the problem of multicasting in mobile ATM networks. To solve the problems of rerouting multicast connections we have shown how intranetwork computation is utilized for a simple, efficient and scalable solution. As a result MCoRe enables connections in a SRT to be adapted efficiently due to source/receiver mobility. MCoRe preserves cell ordering, prevents cell duplication and has minimal cell loss. Our results have shown that MCoRe requires minimal buffering. Also the number of signaling messages is kept low with increasing multicast group size with the help of ASs. Moreover, compared to shared-tree approach MCoRe has lower end-to-end latencies and adapts to source migration. This means the multicast tree is always rooted at the source's current location whereas in shared tree the core is chosen during the initial setup of the multicast session and choosing another core after each host migration would prove to be an expensive process.

References

[1] C. Perkins, RFC2002: IP mobility support. Oct. 1996.

[2] B.A.J. Banh, G.J. Anido, E. Dutkiewicz, Handover re-routing schemes for connection oriented services in mobile ATM networks. In INFOCOM'98, (San Francisco, USA), 1998.

[3] ATM Forum, ATM User-Network Interface (UNI) Signaling Specification, Version 3.0, Prentice Hall, Englewood Cliffs, NJ, 1993.

[4] ATM Forum. ATM user-network interface (UNI) signaling specification, version 4.0, af-sig-0061.0001, July 1996.

[5] G.J. Armitage, I.P. multicasting over ATM networks, IEEE Journal on Selected Areas in Communications 15 (1997) 445–457.

[6] C.-K. Toh, A unifying methodology for handovers of heterogenous connections in wireless ATM networks, ACM Computer Communications Review 40 (1997).

[7] D.L. Tennenhouse, D.J. Wetherall, Towards an active network architecture, Computer Communication Review 26 (1996) 5–18.

[8] L. Wei, D. Estring., The trade-offs of multicast trees and algorithms. Internet Draft, Mar. 1995. draft-ieft-idmr-mtree-00.txt.

[9] S. Komandur, D. Mosse. SPAM: A data forwarding model for multipoint-to-multipoint connection support in ATM networks, in 6th International Conference on Computer and Communications Networks (ICCCN'97), (Las Vegas, USA), 1997.

[10] S. Komandur, D. Mosse. Performance comparison of CRAM, SEAM and SPAM multipoint VC schemes for ATM networks, in 7th International Conference on Computer and Communication Networks (ICCCN'98), (Louisiana, USA). 1998.

[11] R.R. Talpade, M.H. Ammar, Multicast server architectures for supporting IP multicast over ATM: in Proceedings IEEE ATM'96 Workshop, 1996.

[12] R.R. Talpade, G.J. Armitage, M.H. Ammar, Experience with architectures for supporting IP multicast over ATM, in Proceedings IEEE ATM'96 Workshop, 1996.

[13] M. Grossglauser, K.K. Ramakrishnan, SEAM: scalable and efficient ATM multicast. In Proceedings of INFOCOM'97 (Kobe, Japan), Apr. 1997.

[14] A. Ballardie, P. Francis, J. Crowcroft, Core based trees (CBT): An architecture for scalable inter-domain multicast routing. In SIGCOMM'93, (San Francisco), pp. 85–95, 1993.

[15] T. Teuijanto, S. Komandur, R. Cherukuri, A.G. Malis, Support for LIJ in PNNI. ATM Forum Contribution (atm961401R3), 1997. Montreal, Canada.

[16] ATM Forum. Private network–network interface specification version 1.0, Mar. 1996. af-pnni-0055.000.

[17] C.-K. Toh., Performance evaluation of crossover switch discovery algorithms for wireless ATM lans, in IEEE INFOCOM'96, International Conference on Computer Communications, (San Francisco), Mar. 1996.

[18] C.-K. Toh., The design and implementation of a hybrid handover protocol for multi-media wireless lans, in ACM First International Conference on Mobile Computing and Networking (Mobicom'95), (Berkeley, California), 1995.

[19] S. Komandur, J. Crowfort, D. Mosse, CRAM: Cell re-labelling at merge-points for ATM multicast, in IEEE International Conference on ATM (ICATM'96), (Colmar, France), 1996.

[20] A. Acharya, J. Li, B. Rajagopalan, D. Raychaudhuri, Mobility management in wireless ATM networks, in IEEE Infocom 98, (San Francisco, CA, USA), Apr. 1998.

[21] S. Min, H.K. Pung, L. Wong, A source-rooted multipoint-to-multipoint multicast in ATM networks, in Second International Conference on Information, Communications and Signal Processing, (Singapore), 1999.

[22] X. Inc. Xbind broadband source code. http://www.xbind.com.

[23] R. Ramjee, T.L. Porta, J. Kurose, D. Towsley, Performance evaluation of connection rerouting schemes for ATM-based wireless networks, June 1997.

[24] K.W. Chin, M. Kumar, C. Farrell, A model for enhancing connection rerouting using active networks, in The 2nd, ACM International Workshop on Modeling and Simulation of Wireless and Mobile System (MSWiM'99), (Seattle, Washington), Aug. 1999.

[25] H. Mitts, H. Hansen, J. Immonen, S. Veikkolainen, Lossless handover for wireless ATM, Mobile Networks and Applications (MONET) 1 (3) (1996) 299–312.

[26] N. Golmie, F. Mouveaux, L. Hester, Y. Saintillan, A. Koenig, D. Su, The NIST ATM/HFC network simulator. National Institute of Standards and Technology, U.S Department of Commerce, Dec. 1998. http://w3.antd.nist.gov/Hsntg/prd_atmsim.html

[27] P. Mishra, M. Srivastava, Effect of connection rerouting on application performance in mobile networks, IEEE Transactions on Computers 47 (1998) 371–390.

[28] B.A.J. Banh., Handover re-routing schemes for connection oriented services in mobile ATM networks, tech. Rep., Switched Networks Research Center. The Institute for Telecommunications Research, University of Wollongong, Australia, June 1997. http://www.snrc.uow.edu.au.

[29] D.S. Alexandar, W.A. Arbaugh, A.D. Keromytis, J.M. Smith, A secure active architecture, IEEE Network, Special Issue on Active and Controllable Networks (2001) in press..