IMSS: IP Multicast Shortcut Service*

Tal Anker David Breitgand Danny Dolev Zohar Levy

{anker,davb,dolev,zohar}@cs.huji.ac.il
http://www.cs.huji.ac.il/{~anker,~davb,~dolev,~zoher}
Institute of Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel

Abstract

In this paper we present IMSS*: IP Multicast Shortcut Service for ATM networks.* IMSS *pursues the "shortcut" routing paradigm in order to exploit the underlying ATM QoS and native routing mechanisms in the optimal way.*

IMSS *is built on top of* CONGRESS *(CONnection oriented Group-address RESolution Service).* CONGRESS *is an efficient native ATM protocol for resolution and management of multicast group addresses in a large ATM cloud.* CONGRESS *resolves multicast group addresses and maintains their membership for applications. It is not designed to handle the applications' data-exchange.*

An IMSS *extension at a multicast router uses the group membership information that it receives from* CONGRESS*, in order to open ATM connections that bypass ("shortcut") the IP routing mechanism.*

1 Introduction and Motivation

The advent of the ATM networking technology did not reduce the importance of the traditional IP networking. Great effort is put into the implementation of the IP protocol over ATM. The straightforward emulation of IP over ATM, however, results in sub-optimal performance due to the fundamental differences between the connection oriented and connectionless technologies. Some of the more complex problems arise in the implementation of IP multicast over ATM.

In this paper we present an IP Multicast Shortcut Service (IMSS), a novel conceptual solution for IP multicast shortcut (cut-through) routing over large ATM cloud. IMSS is aimed to improve the IP multicast services wherever possible by implementing the shortcut paradigm. The shortcut routing paradigm avoids the mismatch between the IP topology and the underlying ATM topology. We provide a classification of the generic problems that arise when various shortcut routing schemes are deployed. Current proposals for IP multicast over ATM shortcut routing suggest that the shortcut paradigm does not scale well to a large ATM cloud environment. In contrast to current proposals, IMSS overcomes scalability problems by deploying a dynamic mixture of multicast servers and full meshes of direct connections.

IMSS solves generic problems that arise when the shortcut paradigm is used, such as routing loops and datagram duplication. These problems are magnified when deploying shortcut routing mechanisms along with conventional IP multicast routing protocols. Applications and hosts using IP multicast enhanced with IMSS services need not be aware of the operation of IMSS.

The IMSS approach separates the problem of shortcut forwarding decisions from the problem of multicast address resolution and maintenance. In this paper we concentrate on the routing problems and the interoperability with the conventional Inter-Domain Multicast Routing (IDMR) protocols [9, 15, 11, 8, 16]. The multicast address resolution and maintenance problems are delegated to another service called CONGRESS: CONnection oriented Group-address RESolution Service [1]. CONGRESS achieves scalability through a design that is based on the following principles:

- **No flooding:** CONGRESS does not flood the WAN on every group membership change.
- **Hierarchical design:** CONGRESS services are provided to applications by multiple hierarchically organized servers.

Due to network failures and/or network reconfiguration and re-planning, some CONGRESS servers may temporarily disconnect and later reconnect. CONGRESS withstands such transient failures by providing a best-effort service to applications.

*This work was supported by the Ministry of Science of Israel, grant number 032-7658

1.1 Background and Related Work

The classical IP network over an ATM cloud consists of multiple *Logical IP Subnets (LISs)* [14] interconnected by IP routers. The only standardized solution for IP Multicast over ATM, Multicast Address Resolution Service (MARS) [4], follows the classical model. In the MARS approach, each LIS is served by a single MARS server and is termed *MARS cluster*[1]. Roughly, the purpose of the deployment of a MARS server is similar to that of IGMP [10] - to register the hosts that are directly attached to a multicast router and are interested to receive multicast traffic targeted to a specific IP class D address. The important difference, however, is that MARS is aware of the connection-oriented nature of the underlying network. For each relevant IP class D address, the MARS server maintains a set (membership) of the hosts that belong to the same LIS and have been registered to receive IP datagrams being sent to this address. The process of mapping an IP class D address onto a set of ATM end-point addresses is termed *multicast address resolution*. Each such set is used to establish native ATM connections between an IP multicast router and the local members of the IP multicast group. The IP multicast datagrams targeted to a specific class D address are propagated over these connections. The ATM connections' layout within a MARS cluster may be based either on a mesh of point-to-multipoint (*ptmpt*) Virtual Circuits (VCs), or a Multicast Server (MCS), or mixture of these two.

The classical LIS model assumes that hosts from different LISs always communicate through one or more routers. In the MARS model that retains the LIS model, it is assumed that all the multicast communication outside the LISs is performed via multicast routers that run some IDMR protocols. As explained in [18], the classical LIS model may be too restrictive for networks based on switched virtual circuit technology, *e.g.*, ATM. Obviously, if LISs share the same physical ATM network (ATM cloud), the LIS internetworking model may introduce extra routing hops. This mismatch between the IP and ATM topologies may preclude full utilization of the capabilities provided by the ATM network (*e.g.*, QoS).

In addition, the extra routing hops impose an unnecessary segmentation and reassembly overhead, because every IP datagram must be reassembled at every router so that the router can take routing decisions. The development of NHRP [13], a protocol for discovering and managing IP unicast forwarding paths over ATM that bypass IP routers, lead many people to seek for a multicast equivalent. Unfortunately, to provide an NHRP equivalent for multicast is not a trivial task, as will become clear from the next subsection.

[1] There is a work in progress to distribute the MARS server in order to provide for load balancing and fault tolerance [2]. A group of redundant MARS servers will constitute a single logical entity that would provide the same functionality as a non-distributed MARS server.

1.2 The challenges

A designer of a short-cut routing multicast solution is opposed with multiple non-trivial problems. The more prominent problems are discussed below.

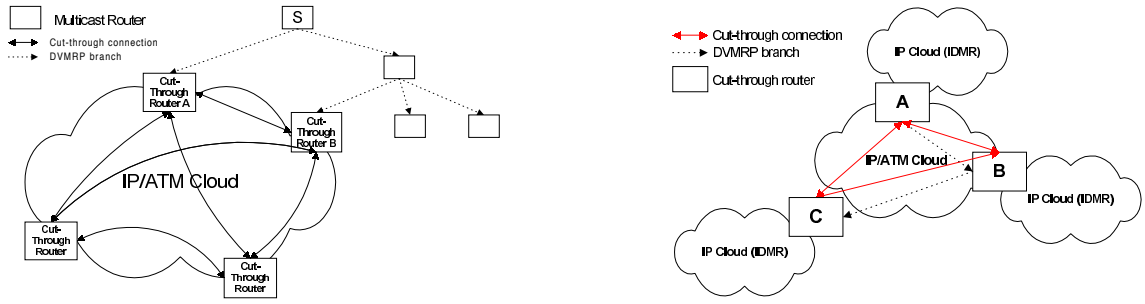
If hosts are allowed to communicate directly with other hosts (as in [3]), bypassing the multicast routers, then each host must maintain membership information about all other hosts scattered all over the internet and belonging to the same IP multicast group. This scheme does not scale well because:

- The hosts must maintain large amounts of data that should be kept consistent and updated.
- A considerable traffic and signalling overhead is introduced when membership changes, *e.g.*, join or leave events are flooded over the network.
- As was noted in RFC2121 [5], an ATM Network Interface Card (NIC) is capable of supporting a limited number of connections (*i.e.*, VCs originating from a NIC or terminating at a NIC). If a full mesh of ptmpt VCs is used for shortcut communication within a multicast group, NICs might not be capable to support all the simultaneous connections.

The IMSS solution presented in this paper performs shortcut only among the multicast routers, reducing the problems above to a certain extent. The NIC limitation problem is not completely eliminated, however. Hence, IMSS facilitates deployment of "multicast servers" for other routers that are termed "clients". In IMSS some of the multicast routers may also function as multicast servers.

Shortcut mechanisms may have a negative impact on the conventional IDMR protocols. For the sake of discussion of the interoperability issues with the IDMR protocols, we divide the IDMR protocols into two large families: "broadcast & prune"- based [9] and "explicit join"- based [11, 15, 8]. In the first model periodical flooding of the network and the subsequent pruning of irrelevant branches of the multicast propagation trees is employed. In the second model, some explicit information about the topology of the IP multicast groups is exchanged among the multicast routers.

As we see it, a shortcut solution will have to coexist with a regular IDMR protocol in the same routing domain. One of the reasons for deployment of an IDMR



(a) Datagrams Duplications

(b) Routing Loop

Figure 1: Inter-operability with IDMR protocols

protocol in addition to the shortcut mechanism, in the same ATM cloud, is that it is not guaranteed that a shortcut connections can reach all the relevant targets in the ATM cloud.

Another important reason is that if a "broadcast & prune" IDMR protocol is used in some non-ATM based IP subnetworks connected to the ATM cloud, the border routers that connect these subnetworks to the ATM cloud, do not receive explicit notifications that some downstream routers could be a part of an IDMR multicast propagation tree (as depicted in Figure 2). Thus, a broadcast & prune mechanism of the IDMR protocol should be exploited periodically by the shortcut multicast routers in order to learn about the downstream routers that depend on them. The discovery process is based on analysis of the prune messages that the multicast router will receive from the neighboring routers.

On the other hand, the co-existence of IDMR protocols with the shortcut solution, raises several problems:

- Routing decisions are normally made at the multicast routers. If hosts can bypass a multicast router, the latter should be aware of all the hosts in its own LIS (and in all of the downstream LISs) that participate in the shortcut connections. Otherwise the IDMR protocols would not be able to construct the multicast propagation trees correctly and the multicast datagrams may be lost.

- If a multicast shortcut mechanism is deployed in conjunction with some IDMR protocol, then conflicts with the Reverse Path Forwarding (RPF) [16] may occur. The RPF mechanisms prevent routing loops and are crucial for the correct operation of IDMR protocols. Thus, the shortcut traffic should be treated carefully in order not to confuse the IDMR protocol.

- A multicast distribution tree of an IDMR protocol may span non-ATM based IP subnetworks and contains more than one border router that connect these subnetworks to the ATM cloud as shown in Figure 1.a. If these border routers maintain the shortcut ATM connections to all other relevant border routers, undesired datagram duplication may result.

- Another scenario that may lead to routing loops and undesired datagram duplication, may arise when both a shortcut mechanism and some conventional IDMR protocol, are deployed in the same ATM cloud (see Figure 1.b). This means that an IDMR tree spans some routers within the ATM cloud and not only the border routers.

1.3 The IMSS Approach

In ATM networks a multicast group can be implemented either using a mesh of ptmpt connections, or a multicast server, or a mixture of these two techniques. Because of the connection-oriented nature of the ATM networking technology, the information about the physical addresses of the members comprising a multicast group (*membership*) should be available at connection setup time [6, 7]. This means, that prior to any data transfer, a group name (address) should be resolved into a set of the ATM addresses of the group members. In contrast to the conceptual solution proposed in VENUS [3], IMSS is not concerned with group membership maintenance and resolution. IMSS' scope is limited to making shortcut forwarding decisions and to forwarding of the multicast data among the multicast routers. Actually, IMSS is an optional extension to the multicast routing protocols as explained in Section 2.

IMSS organizes IP multicast routers into logical groups, where each group corresponds to some class D IP address and contains routers that have members of this IP multicast group or senders to it in their do-

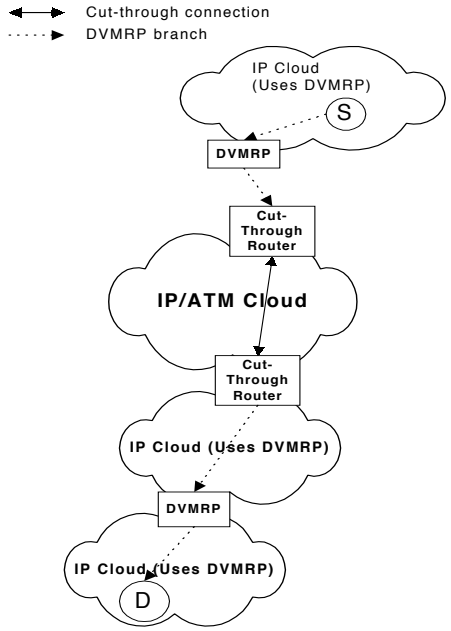


Figure 2: Unknown downstream routers

main. These groups are termed *D-groups*. D-groups will be further discussed in Section 2. The resolution and management of these multicast router groups is performed through the CONGRESS services.

An IMSS extension at a multicast router uses the group membership information that it receives from CONGRESS, in order to open ATM connections that bypass ("shortcut") the IP routing mechanism. Since the number of multicast routers is considerably lower than the overall number of the ATM-based destinations (both hosts and multicast routers), IMSS deals with a considerably lower number of connections from the very beginning. It may still be the case, however, that the number of multicast routers participating in a mesh of ptmpt connections is very large. Using the address resolution services of CONGRESS, IMSS can support both hierarchies of multicast servers and meshes of ptmpt connections, and to switch back and forth between these two layouts as required. This will be described in Subsection 3.2.3.

In order to avoid stable routing loops, an IMSS router never routes IP multicast datagrams using shortcut connections if they were received from another IMSS router. In addition, an RPF-like mechanism is deployed by IMSS in order to prevent the extensive duplication of IP multicast datagrams. Such duplication may result from multiple IMSS routers setting up multiple shortcut connections to the same destinations (see Figure 1.a).

We assume that IMSS will be used along with conventional IDMR protocols and that not all of the multicast routers will run IMSS within an ATM cloud. As was explained in Subsection 1.2, this deployment mode may lead to datagram duplication when a datagram is propagated over some multicast distribution tree and, simultaneously, over a shortcut connection (see Figure 1.b).

IMSS provides a pruning mechanism that cuts the branches of the IDMR multicast distribution tree, so that each branch would end at a non-IMSS multicast router. Note, however, that this mechanism may be employed only if neither of border multicast routers is running a "broadcast & prune"- based IDMR protocol.

2 IMSS Architecture

We differentiate between the two types of IP-multicast routers: a) routers that run some IDMR protocol and b) those that run both some IDMR protocol and the IMSS protocol. We refer to the latter routers as *border routers* or IMSS *routers*. A border router connects a *shortcut routing domain* to some IDMR routing domain(s). IMSS extends a multicast router's software. D-groups of IMSS are managed through CONGRESS. In order to make routing decisions and to open the shortcut connections, IMSS communicates with the CONGRESS protocol that supplies group address resolution and maintenance services.

CONGRESS is an efficient native ATM protocol for resolution and management of multicast group addresses in a large ATM cloud. CONGRESS resolves multicast group addresses and maintains their membership for applications. It is not designed to handle the applications' data-exchange.

The CONGRESS services are provided by a library that includes the functions that allow an IMSS router to *join* a multicast group, *leave* a multicast group, *resolve* a multicast group into a set of the ATM addresses of a multicast group members and *enable/disable* incremental membership notifications.

In the classical IP multicast model [10], a host does not have to become a registered member of a multicast group in order to send datagrams to this group. A sender does not see any difference between sending a datagram to a multicast IP address or a unicast IP address. The difference is in the multicast router, that has to participate in some IDMR protocol that builds a multicast propagation tree. In this model, a multicast router usually should know only about its immediate neighbors that belong to the propagation tree, and not about the whole tree[2].

[2]MOSPF [15] is an exception.

IMSS provides the hosts with the same interface for IP multicast service as in the classical model. An IMSS router, however, should know about all other IMSS routers that should receive the traffic targeted to a specific class D address. The set of these IMSS routers' identifiers that has to be maintained per IP class D address are organized by IMSS in D-groups. A D-group must include the IMSS routers that have either

1. directly connected hosts that registered (*e.g.*, using IGMP) to receive IP multicast traffic pertaining to a specific class D address, or

2. some downstream multicast routers that have receivers in their LISs.

Note that the sets of the downstream routers are different with respect to every possible source of the multicast traffic. Thus, a D-group corresponding to a class D address is a *superset* of the routers that should participate in shortcut connections. Not every member of a D-group communicates over shortcut connections with every other member. Additional per-source information needs to be maintained in the routers for constructing the actual shortcut connections. This is further explained in Subsection 3.1.

In order to obtain the membership of a D-group, an IMSS router joins this group via CONGRESS. The name associated with this multicast group is just a class D address interpreted as a character string. The details of how D-groups are formed and managed are provided in Subsection 3.2.

It may seem that an IMSS router that does not receive any report of downstream *receivers* by the IDMR protocol (neither routers, nor hosts) for a given class D address, does not need to be a member of a D-group because it does not need to receive any traffic. However, such a router might still have to inject multicast traffic for this group at some point. This router could use the CONGRESS *resolve* operation each time it needs to learn about the membership of the corresponding D-group. In this scheme, however, CONGRESS would be heavily used and extra overhead on the network would be imposed. In our approach, an IMSS router joins the relevant D-group even if it does not have to receive the multicast traffic. In this case, it will receive incremental membership notifications concerning the D-group. This scheme is less costly. In order to prevent such a router from being added as a **leaf** to the shortcut connections within the D-group, special sub-identifiers are added to the IMSS router's identifier.

In order to overcome the previously mentioned NIC's limitations on a number of simultaneously opened connections, some IMSS routers may act as multicast *servers*, serving other IMSS routers that are termed *clients*.

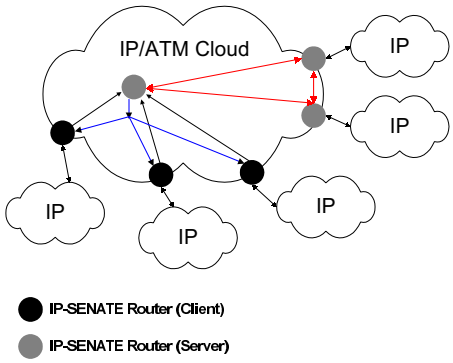


Figure 3: General layout of a D-group

The general layout of a single D-group is shown in Figure 3. It is important to stress that an IMSS router acting as a server in one D-group may act as a client in another one. Moreover, as will be explained in Subsection 3.2.3, the operational roles of the IMSS routers may dynamically change within the same D-group.

It is important to understand that maintaining a distinct multicast group simultaneously for every possible IP class D address is technically infeasible. Fortunately, there is no real need to do this, because only a part of these addresses is actually in use at any given time. In the IMSS approach, membership of *D-groups* is formed on-demand using CONGRESS, as will be explained in Subsection 3.2.

Another very important property of the IMSS solution is that IMSS can tear down the shortcut connections among the members of a D-group when no multicast data is transmitted over these connections for a sufficiently long period of time. The shortcut connections may be resumed later on-demand, using CONGRESS to obtain updated membership information. Note, that when an IMSS router terminates the inactive connections within a D-group, this does not affect CONGRESS which may continue to monitor the membership of the group running "in the background". Thus, when the shortcut connections need to be resumed, the membership information would be instantly available.

For a variety of reasons that were explained in Subsection 1.2, IMSS may have to co-exist with some IDMR protocol in the **same** ATM cloud. This implies that an IMSS router may receive IP multicast datagrams both via an IDMR protocol and the shortcut connections on the same network interface. For the correct operation of IMSS protocol, it is necessary to differentiate between these two cases. One way to do this is to

use the PROTOCOL field of the IP datagram header. An IMSS protocol should be assigned a special unique number. Each time an IMSS router forwards a datagram over a shortcut connection, the original protocol number is extracted and appended to the end of the datagram. The IMSS protocol number is inserted into the PROTOCOL field and all other relevant fields of the IP datagram header (total length, header checksum, *etc.*) should be updated appropriately. Obviously, the reverse operations should be performed by the IMSS routers on the other side of the shortcut connections. We will not elaborate this technique further in the paper, because this is strictly implementation-dependent issue.

3 Protocol

In this section we provide a detailed description of the IMSS protocol. We divide the protocol into two parts: a) D-groups' formation and maintenance, b) datagram forwarding decisions. For the sake of simplicity, we provide all the explanations for a single IP multicast group (*i.e.*, a single class D address).

3.1 Main Data Structures

This subsection depicts the main data structures used by the IMSS routers.

- $RAT[G]$: Redundancy Avoidance Table is maintained for each IP multicast group G with which this IMSS router is involved. $RAT[G]$ has a row for each source (originator) of the IP multicast datagrams that were forwarded to this IMSS router by other IMSS routers. The identifier of the source and the identifiers of these IMSS routers are kept in row $RAT[G][S]$. The information kept in $RAT[G]$ is temporal and is refreshed regularly, as will be explained later.
- eif: expected network interface variable. This variable is concerned with the RPF techniques that are used by the IDMR protocols in order to break routing loops that may occur in *multicast distribution trees.*
- id: identifier of an IMSS router. This is a structure containing the following fields:
  - *physical address*: an ATM address of the IMSS router;
  - operational *role*: *client* or *server*;
  - *mode*: *sender-only* or *regular.*
- $Membership[G]$: group membership table. For each D-group of which an IMSS router is a member, there is a row in this table. Each item in the row is an id structure, as explained above. These memberships are maintained through CONGRESS's incremental membership notifications.

3.2 Maintenance of D-groups

In this subsection we explain in more detailed manner how IMSS routers build and manage D-groups.

3.2.1 Joining D-Groups

In this section we deal with the handling of four kinds of events that cause an IMSS router to join a D-group.

C1: **explicitly requested join:**

1. An IMSS router p finds out (*e.g.*, through processing of IGMP "join_group" request or "MARS_JOIN" request) that there exists some destination within its LIS, that needs to receive IP multicast datagrams that are sent to some IP class D address.
2. An IMSS router p learns via some mechanism (*e.g.*, via some control messages) that there exist downstream multicast routers that depend on it for receiving multicast datagrams for some group.

C2: **traffic-driven join:**

1. An IMSS router p receives an IP multicast datagram via some IDMR propagation tree from some neighboring multicast router.
2. An IMSS router p receives an IP multicast datagram from some directly attached host.

In cases C2.1 and C2.2 an IMSS router should decide (based on local conditions described below) whether to forward a multicast datagram further. Moreover, if it decides to forward, it should also decide which protocol it will use, *i.e.*, via IMSS shortcut connections or via some IDMR multicast distribution tree. The IMSS approach is to use shortcut wherever possible. It should be noted that in order to be useful, a shortcut connection should bypass at least one intermediate router. This is why usually shortcut connections are not opened between the immediate neighbors in the IDMR protocol multicast propagation tree. In order to open the shortcut connections to all other relevant IMSS routers, an IMSS router joins an appropriate D-group.

As was explained in Section 2, an IMSS router may join a D-group assuming either a *server* or a *client* operational role. The operational role of an IMSS router is

indicated by its identifier. Further explanations about the operational roles are provided in Subsection 3.2.3.

If an IMSS router joins a D-group as a *sender-only*, it schedules a timer-related event handler that will terminate the membership of this router in the D-group, if no downstream host emits multicast datagrams for a sufficiently long time. This timer will be referred later, as a *D-timer*.

Note that if downstream routers participate in a "broadcast & prune"-based IDMR protocol, case C1.2 is problematic, since no explicit information about these routers is available. This is a generic problem that does not pertain to shortcut routing only. The same problem arises when any "broadcast & prune"-based routing protocol works in conjunction with a protocol based on "explicit join" messages. As an example consider PIM [11] and DVMRP [9] interoperability issues [12]. Another work in progress that attempts to classify the inter-operability issues that arise from deployment of various IDMR protocols, is given in [17].

In the IMSS approach we solve this problem as follows. Since we allow IMSS to coexist with some other IDMR protocols (see Section 2) on the same NIC, an IMSS router may periodically propagate datagrams using both an IDMR protocol and shortcut connections. This way a multicast propagation tree of an IDMR protocol will be preserved, and all IMSS routers that are also nodes in some IDMR propagation tree (see case C2.1) will join the relevant D-group. As will be explained in the following subsection, an IMSS router leaves this D-group when it receives "prune" messages from all its neighboring downstream multicast routers and no directly attached hosts desire to receive multicast traffic for this class D address.

An IMSS router maintains a special timer that is associated with the periodical propagation through a "broadcast & prune" mechanism. Later we will refer to this timer as *BP-timer*.

3.2.2 Leaving D-Groups

This subsection depicts the part of an IMSS router's algorithm that deals with leaving of the D-groups.

Generally, an IMSS router may leave a D-group corresponding to some class D IP address, when this router has neither directly attached hosts, nor downstream routers that need to receive the IP multicast traffic pertaining to the multicast IP address, or need to send datagrams to it. This happens when

- all directly attached hosts performed IGMP/MARS leave, **and**
- all neighboring multicast routers (of attached networks), running some IDMR protocol, have sent *prune* or *leave* messages (depending on the IDMR protocol) for this group, **or**
- the router is a sender-only member, and its D-timer for this group had expired.

3.2.3 Client and Server Operational Roles

An IMSS router locally decides whether it will assume a *client* or a *server* role upon joining the relevant D-group. The decision depends on a number of connections that are already supported by the IMSS router's NIC and the number of additional connections that need to be supported, if the router decides to assume a specific operational role.

When an IMSS router joins a D-group, assuming the *client* operational role, it expects that some *server* will take care of it. If no server assumes this client for a certain period of time, this client starts using an IDMR protocol for the forwarding of IP multicast traffic. The IMSS routers that act as *servers*, learn through the CONGRESS's incremental membership notifications about the new *client*. Based on the load of the server's NICs and CPU, physical distance, administrative policies *etc.*, each *server* locally decides whether to take care of the new *client*. If a *server* decides to serve a *client*, it tries to open an ATM VC to this client (or to add this client as a leaf to an already opened ptmpt connection). If the *client* has already accepted some other *server*'s connection set-up request, it may either refuse to accept the new connection, or tear down the previous connection and to switch to the new one. In both cases this is a local decision of the *client*.

In case of some *server*'s failure, all its *clients* should re-join the relevant D-group. This will once again trigger the procedure described above.

It should be noted that the operational roles are not fixed "once and for all". Depending on the size of a D-group and the local NIC and CPU load, an IMSS router may desire to change its operational role. In order to do this, an IMSS router should simply leave its D-group and then re-join it with the appropriately updated identifier that indicates its new operational role (see Section 3.1).

3.2.4 Regular and Sender-Only Modes

An IMSS router may operate either in *regular* or *sender-only* mode, as was explained in Section 2. An IMSS router may wish to change its mode from *sender-only* to *regular* if it learns about some downstream host or router that needs to receive the multicast traffic pertaining to a specific class D address. In order to perform this transition, an IMSS router should leave the

relevant D-group and re-join it with the updated identifier indicating that it is acting in the *regular* mode.

Note, that actually there is no need for the transition in the opposite direction, *i.e.*, from a *regular* to a *sender-only* mode. Indeed, if an IMSS router does not have any downstream hosts or routers that desire to receive multicast traffic, this IMSS router will simply leave the relevant D-group (see Subsection 3.2.2). If there exist some down-stream senders, this IMSS router will re-join the group on-demand later, as was explained in Subsection 3.2.1.

3.3 Forwarding Decisions

This subsection depicts the forwarding algorithm executed by the IMSS routers. Due to the assumed heterogeneous network model, there are multiple cases that should be handled carefully. By using CONGRESS membership services and the multiplexing/demultiplexing technique described in Section 2, an IMSS router can differentiate between the multicast traffic that it receives from another IMSS routers via the shortcut connections, from a non-ATM IDMR interface or from another IMSS router that used an IDMR propagation tree. An IMSS server decides how to forward an incoming multicast packet according to the identity and operational role of the sending router and according to its own operational role. For each possible pair of sender and receiver, the table in Figure 4 provides a pointer to the subsection that describes the relevant part of the protocol. The short parts of the protocol are shown directly in the table.

For the sake of simplicity and shorter representation, we assume that the involved IMSS routers have already joined the relevant D-groups, according to the algorithm explained in Subsection 3.2.1.

In all of the following cases we depict the steps taken by an IMSS router, upon a reception of an IP multicast datagram m originated at some source S and targeted to some multicast group G.

3.3.1 A Server Receives a Datagram from a Client

An IMSS router acting as a server, is responsible for the propagation of the multicast traffic that it receives from its *clients*, to all the relevant multicast routers and directly attached hosts.

In order to avoid undesired duplication of IP multicast datagrams, an IMSS router should check whether some other IMSS router(s) might propagate the IP multicast datagrams originating at the same source. This may happen when a multicast distribution tree of some IDMR protocol contains more than one *egress* router that connect the branches of the propagation tree to the ATM cloud. Figure 1.a provides a graphical representation of this scenario. In such a case, it is obviously preferable that only one of the egress routers will transmit the datagrams.

In cases such as described above, IMSS routers belonging to the same D-group, can deterministically choose which router will perform forwarding of IP multicast datagrams by using the CONGRESS membership services. This is done by consulting the RAT[G] table. Each IMSSrouter locates the entry with the maximal recorded TTL in the list RAT[G][S] (it belongs to the router that is the closest one to the datagram source in terms of number of hops[3]) and identifies it to be the *designated injector* for this (G, S) pair. The router compares the value of the maximal TTL with the TTL of the last packet originated by the same source and received from the IDMR interface. If its own recorded TTL is higher, or if RAT[G][S] is empty, the router forwards datagrams to all relevant directions. Otherwise, datagrams received from the IDMR interface are discarded (they will be received through a short-cut VC from one of the other IMSS routers) . Since we assume an asynchronous network model, it is possible that at some point multiple IMSS routers belonging to the same D-group, will consider themselves as the ones that must forward datagrams. As time passes, however, the IMSS routers will learn about this redundancy, because it will be reflected by RAT[G]. In the following subsection more details about RAT maintenance are provided.

The information kept in $RAT[G]$ is temporal. Each time an IMSS router enters information into a row S of $RAT[G]$, it resets a timer associated with the source S. We refer to this timer as *S-timer*. If no traffic from S is encountered during the time window defined by the S-timer, the IMSS router discards the row in $RAT[G]$ associated with S.

When $RAT[G]$ becomes empty, the IMSS router starts another timer, called *G-timer*. In case no multicast traffic is encountered within G during the G-timer, an IMSS router tears down the shortcut connections within the corresponding D-group. These shortcut connections can be resumed on-demand later.

3.3.2 A Server Receives a Datagram from another Server

If a *server* receives multicast traffic from another *server* belonging to the same D-group, the sending *server* believes that it is the designated injector for the relevant

[3]Note that the TTL field might be modified by various Internet Service Providers. In this case a mechanism more specific to IMSS can be deployed.

<table>
<tr><th>Sender / Receiver</th><th>IDMR-only Router or a directly attached host</th><th>IMSS Entity via IDMR</th><th>IMSS Client</th><th>IMSS Server</th></tr>
<tr><td>IMSS Client</td><td>3.3.3</td><td rowspan="2">1. forward m using only IDMR propagation tree.
2. Reset BP-timer.</td><td>X</td><td>Forward m using IDMR protocol to all non IMSS interfaces;</td></tr>
<tr><td>IMSS Server</td><td>3.3.4</td><td>3.3.1</td><td>3.3.2</td></tr>
</table>

Figure 4: Forwarding Decisions Table

(G, S) pair. Otherwise it would not have been sending the datagrams. The receiving *server* should enter the identifier of the sending *server* into the corresponding row of RAT. Note that this operation may change the local notion of the IMSS router with the lexicographically minimal identifier, at the receiving IMSS router.

An IMSS router acting as a server, is responsible for the propagation of the IP multicast traffic to all its clients belonging to the same D-group and to all the relevant IDMR interfaces. The latter case should be treated especially carefully because IDMR routers use RPF mechanisms in order to break stable routing loops. When a multicast IP datagram arrives to an IDMR router, the router checks whether it received it from the "expected" network interface. An IDMR router expects to receive multicast datagrams originated at some source S, from the **same** network interface that this router would use in order to forward **unicast** datagrams to S. If a multicast datagram arrived from an unexpected interface, it is silently discarded, because it was not propagated over the optimal branch of the IDMR multicast propagation tree.

An IMSS router updates the variable eif to be as expected by the IDMR interface. Otherwise, the RPF mechanism might discard the datagrams that should not be discarded.

Obviously, there is no need to forward an IP multicast datagram that came from an IMSS router acting as a *server* to other *servers* belonging to the same D-group. These *servers* are supposed to be the **leaves** of the same *ptmpt* connection as the receiving *server*.

3.3.3 A Client Receives a Datagram from an IDMR Interface

When an IMSS server acting as a *client* receives an IP multicast datagram from an IDMR interface, it should forward it to all other involved IDMR interfaces excluding those that also participate in the IMSS protocol. In order to propagate the datagram to all the relevant IMSS routers, a *client* should forward the datagram to its *server*. The latter will forward it further according to the algorithm described in Subsection 3.3.1.

As was explained in Subsection 3.2.1, IMSS routers that also participate in some "broadcast & prune"-based IDMR protocol, periodically forward multicast traffic over the IDMR multicast propagation tree in addition to the propagation over the shortcut connections.

3.3.4 A Server Receives a Datagram from an IDMR Interface

If an IMSS router acting as a *server* receives an IP multicast datagram from some non-IMSS router via an IDMR multicast propagation tree, it is responsible to forward it to all the relevant non-IMSS multicast routers and to the relevant *clients*. In case this IMSS router also has the lexicographically minimal identifier in the D-group (according to its local RAT), it should also forward the multicast datagram to all other IMSS routers acting as *servers* and belonging to the same D-group.

4 Conclusion

We have presented a conceptual solution to the problem of shortcut routing of the IP multicast traffic over large ATM cloud. IMSS is an "improved best effort" IP multicast over ATM. IMSS improves the performance of the traditional IP multicast routing protocols wherever possible, by using the shortcut paradigm. It is the first attempt to do so in a large ATM WAN. IMSS is inter-operable with the conventional IDMR protocols and preserves the traditional IP multicast interface for the end-users. Multicast group address resolution and IP multicast routing decisions are almost orthogonal problem spaces. Both of these problems are greatly magnified in a large ATM cloud environment. In order to provide the best solutions, each of the problems

should be tackled separately with the most appropriate methodology. IMSS follows this approach. It is built on top of a native ATM multicast group resolution and maintenance service. There is no such service that scales to an ATM WAN environment in the current ATM standards. Therefore, we rely on CONGRESS as a conceptual solution for the scalable multicast group address resolution and maintenance service. The prototypes of CONGRESS and IMSS are currently under implementation.

References

[1] T. Anker, D. Breitgand, D. Dolev, and Z. Levy. Congress: CONnection-oriented Group-address RESolution Service. In *Proceedings of SPIE'97 vol. 3233 on Broadband Networking Technologies*, November 1997. To appear. Available from: `http://www.cs.huji.ac.il/ transis/`.

[2] G. Armitage. *A Distributed MARS Protocol.* Internet Engineering Task Force, Internetworking over NBMA (ION) Working Group, January 22nd, 1997. Internet Draft, expires July 22nd, 1997.

[3] G. Armitage. *VENUS - Very Extensive Non-Unicast Service.* Internet Engineering Task Force, Internetworking over NBMA (ION) Working Group, February 18th, 1997. Internet Draft expires August 18th, 1997.

[4] G. Armitage. *Support for Multicast over UNI 3.0/3.1 based ATM Networks, RFC 2022.* Bellcore, November 1996.

[5] G. Armitage. *Issues affecting MARS Cluster Size, RFC 2121*, March 1997. Internet Engineering Task Force, Network Working Group.

[6] ATM Forum. *ATM User Network Interface (UNI) Specification Version 3.1.* Prentice Hall, Englewood Cliffs, NJ, June 1995. ISBN 0-13-393828-X.

[7] The ATM Forum Technical Committee. *ATM User-Network Interface (UNI) Signalling Specification Version 4.0, af-sig-0061.000*, July 1996.

[8] A. J. Ballardie, P. F. Francis, and J. Crowcroft. Core based trees. In *Proceedings of the ACM SIGCOMM, San Francisco*, 1993.

[9] D. Waitzman, C. Partridge, and S. Deering. *Distance Vector Multicast Routing Protocol, RFC 1075.* IETF, November 1988.

[10] S. Deering. *Host Extensions for IP Multicasting, RFC 1112.* Stanford University, August 1989.

[11] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. The pim architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4 (2):153–162, April 1996.

[12] D. Estrin, A. Helmy, and D. Thaler. *PIM Multicast Border Router (PMBR) specification for connecting PIM-SM domains to a DVMRP Backbone.* Internet Engineering Task Force, Network Working Group, February 1997. Internet Draft, expires August 1997.

[13] James V. Luciani Dave Katz David Piscitello Bruce Cole C. Topolcic. *NBMA Next Hop Resolution Protocol (NHRP).* Internet Engineering Task Force, Routing over Large Clouds Working Group, March 1997. Internet Draft expires September 1997.

[14] M. Laubach. *Classical IP and ARP over ATM, RFC 1577.* Hewlett-Packard Laboratories, December 1993.

[15] J. Moy. *Multicast Extensions to OSPF, RFC 1584*, March 1994. Internet Engineering Task Force, Network Working Group.

[16] C. Semeria. *Introduction to IP Multicast Routing.* Internet Engineering Task Force, January 1997. Internet Draft, expires July 1997.

[17] D. Thaler. *Interoperability Rules for Multicast Routing Protocols.* Internet Engineering Task Force, Inter-Domain Multicast Routing (IDMR) Working Group , November 1996. Internet Draft, expires May 1997.

[18] Y. Rekhter D. Kandlur. *"Local/Remote" Forwarding Decision in Switched Data Link Subnetworks, RFC 1937*, May 1996. Internet Engineering Task Force, Network Working Group.