The Domainserver Hierarchy for Multicast Routing in ATM Networks

Sridhar Komandur*
komandur@cs.pitt.edu
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260, USA

Matthew B. Doar
doar@pobox.com
Vitria Technology, Inc.
500 Ellis Street
Mountain View, CA 94043, USA

Daniel Mossé
mosse@cs.pitt.edu
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260, USA

1 Introduction & Motivation

Multicast service is an important aspect of any routing architecture. An important role of multicast service is to provide *group abstraction*[1]. ISPs and major carriers are building the necessary infrastructure to support multicast and service differentiation capabilities. The popularity of applications that need multicast service is evident in the success of the Internet Multicast Backbone (MBONE)[2].

There are two main approaches to multicasting. One is Source Specific Trees (SST) approach in which each source has a multicast tree with itself as the root and the receivers as the leaves. The other approach is Shared Trees (ST) in which all the members share a tree for data forwarding. The performance of a multicast protocol can be measured in terms of end-to-end latency from a sender to destination, bandwidth consumption, traffic concentration on the links, network resources to maintain the state, join latency, membership management complexity and operating system overhead (due to timers and routing table updates). The choice of multicast tree affects almost all of the above metrics. Recent studies suggest that shared trees (STs) are very scalable in terms of bandwidth consumption, network resources, join latency, membership management complexity and operating system overhead[3, 4, 5]. The importance of STs and increasing deployment of ATM in WANs makes it imperative that we support multipoint-to-multipoint (*mp-mp*)[5, 6, 7] connections efficiently and scalably in ATM networks.

The Public Network-to-Network Interface (PNNI)[8] is a hierarchical routing protocol designed to scale efficiently to large networks. While PNNI specifies point-to-point routing, multicast routing has been left unspecified. The work in [9] provides signaling support for the point-to-multipoint (*p-mp*) connection but mp-mp routing in ATM networks has not yet been studied in depth. This paper tackles this problem. In the rest of this section we will discuss applicability of some of the multicast routing solutions proposed for the Internet and provide motivation for our approach.

The message overhead in DVMRP[10] and the state and computation overhead in MOSPF[11] inhibit the scalability of these protocols to large networks. PIM-Sparse Mode[12] tries to ameliorate the problems of DVMRP, however, the protocol has large state overhead and some complex features of the protocol (e.g., changeover from dense to sparse mode) are yet to be specified or evaluated[4]. HDVMRP[13] is an encapsulation mechanism for quick implementation of DVMRP as level 2 protocol among the regions. It is used to lessen the router state required for the current Internet,

*Author responsible for correspondence. Phone: 412 624 8463 Fax: 412 624 8854

but it inherently suffers from the same scaling limitations as DVMRP. There was an attempt to extend Doar's[14] p-mp work on naive multicast in Hierarchical PIM (HPIM)[15]. HPIM primarily addresses the problems associated with the Rendezvous-point (RP) of PIM, but its performance is yet to be evaluated.

The above protocols take a soft state approach, in which the unicast routing determines the current multicast routing. While soft state approaches are robust during failures, they could potentially cause service disruption to QoS sensitive applications[16]. CBT[17] is an ST approach based on hard state routing in which the route does not change as long as there are no failures. OCBT[18] suggests modifications to make CBT loop-free. OCBT and another approach called CGBT[19] introduce a hierarchy of cores into CBT. The work in [20] maps the CBT approach to the PNNI hierarchy. The key limitation of these approaches is the core placement and management complexity. When several multicast groups are active then unless the cores are replicated, they could become bottlenecks. Also, hard state approaches do not take advantage of better routes that may become available after the connection has been established.

Many of the multicast proposals we mentioned above were designed for the Internet environment, in which the interaction between the different routing protocols at the higher level (e.g., BGP4) and a lower level (e.g., RIP or OSPF) does not permit good routing decisions across a hierarchy. PNNI routing protocol of ATM has been designed from the ground up with scalability as a primary consideration. While it is not clear that ATM will see widespread deployment in the campus networks, if the past couple of years are any indication, ATM has found acceptance in the wide area backbone networks. As experience has shown, multicast is a difficult service to support in the connection oriented ATM networks. MARS[21] attempts to emulate IP multicast paradigm. However, the server based MCS approach and per-source tree based VC mesh approach exhibit scaling limitations[5, 6].

Our proposed architecture *Domainserver Hierarchy* differs from the above in some important ways. We do away with the need to have preconfigured *cores* (simplifying the protocol operation), do not load any one switch for all the multicast connections (better load balancing), leverage the existing mechanisms in PNNI protocol (thereby using already debugged and well-tested mechanisms), have the potential to support dynamic rerouting for quality of service and localized fault handling, and provide opportunities to extend the features of the domainserver as needed (e.g., domain servers could maintain policy and security information). In addition, our approach can potentially be integrated with the hierarchical unicast routing protocols such as Landmark Hierarchy[22] and Viewserver Hierarchy[23]. Due to space constraints we describe only key aspects of our architecture.

2 Domainserver Hierarchy

The PNNI[8] specification defines the protocol for distributing topology information between switches and clusters of switches. The PNNI hierarchy defines a *peer group* as a collection of physical switches at the lowest level, each of which exchanges information with other members of the group, such that all members maintain an identical view of the group. A peer group is represented in the next level of the hierarchy by a single switch called the *peer group leader* (PGL)[1]. The PGL is responsible for the aggregation and distribution of information for maintaining the PNNI routing hierarchy. Figure 1 shows an arbitrary network with three peer group levels, where PGLs are shown as dark nodes.

There are two modes of mp-mp communication, namely a *control mode*, for signalling (host join and/or leave, establishment of the tree), and a *data forwarding mode*, for data communication

[1] A PGL is elected through a continuously running peer group leader election process.

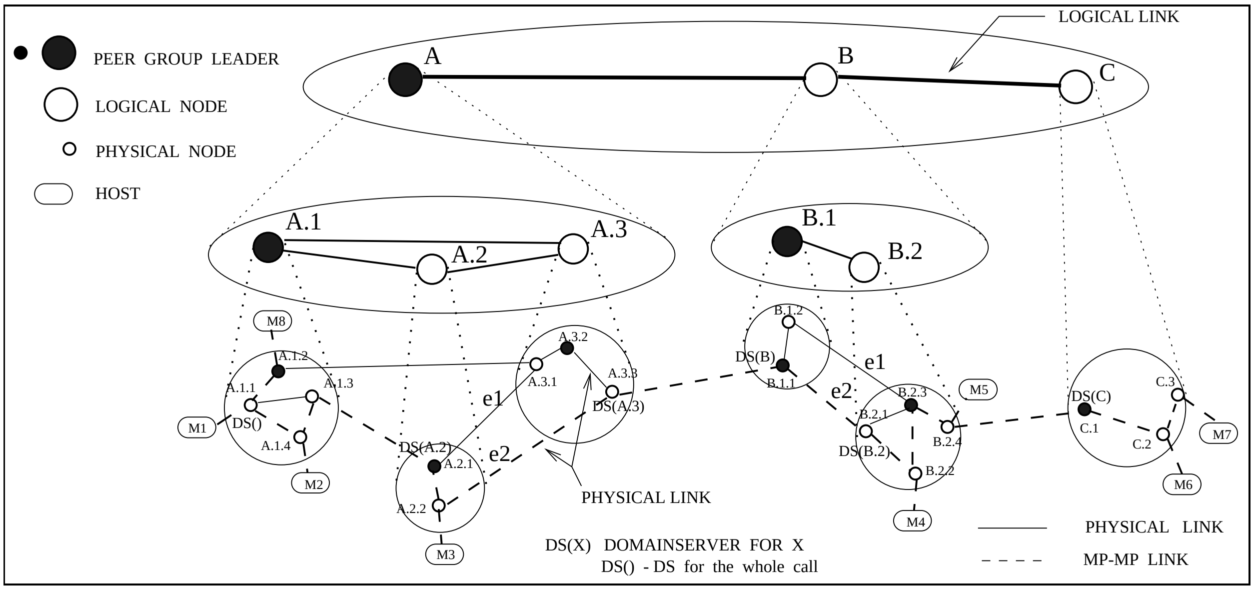


Figure 1: An MP-MP Call in an Arbitrary PNNI Routing Hierarchy

among group members. For control, we impose a directed logical tree over the multicast connection (also called an *mp-mp call*). In other words, for control purposes every switch in the connection has an incoming link and one or more outgoing links. During data forwarding, there is only the concept of an undirected shared tree: a packet arriving in one edge is forwarded to all other edges of the tree. We also assume an mp-mp connection is identified by a *group address*, which can be obtained by a host that wants to join the group.

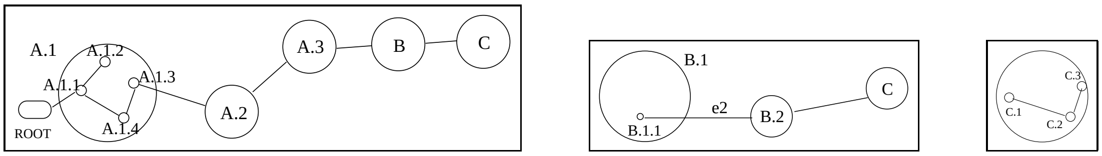
2.1 Key Concepts

There are three key concepts in our approach, namely *domains*, *views*, and *domainservers* (DS). The first two are defined for each switch in the multicast group.

Every switch in a mp-mp connection has a *domain*. The domain at a switch is defined as the highest level peer group entered from the incoming link (for the connection); the domain is the entire network if the incoming link is a UNI. For example, in Figure 1, the domain of $A.2.2$ consists of itself. The call enters from $A.2.1$, thus the call enters the domain consisting of just $A.2.2$. The domain of $B.1.1$ is the peer group B. The call enters $B.1.1$ from $A.3.3$, thus the call is entering the highest level peer group B at $B.1.1$. The domain of switch $A.1.1$ is the whole network, since the incoming link is a UNI.

In addition to the domain, every switch on the mp-mp call has a *view*. A view at a switch is defined as the union of links in the source routes (i.e., Designated Transit Lists - DTLs in the PNNI terminology) constructed by the switch (for the call). A source route is constructed based on the information available in the PNNI database; since we assume dynamic joins, the view at a switch will contain only information about the mp-mp call not available at the switches up in the control tree towards the group initiator (see below).

Every switch has detailed information about its peer group and its ancestor peer groups in the hierarchy. So as we go up in the hierarchy the routing information aggregation increases. For example, switch $A.1.1$ has detailed information of peer group $A.1$ and the ancestor peer group A. It may not be aware of the details within $A.2$ or $A.3$ or even the existence of second level peer groups within B, depending on how the PGLs advertise aggregated topology at the higher levels.



(a) $A.1.1$'s View (b) $B.1.1$'s View (c) $C.1$'s View

Figure 2: View of a Node

The source route from $A.1.1$ to, say, member $M4$ will take the join request to B, and then to the border switch $B.1.1$ (creating the source route within $B.1$) and to the border switch $B.2.1$ (creating the source route within $B.2$). Figures 2(a), 2(b) and 2(c) illustrate the views of the switches $A.1.1$, $B.1.1$ and $C.1$ respectively. Similarly, the view of the switch $B.2.2$ is just the link $B.2.2$ - $B.2.3$.

The application host that initiates the group or the entry border switch in a peer group belonging to the mp-mp call acts as *domainserver* (DS). The role of domainserver will become clear in the discussion below.

Group Creation: The multicast group initiator (typically an application at a host) registers the group address, along with itself as the DS, with the local PGL. This information is passed up in the hierarchy, to the extent permitted by the distribution scope of the group. The scope of the call shown in the Figure 1 spans the whole hierarchy, so it reaches the PGL of the third level (A). In this example, switch $A.1.1$ is announced as the DS in the peer group A (the PGL representing group A is the physical switch $A.1.2$).

Note that unlike the PNNI routing information, the group address information is not passed down the hierarchy unless needed. The PGL was chosen for information dissemination as it is known to every switch in the peer group and is maintained by the PNNI routing. This facilitates efficient/speedy implementations.

Group Join: Assuming a multicast group address is known, when a host intends to join the mp-mp call, there are two options available. It can directly send the Leaf-Initiated Join (LIJ [9]) request to the PGL, which in turn forwards it to the DS, or obtain the DS address from the PGL and send the join request directly to the DS. The DS uses its view to establish a path to the new member based on QoS or policy constraints.

To explain the join process, let us consider an example. If $M4$ intends to join the group and is the first member outside the peer group A, the join request is forwarded to the PGL $B.2.3$ which in turn forwards it to the PGL of B, $B.1.1$. Since the call is not known at the second level in B, the join request is forwarded to the PGL of the third level, A (i.e., the physical switch $A.1.2$), which is aware of the multicast group. Thus the join request is eventually forwarded to the DS $M1$. The DS sends an join acknowledgement to the switch $A.1.1$ with $M4$ address as the destination. $A.1.1$ creates the appropriate DTLs. When the join acknowledgement is received at $B.1.1$, it announces itself as the DS for the domain B. Similarly upon receiving the join acknowledgement, $B.2.1$ announces itself as the DS for $B.2$. Later, when $M5$ wants to join the multicast group, the PGL $B.2.3$ forwards the join request directly to the switch $B.2.1$. In this case, the join request travels only as far as the nearest DS for the multicast group.

Fault Handling: Among the advantages of the hierarchical domainserver architecture is fault handling. Since each DS is responsible for the maintenance of the call within its domain, faults can be handled within a limited scope without the need for a complex network-wide recovery algorithm. For example, if the link $B.2.1$ - $B.2.2$ goes down, the DS $B.2.1$ tries to establish an alternate route to $B.2.4$, based on the view maintained at the DS. If the switch $A.3.3$ goes down, the failure information is passed up the control tree to the next level (A) DS, which happens to be switch $A.1.1$ in our case. It is easy to see that fault propagation is as limited as possible and the DS tries to recover as quickly as possible.

3 More on the Joining Process

One of the important issues in multicasting is *tree establishment*, which addresses the issue of how and where to join a new user. LIJ has been widely recognized as a requirement for a scalable multicast architecture[17, 12]. Clearly, when the initial multicast group membership is known in advance (e.g., an audio/video conference), the network multicast service should allow the initiator to join the group members as in the current UNI 3.1 p-mp connection. This service may be necessary for either compatibility, policy, or efficiency reasons.

One issue facing all the protocols that support LIJs is how to initiate the join process. As discussed above, we deal with this as two different issues: where to send the join request and where to join the new member. We assume that the multicast group address can be obtained (perhaps through a hierarchical version of Session Directory[24]) and that the DS address is obtained through interaction with PGL. When a join request is received at an on-tree switch, then the switch simply forwards the request towards the DS. When a join request is received at a DS, the DS checks if the joining switch is in its domain and if it is not, the request is forwarded towards the next level DS. Thus, the join request is eventually processed at the nearest on-tree DS whose domain encompasses the joining switch (Our scheme can degenerate to CBT-like join if the first on-tree switch acknowledges the join request).

While adding the new member, the DS makes necessary changes to its view based on the policy or group requirements. Thus, our architecture clearly draws a line between policy and mechanism.

4 Summary

In this paper we introduced the *Domainserver Hierarchy* for scalable multipoint-to-multipoint multicast over ATM networks using PNNI. Although much effort has been dedicated to tree establishment in this and other works, tree maintenance has not received much attention in the research community. Tree maintenance involves increasing/maintaining network performance while managing the tree in tune with the multicast group requirements. To achieve these objectives, portions of the tree may need to be reconfigured. Tree reconfiguration involves addressing the following issues: who is in-charge of the process, how often and what are the criteria (i.e., how do we choose the metrics for evaluation), and, finally, the scope of reconfiguration to ensure that the benefits outweigh the potential overheads. We are currently addressing these issues.

Acknowledgements: We would like to thank Ted Tedijanto, FORE Systems and Jon Crowcroft, UCL for insightful discussions and comments.

References

[1] K. P. Birman, R. V. Renesse, and S. Maffeis. Horus: A Flexible Group Communication System. In *Communications of the ACM*, volume 39, pages 76–83, April 1996.

[2] S. Casner. Are you on the MBone? In *IEEE Multimedia*, volume 1, Summer 1994.

[3] L. Wei. *Scalable Multicast Routing: Tree Types and Protocol Dynamics.* PhD thesis, Department of Computer Science, University of Southern California, December 1995.

[4] T. Billhartz, J. B. Cain, E. F. Goudreau, D. Fieg, and S. Batsell. Performance and Resource Cost Comparisions for the CBT and PIM Multicast Routing Protocols in DIS Environments. In *IEEE INFOCOM*, 1996.

[5] M. Grossglauser and K. K. Ramakrishnan. SEAM: Scalable and Efficient ATM Multicast. In *IEEE INFOCOM*, 1997.

[6] S. Komandur and D. Mossé. SPAM: A Data Forwarding Model for Multipoint-to-Multipoint Connection Support in ATM Networks. In *6th International Conference on Computer Communications and Networks.* IEEE Computer Society, September 1997.

[7] S. Komandur, J. Crowcroft, and D. Mossé. CRAM: Cell Re-labeling at Merge-points for ATM Multicast. *To appear in IEEE International Conference on ATM (ICATM)*, June 1998.

[8] PNNI. Private Network-Network Specification Interface. Technical report, The ATM Forum Technical Committee, March 1996.

[9] T. Tedijanto, J. Drake, R. Rennison, S. Komandur, R. Cherukuri, W. L. Edwards, B. Miller, J. H. Halpern, R. Callon, E. M. Spiegel, Y. Shavit, and R. Bhat. Support for Leaf Initiated Join (LIJ) in PNNI. In *ATM Forum Contribution*, Chicago, IL, April 1997.

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

[11] J. Moy. Multicast Extensions to OSPF. *RFC1584*, 1994.

[12] S. Deering, D. Estrin, D. Farinacci, and V. Jacobson. An Architecture for Wide-Area Multicast Routing. In *ACM SIGCOMM*, September 1994.

[13] A. S. Thyagarajan and S. E. Deering. Hierarchical Distance-Vector Multicast Routing for the MBone. In *SIGCOMM.* ACM, August 1995.

[14] M. Doar and I. Leslie. How Bad is Naive Multicast Routing? In *INFOCOM*, 1993.

[15] M. Handley, J. Crowcroft, and I. Wakeman. Hierarchical Protocol Independent Multicast (HPIM). Unpublished draft, November 1995.

[16] D. Zappala, B. Braden, D. Estrin, and S. Shenkar. Interdomain Multicast Routing Support for Integrated Services Networks. Technical report, University of Southern California, November 1996.

[17] A. Ballardie, J. Crowcroft, and P. Francis. Core Based Trees (CBT) - an Architecture for Scalable Inter-Domain Multicast Routing. In *ACM SIGCOMM*, pages 85–95, September 1993.

[18] C. Shields and J. J. Garcia-Luna-Aceves. The Ordered Core Based Tree Protocol. In *IEEE INFOCOM*, April 1997.

[19] Y. C. Chang, Z. Y. Shae, and H. W. LeMair. CGBT: Multiparty Videoconferencing using IP Multicast. *Multimedia Computing and Networking*, 1996.

[20] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. P. Kumar. Hierarchical Multicast Routing in Wide-Area ATM. In *IEEE International Conference on Communications (ICC)*, June 1996.

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

[22] P. F. Tsuchiya. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. In *ACM SIGCOMM*, August 1988.

[23] C. Alaettinoglu and A. U. Shankar. The Viewserver hierarchy for Interdomain Routing: Protocols and Evaluation. *IEEE Journal on Selected Areas in Communication*, pages 1396–1410, October 1995.

[24] M. Handley and V. Jacobson. SDP: Session Description Protocol. *Internet Draft draft-ietf-mmusic-sdp-02, Work in progress*, November 1996.