IEEE
COMMUNICATIONS
SURVEYS &
TUTORIALS
The Electronic Magazine of
Original Peer-Reviewed Survey Articles

Intra- and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy

Maria Ramalho, Alcatel Corporate Research Centre

Abstract

A multicast routing protocol should be scalable, robust, use minimal network overhead, consume minimal memory resources, inter-operate with other multicast routing protocols, and be easy to implement. The mechanism used to manage the multicast group (participants join and leave throughout the multicast session) is an important issue when designing a protocol for multicast routing. Other design choices are influenced by the distribution of the participants over the routing domain (sparse or dense), the role of the participants in the group (source, receiver, or both), the number of groups and participants per group, and the requirements of the participants in terms of transmission delay. In this study a taxonomy of IP multicast routing protocols for dynamic groups will be presented. This taxonomy will be used to classify a surveyed set of intra- and inter-domain multicast routing protocols and to discuss successful protocol design regarding satisfaction of the multicast application's latency requirements as well as the network's resource consumption requirements.

The purpose of IP multicast routing is to provide efficient communication services for applications that send the same data to multiple recipients, without incurring network overloads. Hence, at each router, only one copy of an incoming multicast packet is sent per link, rather than sending one copy of the packet per number of receivers accessed via that link [1]. Some of the applications for which multicasting is advised are: video-conferencing, shared workspace, distributed interactive simulation (DIS), software upgrading, and resource location.

In Deering's model [1], IP Multicast is associated with the notion of a *group*, identified by a certain *address* (the IP class D address) and composed of a certain number of *participants* (senders or receivers).

In IP networks, the notion of *member* of a multicast group is associated with a certain registration mechanism [2]. The registration is required only for receivers. Thus, a source does not need to register as a member to *send* packets to a multicast group. Members can join and leave during the life of a multicast session.

The mechanisms required to support IP multicast consist mainly of:

- Class D address allocation.
- Group membership management.
- Routing of multicast data.

The dynamic aspects considered in IP multicast routing are not just topology changes but also changes in the set of members of the group. In order to understand the idea of a dynamic multicast routing environment, let us decompose it in changes in:

- The set of sources: an active source stops transmitting or a new source becomes active.
- The set of receivers: a new member joins the group or an existing member unsubscribes.
- The routing topology: a new node is added, an existing node becomes unaccessible.
- The cost values associated with network links.

The first two items are specific to multicast routing. The third is not; unicast routing protocols also take into account dynamics in the routing topology. Moreover, the first item only needs to be taken into account for multiparty to multiparty communication, and the fourth when constrained (QoS) routing is applied. In this study, we will focus on multicast-specific routing aspects. Thus, wherever mention to unicast routing protocols is made, it is assumed that the unicast routing protocol is loop-free, robust in terms of changes in the routing topology, efficient in terms of the control message overhead, and scalable to networks with large numbers of nodes.

In the following, an introduction to different mechanisms for group management is presented. The taxonomy proposed in this article to characterize multicast routing protocols is presented, followed by a description of the main characteris-

Currently with Starlab N.V., Brussels-Belgium

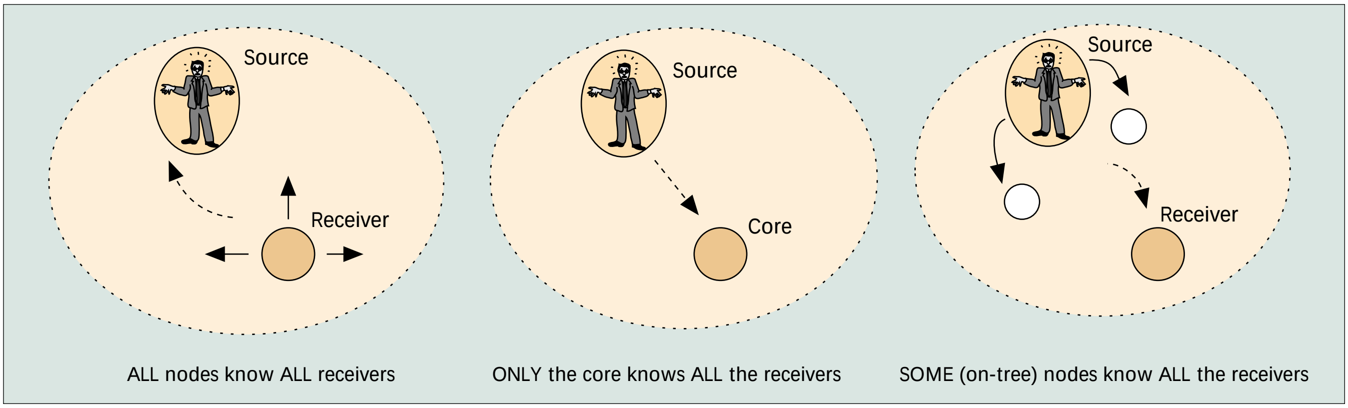

Source
Receiver
ALL nodes know ALL receivers
Source
Core
ONLY the core knows ALL the receivers
Source
Receiver
SOME (on-tree) nodes know ALL the receivers

FIGURE 1. *Who are the receivers?*

tics of the intra- and inter-domain multicast routing protocols illustrated in this article. This survey is by no means exhaustive, since the research in this field is in constant evolution. However, the most important protocols that resulted from research papers and standardization fora will be identified. Finally, comments are made on the implications of specific protocol design choices on scalability, robustness, efficiency, and other performance requirements specific to a certain type of multicast application. It is also recommended to read a previous and more general survey of multi-point communications in IP and ATM by Diot *et al.* [3].

GROUP MANAGEMENT

In IP multicast, the mechanism used to detect changes in the set of participants at session start-up will *also* be applied during the life of the multicast session. Thus, a flexible and robust group management scheme is required. IP multicast packets are sent to a class D destination address that is independent of the physical destination address, that is, it identifies not one but the whole set of receivers of a multicast group. From a data forwarding perspective, sources of a certain multicast group do not need to know[1] the recipients to whom the packets are being sent: receivers are anonymous! Thus, it is the task of the multicast routing protocol to actually locate those receivers and set up a multicast tree that links the source to each receiver. There are three main schemes to locate and detect changes in the set of receivers (Fig. 1):

- **Flooding:**[2] "All nodes know all receivers." The receiver advertises its address to all the nodes in the domain.
- **Centralized:** "Only the core knows all the receivers." A node is configured as a rendezvous point (RP) or core node for a multicast group. This node acts as the meeting point for the sources and the receivers. The receiver advertises its address only to this node.
- **Distributed:** "Only nodes that are part of the multicast tree know all the receivers." The receiver advertises its address only to the nodes of the tree. It discovers these nodes via successive probe messages between itself and its neighbors.

IP multicast allows not only the receivers but also the sources of a multicast group to change over time. Thus, a mechanism is also required to trace changes in the set of sources. The latter is a tricky issue in IP multicast. At this point, a small note is necessary to differentiate between different types of multicast or multipoint-to-multipoint communication. Both of these terms are ambiguous since they do not reflect whether the endpoints of the communication act as (a) either sources or receivers or (b) both as sources and receivers (in this case we can speak of group "participants" since the terms are used interchangeably). In IP multicast, receivers are required to register as members of a certain multicast group. Thus, it is possible to "map" a multicast address to a set of receivers. The same is not valid for sources since, unless they

[1] *For security purposes, however, it is highly desirable that the source be informed of the identity of the receivers and vice-versa.*

[2] *Flooding consists in forwarding a message on all outgoing interfaces (except the one from where it arrived from). The flooding process generates a vast amount of duplicated packets and stops only if some mechanism is used (e.g., decrementing the Time-To-Live field at each traversed hop).*

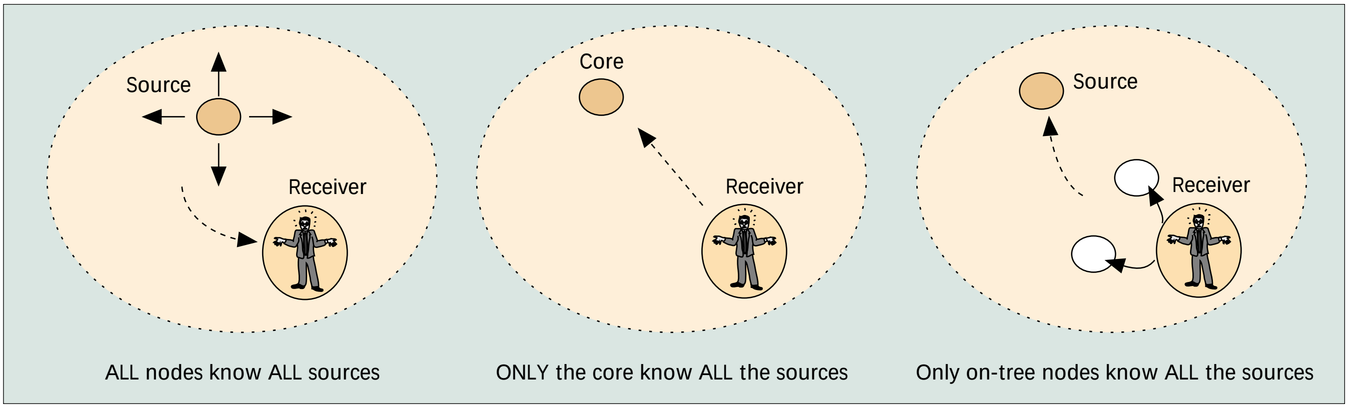

Source
Receiver
ALL nodes know ALL sources
Core
Receiver
ONLY the core know ALL the sources
Source
Receiver
Only on-tree nodes know ALL the sources

FIGURE 2. *Who are the sources?*

want to receive data from the group, they are not required to register. Thus, how can changes in the set of sources be detected? Similar schemes to the ones applied to detect changes in the set of receivers can be used for the set of sources (Fig. 2).

For communications of type (a) and if the set of sources and the set of receivers do not change very often over time, flooding constitutes a simple and robust means of advertising either changes in the set of sources or changes in the set of receivers. For example, DVMRP uses flooding to advertise a new source and MOSPF uses flooding to advertise changes in the set of receivers. However, flooding does not scale to large routing domains, large numbers of multicast groups, or large numbers of participants per group. This is because of the high control message overhead. A hierarchical network topology is an alternative to limit flooding to small routing domains resulting from the partition of the whole routing domain into a set of hierarchically distributed routing domains. (For example, MOSPF uses a two-level hierarchy whereby a routing domain is divided into areas connected via a backbone area (Fig. 7)).

If both sources and receivers join and leave the group quite often during the life of a multicast session of type (a), then a centralized solution is recommended to detect changes in the set of sources and the set of receivers. Both the centralized and distributed solutions assume that a shared multicast tree is used: packets flow from the sources to the RP and from the RP to the set of receivers (*unidirectional* forwarding (Fig. 3)). In unidirectional forwarding, one can also distinguish between the upstream and downstream tree nodes relative to a certain node in the tree. Thus, *upstream* nodes are those on the branches between the current node and the sources, and *downstream* nodes are those on the branches between the current node and the receivers.

A centralized scheme is also advisable for the management of a group of type (b). However, the non-differentiation between the role of source and receiver will result in *bi-directional* rather than unidirectional forwarding (Fig. 4). In bi-directional forwarding, data injected in one of the branches of the tree does not necessarily have to go via the RP to reach an end-node of the tree. This property makes bi-directional forwarding quite attractive when the end-node can be both a source and a receiver. A distributed scheme is advised to avoid the use of a core. In a distributed scheme all the nodes of the tree act as cores. The disadvantage compared to the centralized scheme is that flooding is still necessary to locate which are the nodes that are part of the multicast tree, but flooding stops when a tree node is found.

▼ $T = 2$	Shortest Path Routing	$C = O(N \log N)$
▼ $2 < T < N$	Steiner Minimal Tree	$C = O(N^2 2^{N-T})$
▼ $T = N$	Minimum Spanning Tree	$C = O(N \log N)$

Table 1. *Order of complexity of multicast routing algorithms.*

Multicast Routing Algorithms

Routing algorithms for multicasting make it possible to construct an acyclic (loopless) spanning tree between the participants of a multicast group. Consider a network, as illustrated in Fig. 5, with N nodes, connected by a number E of links and containing a certain number T of participants of a multicast group. Let us consider also that each link is characterized by a single, non-negative real number metric value (reflecting a delay, administrative weight or, in general, an additive measure).

Let us assume also that the full knowledge of the network topology and the set of participants is available at a certain node. Two solutions, an optimized and non-optimized spanning tree construction, will be described below. The first type tries to minimize the sum of the weights (link measure) over the spanning tree. The non-optimized approach has the advantage of being simpler and more suitable to be used in conjunction with a multicast routing protocol for dynamic groups.

Optimized Spanning Tree — Although multicast routing seems at first glance strongly related to the problem of finding the minimum spanning tree, it is actually more difficult than that. This is because in IP multicast the number of participants can be less than N (broadcast). There are two special (and simple to solve) cases of the multicast problem, for:

- $T = 2$, the multicast routing problem reduces to shortest path routing between two terminals. Typical algorithms are due to Dijkstra [4] and Bellman-Ford [5].
- $T = N$, the number of participants equals the number of nodes in the network. This instance of the multicast problem is precisely the *minimum spanning tree* problem. The Prim and Kruskal [5] algorithms are the two most famous algorithms to construct a minimum spanning tree.

The case in which $2 < T < N$ constitutes a very difficult routing problem. This is because there is a large variety of possible ways to construct a minimum spanning tree that only includes the set of nodes in T, using one or more nodes in the network as intermediate or auxiliary nodes. These auxiliary nodes are called Steiner points (e.g., node

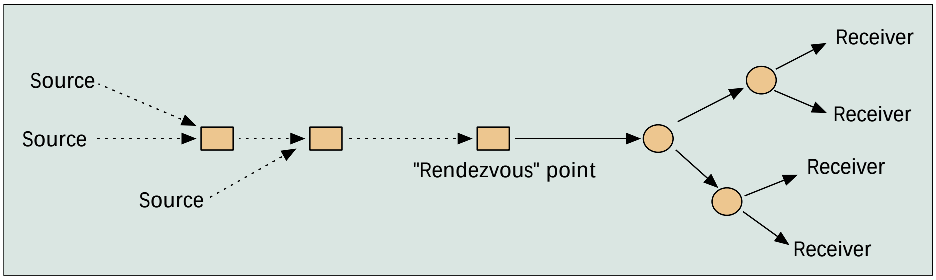


Figure 3. *Uni-directional multipoint to multipoint communication.*

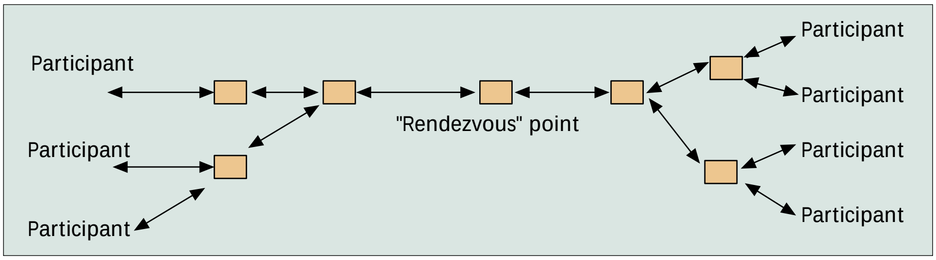


Figure 4. *Bi-directional multipoint to multipoint communication.*

E, D, J in the graph of Fig. 5). The enumeration of all possible ways is as follows. First, we use all nodes in the network and construct a minimum spanning tree. Second, we construct the minimum spanning tree only using $N - 1$ nodes. Since there are $N - T$ Steiner points, there are precisely $N - T$ such sub-graphs of the original graph. Subsequently, we omit two Steiner points resulting in $(N - T)(N - T - 1)/2$ possible sub-graphs in which we again construct the minimum spanning tree. Hence, continuing the reasoning, it is not difficult to verify that $\Sigma_{0 \leq I \leq N - T}$ *Binomial*$[N - T, i] = 2^{N-T}$ sub-graphs exist. Out of all these possible sub-graphs containing all the participants T, the minimum spanning tree in that sub-graph with minimal value of the criterion (sum of the weights over the spanning tree) is the minimum Steiner tree. Hence, the complexity is roughly $O(N^2 2^{N-T})$ (Table 1) and heuristics are required to solve this NP-complete problem. There exists a large literature discussing both the exact Steiner tree problem and its heuristics [5]. The better approach is the one proposed by Kou *et al.* [6]. The Steiner tree algorithm is aimed at a centralized calculation, but Wall proposed distributed heuristics [7]. None of the proposed Steiner tree heuristics[8] can be easily applied in multicast protocols designed to scale for large internetworks [9]. This is because most Steiner tree heuristics are centralized heuristics, and thus require complete knowledge of the network topology. Using distributed heuristics will be advisable since, in IP unicast routing, each node has only a partial knowledge of the network topology. However, this would lead to an increase of the control message overhead of the routing protocol.

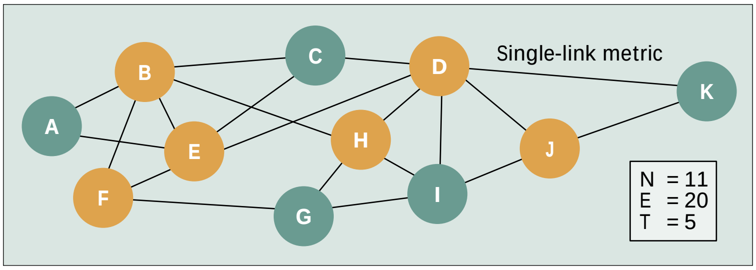


FIGURE 5. *A graph representation of a multicast session involving* T =*5 participants (*N *is the number of nodes,* E *the number of edges, and* T *the participants of the group.*

Non-Optimized Spanning Tree — A class of heuristics to the Steiner tree problem is based on the use of the shortest path between the terminals (typically when $T < N/2$). Another type of heuristics uses minimum spanning trees (typically when $T > N/2$). The interest of heuristics that construct a tree of shortest paths between end-nodes lies in its application to a dynamic set T of participants, as is the case in IP multicast.

For a dynamic set of nodes T, every time the set T changes, a new minimum Steiner tree (MST) must be computed. Apart from the computational complexity, the new MST can be significantly different from the previous one, implying that the forwarding of IP multicast packets may dramatically change, resulting in undesirably transient routing effects. Therefore, most multicast protocols require stable spanning trees that minimize the number of branches where routing changes will occur. The compromise results in the use of a non-optimal spanning tree. One of the simplest approaches is to build a spanning tree by adding one participant at a time, using the shortest path from the new participant to the nearest node of the spanning tree: *shortest path tree* (SPT). Due to the importance of multicast group dynamics, most multicast protocols already have a in-built (i.e., the algorithmic aspect) spanning tree construction. In addition, most methods to calculate SPTs are well suited for a distributed computation.

MULTICAST ROUTING PROTOCOLS

To be of practical use, IP multicast must be efficient, scale well and be incrementally deployable. By *efficiency*, it is meant that setting up and maintaining the group should require only a few control messages. By *scalability*, it is meant that the number of control messages and the amount of state in network elements should grow at most linearly with the number of receivers and the size of the network. By *incrementally deployable* it is meant that it should be possible to add the multicast algorithm to the Internet without requiring a simultaneous change to all routers and endpoints.

The technical challenges of a multicast routing protocol are [3]:

- Minimize the network load (avoid loops and avoid traffic concentration on a link or a sub-network).
- Provide basic support for reliable transmission, that is, make sure that route changes have no side effects on the way data is delivered to group members that remain in the group.
- Consider different cost parameters when optimally designing the multicast routes (the cost parameters can be the availability of the resources, bandwidth, number of traversed links, node connectivity, charged price, end-to-end delay). This is also closely related with maintenance of the optimality of a certain route, when changes occur either in the group or in the network. Thus, a good compromise should be achieved between the optimality of the route and the group dynamics
- Minimize the state stored in the routers, otherwise delivery to a large number of groups is not realistic
- Minimize computer processing at the network nodes

Table 2 lists the set of parameters of a taxonomy to characterize intra- and inter-domain multicast routing protocols (each parameter will be identified by (I) if it applies to an inter-domain protocol parameter and/or (i) for an intra-domain protocol parameter). This taxonomy will make it possible to highlight the impact of certain protocol features on multicast routing.

INTRA-DOMAIN MULTICAST ROUTING PROTOCOLS

This section provides a brief description of the intra-domain multicast routing protocols proposed by the Internet Engineering Task Force (IETF) and by the research community.

DVMRP — The Distance Vector Multicast Routing Protocol defined in RFC-1075 is a *distance vector* routing protocol. The original specification of DVMRP was derived from the Routing Information Protocol (RIP), designed for unicast routing. The major difference between RIP and DVMRP is that RIP calculates the next-hop toward a destination, whereas DVMRP computes the previous hop back toward a source. However, DVMRP performs this computation based on the unicast routing tables constructed by RIP. Thus, it can only be used if RIP is the unicast routing protocol.

DVMRP enables the *incremental deployment* of IP multicast since it supports the use of tunnels to bypass routers that do not speak IP multicast. This characteristic was of extreme importance in order to set-up the first experimental IP multicast network: the Internet Multicast Backbone[3] (MBone). The

[3] *See http://zeus.arc.nasa.gov/mbone.html*

MBone is an overlay network that consists of IP multicast-enabled routers linked via DVMRP tunnels. In fact, it is a collection of autonomously administered multicast regions, defined by one or more multicast-capable border routers. The regions interconnect via the "backbone region" that uses DVMRP as the routing protocol.

DVMRPv3 [11] constructs a *source-based multicast tree* per source, using as routing metric the number of hops in the path. The tree is constructed on demand, that is, when a source transmits the first packet, using the "Flood and Prune" or Reverse Path Forwarding (RPF) algorithm [12]. DVMRP forwards data packets *unidirectionally* along the tree. In order

Parameter	Definition
(I) Independent of the intra-domain multicast routing protocol	Independent of the specific intra-domain multicast routing protocol deployed (e.g., DVMRP, MOSPF, PIM-SM).
(I) Inter-operability with (existing) intra-domain routing protocols.	Inter-operability with the specific intra-domain multicast routing protocol deployed (e.g., DVMRP, MOSPF, PIM-SM).
(i) Independent of the unicast routing protocol	Independent of the specific underlying unicast protocol deployed (e.g., RIP or OSPF).
(Ii) RPF-based	RPF-based protocols are those protocols whose forwarding algorithm performs an RPF check on the incoming interface prior to forwarding a multicast packet on each downstream interface.
(Ii)Uni/bi-directional forwarding	Support of bi-directional trees (trees for which data flows in both directions) or uni-directional trees (trees for which data flows only in the direction of the receiver) or both.
(Ii) Multicast tree types	Source-specific tree or shared tree (via core).
(Ii) Multicast routing algorithm	Algorithm applied to construct the multicast tree.
(Ii) Core selection method	Algorithm applied to determine the core (root domain) location.
(Ii) Loop free	Ability to construct multicast paths that are free of loops in the presence of a network topology subject to failures, congestion, etc.
(Ii) Third-party dependent	Dependent on a pre-configured node (e.g., core) to track changes in the set of participants.
(Ii) QoS/policy-aware	Support of requirements in terms of packet delay, loss, etc.(QoS routing) and/or support of routing between domains according to pre-specified routing policies (policy routing).
(Ii) Security	Ability to make sure that only allowed sources are entitled to send to the group and only entitled receivers are able to receive packets for the group.
(Ii)Incremental deployment	Possibility to add the multicast algorithm to the Internet without requiring a simultaneous change at all routers and endpoints.
(Ii) Deployment stage	How close is the protocol from being deployed? Is it an Internet Engineering Task Force (IETF) standard?
(Ii) Idea brought forth	Main idea brought forth by the protocol.
(Ii) Relevant assumptions	Assumptions of importance made at protocol design.
(Ii) Group management	Efficiency in the presence of a high degree of group dynamics.
(Ii) Computational complexity	Computer processing required (e.g., due to timers and routing table updates).
(i) Latency	Delays incurred to receive the first packet for the group (join latency) and delay from when the source transmitted the packet till its reception by all receivers (end-to-end delay).
(Ii) Traffic concentration on links	Tendency for congestion due to concentration of traffic from several sources on the same links.
(Ii) Control message overhead	Overhead due to protocol-specific control message interchange between routers.
(Ii) Memory requirements	Network resources required at nodes to store and maintain routing state.
(Ii) Scalability	Ability to adapt to a routing domain with many multicast groups, with a high number of participants per group and groups for which the participants' membership changes very often over time.
(Ii) Easy to implement	Complexity of the protocol in terms of the routing, forwarding, or the algorithm's ability to adapt to dynamics in group membership.
(i) IP mobility	Able to route to users reached via a mobile (wireless) network.
(i) IP over ATM	Able to make use of an ATM network for data forwarding (routing will be IP routing). See also [10].

■ **Table 2.** *Taxonomy of multicast routing protocols.*

to avoid the forwarding of duplicate packets (due to routing loops[4]), the incoming interface of every IP multicast packet received is checked against the interface used to send packets (unicast) back to the source (Reverse Path Forwarding check (RPF *check*)). The RPF algorithm takes advantage of the existing unicast routing table to look up routing state information and perform the following tasks:

- When a multicast packet is received, save the source's address *S* and the incoming interface identifier *I*.
- If *I* is the interface used to forward a unicast packet back to the source *S* (RPF check), then:
  –Forward the packet on all interfaces except *I*.
  –Else, the packet is discarded.

DVMRP guarantees the *minimum* packet end-to-end delivery, since the packets follow the shortest path from source to destination. Furthermore, the RPF algorithm is *robust* regarding routing loops. However, transient loops can still occur during unicast routing table updates.

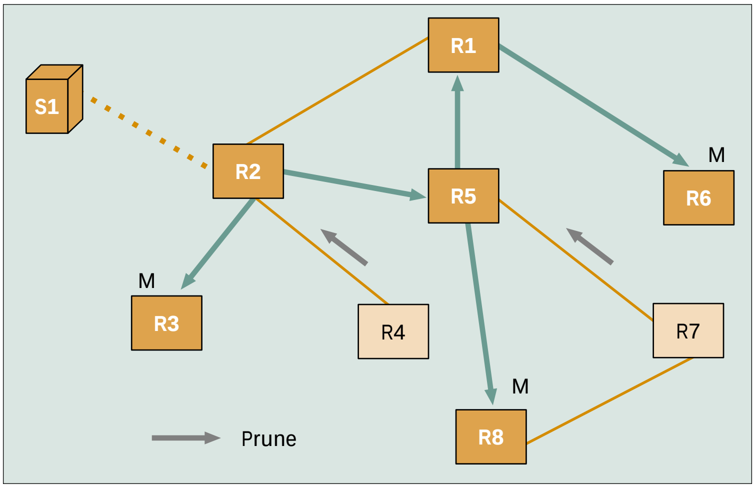


FIGURE 6. *Flood & Prune RPF algorithm: only routers* R3, R6, *and* R8 *have group members. Routers* R4 *and* R7 *prune themselves from the tree and will not receive packets from source* S1.

The RPF algorithm is a "flood and prune" algorithm that takes into account group membership to prune those branches of the tree that do not lead to group members. The Internet Group Membership Protocol (IGMP [2]) is used to detect whether there are group members at the leaves of the tree. This information is passed to routers "upward" the tree in order to prune branches that have no members "downward" the tree (Fig. 6):

- If there is no group member attached to a "leaf" node of the tree, a "prune" message is sent back toward the router that sent the packet (upstream router) indicating that no packets should be send from source *S* to group *G* on interface *I*. A flag is set for interface *I* indicating that the interface has been pruned (prune state).
- If the upstream router receives a prune message from all interfaces on which the first packet was forward, then it will forward a prune command up toward the root (source) of the tree.

This has the following drawbacks. First, the first packet still has to be flooded to the whole network. In addition, after a limited period of time (set according to the dynamics of the membership and the network topology), the prune state is deleted from the local memory and the multicast packet will be flooded to all destinations (periodic prune state refresh). This is done in order to adapt to changes in the network topology. The second drawback is that routers must keep routing state per group and per source. Moreover, apart from the routing state maintained at routers of the multicast tree (also referred as "on-tree" routers), prune state has to be maintained at routers that do not belong to the multicast tree (and thus, should incur no routing burden). This is in the hope that, in the future, new members will be reached via those nodes. If so, a simple "graft" to the tree will add the new member. For groups in which most of the receivers are also sources, or there is a large number of sources and groups, this scheme is very demanding both in terms of memory resources and network utilization [9].

DVMRP uses no special control messages to advertise the source, but its identity is obtained when receiving the first flooded data packet. *Security* aspects (e.g., which source is entitled to send to which receivers) and constrained (QoS) and policy routing have not been foreseen for DVMRP.

DVMRP is a routing protocol easy to implement when compared, for instance, with MOSPF, described later. Its computational complexity is also fairly low (resumed to the RPF check for every packet and maintaining "prune" timers at every node for every active source and downstream interface).

DVMRP assumes that routes between every two nodes are *symmetric* and of equal cost and tunnels can be used when these assumptions do not apply. DVMRP's *deployment* is mainly bounded to the Mbone. Because DVMRP is available public domain (m-routed), it is accessible to all who want to participate in Mbone multicast sessions.

MOSPF — The Multicast Open Shortest Path First protocol is defined in RFC-1584 [13] and *depends* on OSPF, RFC-1583 [14] to construct the unicast routing table. OSPF can use different types of a single link state metric (e.g., delay, number of traversed hops) to express the cost of a path. MOSPF complements OSPF's routing database with a new type of "link state advertisement" records: the group memberships. In this way, MOSPF routers can essentially perform the *RPF check* and join and prune computations locally, since every MOSPF router has complete information about the routing topology and receivers' locations. Thus, on-tree routers can build *source-rooted trees* (shortest path trees) without having to flood the first datagram of each of the sources. The *unidirectional* tree is built on-demand when the first datagram from a source reaches an MOSPF router. Thus, routers that are not part of the tree do not perform any computation for the group.

MOSPF requires *heavy computation* for each source-group combination. Considering that in a routing domain there are as many potential sources as the number of hosts and that the number of groups is likely to grow with the size of the routing domain (also referred to as "autonomous system" in MOSPF), the number of computations that follow any routing update is likely to grow at the $O(N2)$, where N is the number of nodes in the network. The best possible case for *Dijkstra*'s computations is of the order of $O(N.\log N)$.

One of the solutions to improve *scalability* is to carry out the computation of the distribution tree on demand. This means that the distribution tree used to forward packets will be calculated only when the first packet from a source *S* to a group *G* is received. After that, the group membership information is used to prune the branches of the tree that do not lead to any group member. Finally, the multicast packet is forwarded to those outgoing interfaces that belong to the pruned

[4] *Note that routing loops can still occur in transient periods when the unicast routing tables are being updated.*

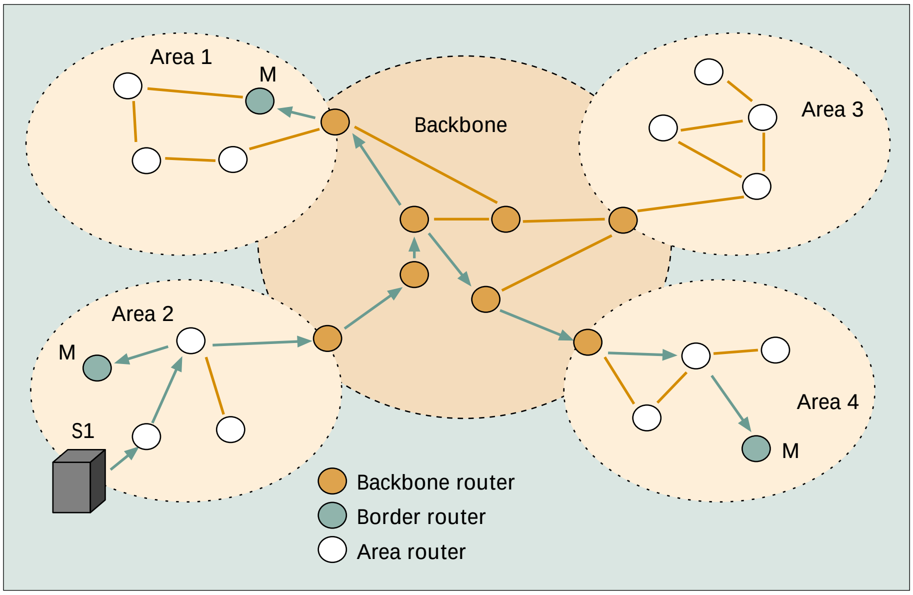


FIGURE 7. *A multicast tree for source S1 visualizing MOSPF division in areas connected via a backbone area. Border routers advertize the existence of members in their respective areas to the backbone area. The border routers of the area for which there are sources of the group will, then, extend the tree to reach the new member.*

multicast tree (source-based tree). Another solution is to divide the routing domain in routing areas inter-connected via a backbone area (Fig. 7). The number of routers per area is limited to a maximum and multicasting between areas is always done via the backbone area. However, there are a number of special cases that make the "shortest path tree" computation in MOSPF slightly more complex than what we have just explained:

- Consider the partition into areas and the need to support multicast between areas.
- Solve the ambiguity resulting from the possibility of having more than on path with equal cost. (Note that in order to avoid routing loops, all routers should construct locally the same shortest path tree.)

The first issue is related to the group membership information "advertised" by the border routers (Fig. 7). Border routers need to advertise to the backbone the presence of at least one member in their respective area. This limits the number of group membership advertisements (LSA) to one per group. For *inter-operating* with other protocols, there are external routers (border routers of the autonomous system). The external routers should not advertise internally all the groups that have been defined on the whole Internet. The solution is to consider, as default, that the external routers are members of all the groups, and thus part of the source-based trees is computed in the backbone. The second issue is solved by giving privilege to broadcast networks as well as paths serving multiple members.

As far as multicast group membership dynamics is concerned, MOSPF advertises *changes in the set of receivers* to all the nodes of the area. This will trigger an update of the routing state at every on-tree node, for each of the sources of the group. If a new source becomes active, its adjacent router just needs to calculate the shortest path tree rooted at the new source, since it has updated information on the set of receivers. Given the above, one can conclude that MOSPF is *slow* to react when there is a high degree of dynamics in the set of receivers and incurs a *high control message overhead* in order to advertise membership changes. Moreover, it maintains a *routing state entry per every source and group address*, even if the source is just transmitting sporadically.

Given the above, it can be concluded that MOSPF is *not scalable* for domains with a large number of nodes. The two-level hierarchy (areas connected to a backbone area) has been one of the steps taken in order to overcome that. However, the hierarchy does not provide any added value for multicast routing since there is no connection between group members and routing areas. Because of all this, MOSPF has *not* been *widely deployed*. MOSPF does not support tunnels nor any feature for *incremental deployment*.

MOSPF supports *constrained routing* in the sense that more than one tree can be constructed per source according to a different (but single) routing metric (e.g., delay, number of hops). This assumes that all nodes in the area have information regarding the different metric values for each of the links in that area. Given this, *policy routing* could be supported in MOSPF associating with every link a metric that reflected the cost of an incoming link. This is possible in MOSPF since every node has an image of the routing topology at its disposal. For the same reason, *routing asymmetries* are not a problem in MOSPF.

CBT — The Core Based Tree routing protocol, RFC 2201 [15], is an attempt to improve the scalability of DVMRP and MOSPF by addressing:

- The periodic flooding to all network sites in order to trigger pruning.
- The need to keep routing state per source and per group.

Building a core-based tree involves the following steps (Fig. 8):

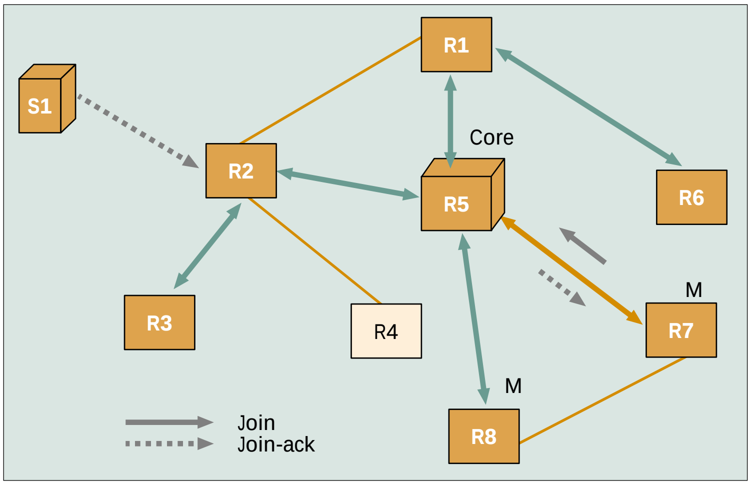


FIGURE 8. *The shared centered tree built by CBT. When a new member* R7 *joins the tree a "Join" is sent in the direction of the core. The "Join-Ack" that follows sets up bi-directional forwarding state in the nodes that constitute the new branch to* R7.

- Locate a "core" router, that is, a fixed point[5] in the network that will be the center of the multicast group *G*.
- Every time a new member wants to join group *G*, it sends a "join" messages via the shortest path toward the core. The join messages are processed at each of the intermediate routers on the path and set up a transient state for the group (incoming and outgoing interfaces).
- If the intermediate router that receives a join command is already a member of the core-based tree, then a join-acknowledgement is sent on the reverse path that the join message has followed, and each of the nodes till the node closest to the receiver:
  –Adds the incoming and the outgoing interface to the set of interfaces for the group.
  –Creates a new state entry that contains the incoming and outgoing interfaces in the list of interfaces for the group.

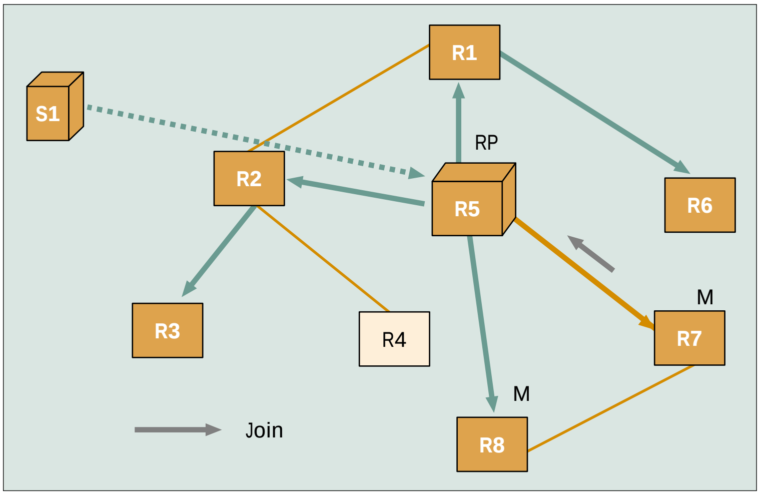


FIGURE 9. *The shared centered tree built by PIM-SM. When a new member* R7 *joins the tree a "Join" is sent in the direction of the RP. The join message sets up uni-directional forwarding state in the nodes that constitute the new branch to* R7.

In CBT, there is no distinction between parent and child interface; an interface is either on-tree or off-tree! When a non-member source sends a packet, the packet is forwarded in the direction of the "core" until it reaches a node that already belongs to the tree. From there the packets are forwarded on *all* the interfaces for the group, except the interface from where the packet has arrived (*bidirectional forwarding)*. Thus, not all packets need to cross the center node to reach the receivers. This characteristic minimizes the influence of the center node in the forwarding of the data. The center node is just there to join the tree; otherwise, it acts just like any other on-tree node.

CBT does not apply RPF checks. Robustness to routing loops is obtained given that the join-acknowledgement should be received via the same interface through which the join-message as been sent. If not, a loop is detected and a new join process starts.

The routing algorithm of CBT is equivalent to building a spanning tree per group that spans to all the group participants plus the core. The CBT tree is a shared tree, that is, the same tree is shared by all the sources of the multicast group. The use of a single (shared) tree per group gives CBT an advantage over DVMRP and MOSPF, since routers that implement CBT only need to maintain one state entry per group instead of one state entry per pair of group and source. This is an important advantage for multicast groups with a large number of sources or when most of the sources are also receivers (interactive groups). Source-specific state can be used in CBTv3, for backward compatibility with other protocols that might use the CBT domain as transit domain. However, source-specific state is only set up on the tree branches spanning the border router and the core.

CBT uses the unicast routing tables in order to obtain the next hop router to the core. However, any of the existing unicast routing protocols can be applied to use in conjunction with CBT. In contrast with DVMRP and MOSPF, which are linked with a specific unicast routing protocol (RIP and OSPF, respectively), CBT is *independent* of the unicast routing protocol.

The disadvantages of CBT when compared with protocols that use source-specific trees is that, since CBT uses a shared tree, it concentrates traffic on fewer links than protocols that use source-based tree schemes. However, because it uses bi-directional forwarding, it enables any node to fan-out traffic rather than requiring that traffic be sent to the core prior to forwarding (unidirectional forwarding).

In order to scale in the presence of a high number of (high rated) sources, CBTv1 proposed the use of multiple cores. This proposal was found to be not robust by Shields, who proposed the Ordered Core Based Tree (OCBT) protocol as a solution for the problem [16]. Hence, in CBTv2 [17] it was decided that only one core should be supported per multicast group in order to make the protocol easy to implement.

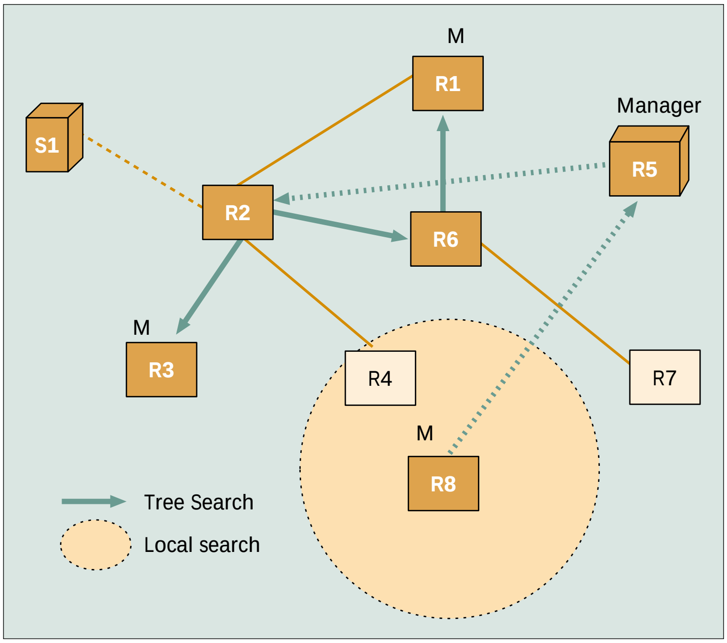
CBTv2 assumes symmetric routing paths and thus is not suited for use in conjunction with QoS or policy routing. As far as security is concerned, the core or center node can be used to send information to the receivers on the set of secured sources, and receiver authentication mechanisms can also be easily added to the protocol (see also [18]).

Choosing a center for the tree is an NP-complete problem in a multi-sender, membership-dynamic environment. There are several algorithms [19, 20] that have been compared in order to localize the best node (center) that minimizes the delay of the transmission from sources to receivers. These algorithms require the complete knowledge of the network topology and the group membership. However, only for relatively static groups and groups for which the receivers are all confined to a domain, does the computation of the center node make sense. Thus, for the majority of the groups, the center node can be chosen as the router closest to the first receiver of the group or the router closest to the main source.

PIM-DM — The Protocol Independent Multicasting-Dense Mode (PIM-DM [21]) was designed to be used for groups with a large number of members (dense mode). As in DVMRP, "Flood and Prune" Reverse Path Forwarding (RPF) is used in PIM-DM. However, PIM-DM is simpler than DVMRP because it does not construct unicast routing tables. In fact, PIM-DM is independent of the unicast routing protocol; it simply assumes that a unicast routing protocol exists to construct unicast routing tables and that the unicast routes are symmetric. The RPM algorithm as used in PIM-DM is:

- If a router receives a multicast packet from source *S* to group *G*, it first checks (in the unicast routing table) if the incoming interface is the one that is used to send unicast packets toward *S* (RPF check):

[5] *Note that the core is not necessarily a member of the multicast group, although it wouldn't be such a bad idea to do so, since in this case the core will be located close to at least one of the members.*



■ **FIGURE 10.** *The local search and the multicast tree search (inspired by Fig. 2 of Faloutsos* et al. *in [32]).*

–If so, the router forwards a copy of the packet on all the interfaces for which it has not received a prune message.
–Else, the packet is dropped and a prune message is sent back on the incoming interface.

- If all interfaces have been pruned, a prune message is sent back on the incoming interface.

The difference between DVMRP and PIM-DM is that in DVMRP, prior to forwarding to a certain interface, DVMRP makes sure that the interface leads to a node that will recognize the local node as a node that is in the shortest path between it and the source (poison-reversed route). PIM-DM decides to accept additional overhead in order to simplify the forwarding algorithm. Apart from this, the protocol is very similar to DVMRP and thus, all that has been stated for DVMRP applies to PIM-DM also.

PIM-SM — In the Protocol Independent Multicasting-Sparse Mode [22] (RFC 2362 obsoletes RFC 2117), the notion of a center node of a multicast tree is interpreted as a rendezvous point (RP) or meeting point where sources can meet receivers and vice versa. The PIM-SM (Fig. 9) join mechanism is the following:

- A receiver joins a group by sending a "join message" toward the RP (receiver-driven protocol).
- The "join message" is processed by all the routers between the receiver and the RP, which will save the status information for the group. Thus, a new branch of the distribution tree for the new member is set-up.

Now, let us see how packets are sent to a group:

- A source starts sending a data packet to group *G* encapsulated in a unicast packet directed to the RP. The source does not have to know who the group members are; only the RP address for a certain group is required.
- Upon reception of a multicast packet, the RP will de-capsulate the packet and forward it to all the interfaces of the distribution tree that lead to group members.

PIM-SM and CBT are protocols designed for groups where members are sparsely distributed over the routing domain. Comparing PIM-SM's forwarding mode with CBT's, it can be observed that in PIM-SM, packets sent from a source must first be transmitted to the RP (*uni-directional forwarding*, Fig. 1). In CBT, only the members that share the same tree branch as the core will receive via the core (*bi-directional forwarding*, Fig. 2). Another difference between PIM-SM and CBT is that if the data rate of the source is over a certain threshold, a source rooted tree can be used in PIM-SM instead of a RP shared tree:

- The router will then send a "join" packet toward the source and a "prune" toward the RP.
- Routers that are closer to the leaves of the RP multicast tree will now also automatically switch to the "source rooted tree" route.
- The source will continue to send a copy of its packets to the RP, considering that there might be members in the group that are still receiving packets via the RP rooted tree.

In addition, PIM-SM uses "semi-soft" states. Semi-soft state is a state that has to be refreshed by a join message sent periodically. If the join message is not received within a time-out period, the state entry is deleted.

PIM-SM forwarding uses the RPF *check* on the incoming interface to track looping packets. The unicast routing information is derived from the unicast routing tables, *independently of the unicast routing protocol* that constructed them.

In PIM-SM, even if a receiver has switched to source rooted trees for all active sources, state still needs to be maintained for the RP rooted tree at the shared tree routers. This is in order to receive packets from a new source of the group. A more efficient way of managing state in PIM-SM is given by Billhartz *et al.* in [23], after assessing the performance of PIM-SM compared with that of CBT. It states that PIM-SM is a complex routing protocol given the size of the routing table and the impact of the timers on the operating system overhead for a large number of members that can potentially become sources. In spite of these results, PIM-SM is a widely deployed protocol.

The considerations made for CBT regarding use of symmetric paths and security aspects apply also to PIM-SM. As far as RP location is concerned, PIM-SM uses a mechanism built in the protocol to advertise the set of possible cores within a domain (bootstrap mechanism). Each node that receives a join message for a group *G* can then, via a hashing function, determine the RP node for a certain group and send the join message accordingly. This mechanism has been proposed in order to provide a fast regeneration of the tree in case the current RP node fails.

MIP — The Multicast Internet Protocol [24] (MIP) is similar to PIM-SM in the sense that it makes possible the construction of both shared trees and shortest-path trees and is independent of the underlying unicast routing protocol. The novelty in MIP is that the construction of the multicast tree can be sender-initiated, receiver-initiated, or both. These two modes of operation are interchangeable and make it possible to tailor the construction of the tree to the particular nature of the dynamics of the multicast application and the group size. Hence, the sender-initiated type of construction is well suited for small groups, in which it is manageable for the source to know the identity of the receivers (e.g., a video-conferencing session involving only a few sites). On the other hand, the receiver-initiated construction is well suited for groups with a large number of receivers and is based on the explicit join mechanism of CBT and PIM-SM; the receiver only needs to know the address of a router in the multicast tree. This router will act as the core or RP node in CBT or PIM-SM, respectively.

MIP does not use preconfigured (core) meeting points in the manner of CBT and PIM-SM. Instead, the shared tree is rooted at a router that is either a source or a receiver of the multicast group. Thus, the issue of core placement is avoided. The root node address is known by all participants of the multicast group just as the core address in CBT. To obtain the root of the shared tree in case of failure, MIP employs a ring protocol between the root and all its neighbors. Only routers on the ring can become the root of the shared tree. This scheme is more dynamic than the scheme used in PIM-SM, in which a static ranked list of possible cores/RPs needs to be advertised to all the routers.

The shared tree used in MIP is a bi-directional tree. Thus, RPF checks are avoided for each multicast packet received. If a new source wants to send to an existing multicast group, it first joins the shared tree of the group in the same fashion as a receiver. When a source finishes transmitting to a group, it tears down the part of the routing structures for the group that it solely used.

In CBT and PIM, data packets can experience arbitrarily large delays in the shared tree because the core or the RP can become poorly placed as a result of failures or recoveries of links and routers in the network. In MIP, as in PIM, those receivers on the shared tree who want optimal delay can switch from the shared tree to shortest path tree for chosen sources.

MIP has been proposed in order to provide a means of constructing a multicast tree that is free of loops, even when the underlying routing tables are inconsistent and contain routing loops. This is because MIP uses *diffusion* operations [25, 26] to construct the multicast tree, tear down the tree, add/remove a new member, and update cost metrics after changes in network topology. Basically, a router initiates an operation by sending queries to all of its neighbors and waits for replies from the neighbors to detect whether the operation has been successfully completed (positive acknowledgement) or if it cannot be completed (negative acknowledgement). Each neighbor sends a reply to the query after it terminates the operation. This might require that the router sends its own query and receives replies from the corresponding neighbors. The mechanism to detect loops is very simple. If a router that has just issued a query receives back a query message for the same operation before having completed it, then the router assumes that the operation requested has failed and sends a negative acknowledgement to the neighbor that had initially issued the query. Hopefully, an acknowledgement is received on some other interface indicating that the operation has been successful.

The use of diffusion is the added value of using MIP, since the multicast tree is guaranteed to be loop-free. MIP's authors have pointed out in [27] cases where both CBT and PIM-SM suffer from temporary loops resulting from the use of inconsistent unicast routing information, and claimed that neither protocol has been verified to provide correct multicast trees in periods when changes in topology cause the unicast routing tables to be updated. However, the diffusion mechanism is heavy in terms of control overhead, and that explains why in PIM and CBT temporary loops are accepted for the sake of keeping the protocols simple, since unicast routing tables do not change so often.

INTER-DOMAIN MULTICAST ROUTING PROTOCOLS

The following set of surveyed multicast routing protocols begins with those protocols that address multicast routing in directed graphs where asymmetric routing topologies tend to prevail. YAM and QoSMIC address constrained multicast routing and PTMR policy routing. The remaining protocols, MSDP, BGMP, SM, and EXPRESS, attempt to construct a multicast tree between domains or tree branches of an existing intra-domain tree that expand inter-domain. Policy and constrained routing are not addressed by these protocols.

As an introduction to routing in asymmetric topologies, Hodel states in [28] that asymmetries can be originated by access, transit and route selection policies that, by not being well concerted or even published, prevent coordination altogether. Asymmetries may also arise when routing boundaries are traversed, either between regions with different routing protocols or in hierarchical routing, between two separate layers. In constrained routing (also called QoS routing), asymmetries for a specific service quality may result from the available network resources. Similarly, if the Reverse Path Forwarding (RPF) algorithm is applied, the forwarding path may not coincide with the reverse optimal path according to prevailing requirements for a given type of multicast traffic. Furthermore, path divergence may be evoked by load splitting or tie-breaking across multiple equal-length shortest paths.

In the Internet the routing topology is far from being symmetric. Traceroute experiments [29] investing route asymmetries showed that sequences of cities and autonomous systems (ASs) visited by routes in the two directions of a virtual path differ quite frequently. It was found that overall, 50 percent of the paths include an asymmetry in terms of cities. In terms of ASs, about 30 percent path asymmetries occurred, mostly due to a single hop in one direction.

YAM — The "Yet-Another-Multicast" (YAM) routing protocol is a step mark in proposals for multicast routing protocols that make it possible to choose branching points to an existing tree other than using the Reverse Shortest Path, that is, without assuming an underlying symmetric network topology. Indeed, research studies have shown that the Internet, as of today, has 30 to 50 percent asymmetric paths [29].

In any case, when a multicast protocol needs to be applied both intra-domain and inter-domain, asymmetric paths become an issue difficult to be ignored. This is because inter-domain routing policies might be such that two distant domains might not agree on the same forwarding domain, giving rise to distinct paths for opposite directions. Another possible source of route asymmetries is when cost measures, rather than just hop-count, are used to build a path.

Considering the above, a protocol that can either:

1)Use a link state protocol to provide information to calculate alternate paths (subject to constraints)

2)Discover (potentially) multiple paths from an existing tree onto a joining node and afterward make a choice based on a certain criteria

can be applied in the context of asymmetric paths (e.g., a satellite network with downlink and no uplink); inter-domain routing; or constrained (QoS) routing based on user selected parameters.

YAM aims at the second approach described above. QoS routing in the context of YAM consists in the use of multiple static parameters of the route (e.g., link capacity, reliability) to select a path, as suggested in [30].

YAM builds shared trees that have the capability to provide multiple routes to connect a new node onto an existing tree. YAM's join mechanism is split between intra-domain and inter-domain. Inside a domain, YAM follows a similar design rational as CBT, building tree branches triggered by the joining node and operating independently of the unicast routing protocol. The mapping (core, multicast group) is substituted in YAM by egress node and multicast group. The egress node functions both as core or meeting point and as the border node for a domain.

In the absence of state at the egress node for the target group address, a multicast tree branch needs to be constructed inter-domain. The egress node starts the discovery process of other on-tree nodes in other domains issuing a one-to-many join. The discovery process is thus limited to egress nodes of other routing domains. If a node that receives an inter-domain join has no state for the group, it multicasts (instead of unicasting to the core) a join message to the "All-YAM-routers" multicast address. The join message is forwarded based on the Reverse Path Forwarding [12] algorithm, that is, the message is sent out to its other interfaces if it arrived from the interface to forward packets back to the source (no state is maintained): *one-to-many join*. When the message is received by an on-tree node that has state for the group, the message is terminated and the on-tree node responds with a unicast join-request toward the egress node that initiated the one-to-many join. This join request installs temporary state along the path. If there is more than one on-tree node that replies, then state is maintained for more than one possible path. The egress node can then choose from several inter-domain paths to join an existing tree. Once the path is chosen, the node sends a join acknowledgement via the path chosen to graft the tree. The temporary state is substituted by definite state on the nodes of the path. The intermediate state in nodes of the remaining paths is left to expire.

The fact that YAM suggests that state be maintained for more than one path from a branching node to the egress node inter-domain is the main criticism of this approach. Although this is required only once for each domain, if the membership of the groups is very sparse and the level of membership dynamics quite high, then scalability becomes an issue.

The core (root) of the shared tree is established when the very first receiver joins a group. The core is thus the egress node for the domain where the first receiver resides. Subsequent one-to-many joins from egress nodes in other domains graft this egress node in order to establish the tree.

The work published in 1997 by Carlberg and Crowcroft [31] on what was going to be called YAM (see bibliography) was the birth of another protocol based on the same idea, by Faloutsos *et al.* in 1998: QoSMIC [32].

The issues left for further study in YAM are:

- The ability to support efficiently non-member senders. The problem arises because low-rated sources or not very active sources may not tolerate the delay involved in the discovery process.
- Inclusion of policies both rational (e.g., number of fan-in and fan-out links of the tree node) and irrational (e.g., policies that are not related to the capability of the network such as pricing and access lists) to determine how constrained routing (also referred to in the literature as QoS routing) is used to select the best branch of the shared tree.

QoSMIC — The Quality of Service-sensitive Multicast Routing Protocol [32] follows YAM in the quest for QoS routing in IP Multicast. QoS is defined as the user-perceived quality of a service.

QoSMIC distinguishes between two terms: Quality of Service (QoS) and Quality of Route (QoR). QOR refers to the multiple static parameters of the route (e.g., link capacity, cost or reliability). QoS refers to the use of dynamic metrics (e.g., available bandwidth, current delay) that reflect the characteristics of a path at a given moment. YAM considered only QoR. QoSMIC is compatible with any metric of the routing quality of a path. However, routing with dynamic metrics is preferred since dynamic metrics can be proactive in case of link congestion.

With QoSMIC it is possible to construct both a shared and source-specific trees. As with PIM-SM, QoSMIC starts building a shared tree and individual receivers switch to a source-specific tree when necessary. The mechanism to graft a branch of the tree strongly resembles that of YAM described earlier for the inter-domain case.

QoSMIC introduces the concept of *manager* for a multicast group. The manager facilitates the joining of new group members, as does the rendezvous point (RP) in PIM. The difference between the RP and the manager is that the tree is not rooted at the manager. This way, the location of a manager has a small effect on the topology of the tree and managers can be substituted in case of failures without any data loss. Because administrative routers (e.g., manager) are not necessarily part of the tree, QoSMIC will introduce additional robustness in case of failures in such routers.

QoSMIC follows a receiver-driven mechanism to add a new branch to an existing tree. The designated router of the new member will start a discovering procedure for multiple paths to reach promising routers already on the tree. This search is conducted in two phases: one from the side of the new member (local search) and one from the side of the tree (multicast tree search).

The local search consists of the designated router close to the new member sending a "Bid-Request" using Reverse Path Forwarding, with scope limited by the use of the time-to-live field (TTL). Every on-tree router is considered a candidate branching point and replies with a "Bid" message unicasted to the designated router. The "Bid" message, on its way from the candidate branching point to the receiver, collects information on the expected performance of the path, based on dynamic metrics (e.g., delay, load). The designated router for the new member will choose among the alternate candidates according to a certain QoS or QoR criteria. After that, a "Join" message will be sent over the selected path that will create *soft state* in the nodes of the path. The local search is heavy in terms of network control overhead and thus is supposed to be used only intra-domain and with a limited TTL value.

The multicast tree search works as follows: the designated router contacts the manager router[6] and the manager will send a "Bid-Order" to the root[7] of the tree. Some subset of the on-tree routers are selected as candidates to make a "Bid" using a distributed election (see [32] for details on candidate selection algorithms). The candidates unicast "Bid" messages to the designated router of the new member. From this point onward, the process of candidate selection and subsequent "Join" is identical to the local search case. The join procedures for source based trees are exactly the same as for the shared tree, and the switching mechanism from the shared tree to a source-based tree are equivalent to the ones proposed for PIM-SM. The local and multicast tree searches are illustrated in Fig. 10.

QoSMIC is a complex protocol that claims to be scaleable enough to be deployed both intra-domain and inter-domain. In practice, only the multicast tree search join procedure has the potential to scale inter-domain since the control overhead can be quite high depending on the number of alternate paths to be offered. Moreover, the multicast state should be main-

[6] *The Manager is suggested by the authors to be the Border Router nearest to the member initiator of the bidding process. It is also suggested to have one or more Managers per routing domain. For a source based tree, the source is selected as the Manager to simplify the tree administration.*

[7] *The root of the tree in QoSMIC coincides with the router next to the first receiver of the group.*

tained only after the candidate branching point is selected (triggered by the "Join" message from the designated router) and not before hand, as in YAM. The question that can be asked at this point is, "How much routing state is required to be maintained in on-tree nodes?" The routing table entries made by QoSMIC are indistinguishable from that of PIM-SM or CBT. Hence, once the "Join" is complete, QoSMIC does not require more state than PIM-SM or CBT in inter-domain and intra-domain routers.

The interesting idea to bear in mind with QoSMIC and YAM is that the path is constructed from the tree to the new member and not on the reverse shortest path, as with currently deployed multicast routing protocols. Thus, YAM and QoSMIC can be used in directed networks where asymmetric paths are likely to exist.[8]

QoSMIC is proven to be loop free. A public domain implementation of QoSMIC will be available in the future, but for the time being only simulation results exist to test the protocol [32].

PTMR — The Policy Tree Multicast Routing protocol [28] is a single layer protocol that extends the PIM-SM protocol with a policy-dedicated delivery mode. PTMR constructs the multicast tree branches that connect several routing domains (or between policy domains if these do not coincide with the routing domain). What is so special about PTMR is that it constructs multicast trees which, even under asymmetric conditions, readily comply with imposed macroscopic policies still supporting shortest path from source to receivers.

In a routing environment where routing policies are applied, the guarantee that symmetrical paths will exist between two network addresses is broken. Asymmetrical routing conditions lead to multicast delivery on an alternative, unintended path, if a forwarding algorithm based on Reverse Path Forwarding is used. Even more severe is the case whereby on the RPF path multicast traffic cannot flow at all, and group members will never receive any traffic from the source(s). Divergent paths in the two reverse directions can be avoided by means of a genuine "come-from-routing" (CFR) mechanism. However, this implies that routers need to be provided with specific multicast routing information bases (RIB) containing actual CFR paths.

The easiest way to support CFR-type routing between routing domains is via BGP4+ [33] multicast-specific path vector updates. This way, ASs can exert control over which multicast path traffic from a given network is delivered. In intra-domain routing, in order to install genuine CFR routing, one would expect that many unicast routing components would more or less have to be duplicated. There is a way to avoid this if "come-from" cost can be treated as an additional set of metrics (applied to multicast traffic). CFR can then be accommodated by a conventional routing protocol that supports routing based on alternative metrics (e.g., type of service routing in OSPF).

Policy routing in the context of PTMR is seen as constrained routing subject to a routing domain's qualitative requirements (e.g., domain *A* does not accept to forward traffic from domain *B* to domain *C*) and its quantitative requirements (e.g., bandwidth, delay, jitter or loss). PTMR gives no consideration to how the discovered policy routes have been established or what route selection criteria has been involved. PTMR builds on top of BGP and its extensions, BGP4+, in order to have a mechanism to advertise routes between different domains that reflect the pre-specified policy constraints. BGP's path vector announcements contain the sequence of routing domains through which a network is reachable, or when applying RPF, the sequence of routing domains on which traffic is to be sent to multicast sources in the advertised networks.

PTMR chooses to apply a *source originating* delivery tree because it makes possible a routing mechanism that does not consider how the discovered path has been established and what route selection criteria was involved. That is, source originating tree deployment is independent of the underlying policy model as well as the prevailing routing procedure. In particular, enforcing source demand policies can be easily supported via header routing, whereby the marked routers provide a loose path setup.

In source originating tree deployment, the path from the source *S* to the receivers needs to be previously marked. For this purpose, some sort of control packet is sent hop-by-hop toward each of the receivers, and the packet is tagged with the multicast group address *D*. The control packet triggers the creation of (*S*, *G*) state at each of the nodes along the path used to forward packets from that source and group address. In order to dispatch the pilot packets, sources need to know the group member locations. However, in order to reduce the join delay in a sparse environment, source/member handshake needs to be initiated by the receivers. In consequence, a mechanism needs to exist by which group members learn about active sources. PTMR chooses to do this via a meeting point approach such as the one used in the shared tree of PIM-SM. PTMR answers the implosion of control messages to be processed by the router closer to the source, by applying receiver-initiated, source originating tree deployment to an inter-domain level only. A single router (e.g., the border router in a domain) will then request the source's first hop router to deploy a domain-specific delivery path. Although control message congestion at the sources is still an issue, PTMR proposes a solution to alleviate it for every new source that becomes active. The idea is that the (periodic) marking cycles[9] originated by the router closest to the source are delayed by the initiating last hop border router which holds back the join message for a random delay time within a given interval.

The PTMR architecture is characterized by a structure called policy tree, which is a multicast extension of the policy route as defined in RFC 1102. Group members receive source location information by way of the PIM RP (shared) tree following the PIM-SM routing and forwarding mechanism (RPF-based). If the receiver would like to switch to a source-specific tree it does so according to the PIM-SM mechanism. If the source is not located in the same domain then a policy tree is constructed using PTMR. The process to construct such a tree consists in the egress multicast border router (MBR, also called in the following the last-hop peg router) for that source network to send a request message in the direction of the source. This MBR induces, with the request message, that the first-hop router for the active source will establish a policy route from it to the requesting last-hop peg router. After that, the first-hop router for the source will return a pilot ("Mark") message to the requesting last-hop peg router, which mimics multicast packets sent by that source. The policy route is

[8] *If RPF based protocols are applied in asymmetric routing conditions the resulting forwarding path could be sub-optimal (e.g., links have different metric values for opposite directions) or non-existing (e.g., satellite links with only downlink).*

[9] *Initially, the marking cycles are triggered by a switch over from the shared tree to a source specific tree, when the receiver manifests the need for an improved delivery service.*

pegged out to the "Mark" message by the sequence of MBRs (peg routers) through which the "Mark" message enters transit multicast domains, ending with the peg router (MBR) for the multicast domain where the receiver is located. When the requesting MBR receives the requested "Mark" message, it multicasts an "Announce" message in the RP-shared tree containing the address of the (new) last-hop peg router to the receivers. Finally, with this information the routers close to receivers join toward this new last-hop peg in order to receive traffic from the source via the policy tree.

PTMR performance and policy control are restricted by the following:

- Congestion of control messages at the multicast sources, introduced by the construction of the multicast tree originated by the source.
- Possible extent of policy-sensitive path aggregation.
- Transit (intra-) domain delivery conditions.

The PTMR proposal demonstrates that the solution for the problem of policy-sensitive multicast routing is inherently complex. Because PTMR builds on top of another (existing) multicast routing protocol for intra-domain routing, PIM-SM, it is applicable only to homogeneous multicast routing environments, that is, whenever PIM-SM is the multicast routing protocol deployed intra-domain.

PTMR suggests the use of a policy tree per active source, assuming (for scalability reasons) that for less active sources (in terms of bit rate) or sporadic sources, delivery of multicast messages is via the PIM-SM shared tree. However, PTMR does not provide a complete solution of the inter-domain routing problem. For instance, it does not mention that if the sources are not located in the same routing domain as the RP node, another mechanism is still required to discover the sources of a multicast group (see MSDP below).

Finally, PTMR tries as much as possible to fuse source-originated paths toward the multiple last-hop peg routers. However, if divergent routes (caused by policy or link cost metrics) are obtained for group members in different last-hop multicast domains, then these routes might merge at any intermediate peg router. This can originate duplicate packet delivery if more than one parent-peg is accepted for the same (*S*, *G*) pair. PTMR proposes two solutions for this. The first is to give preference to the new parent-peg if the old parent-peg entry has been used for awhile. The second consists of using last-hop peg addresses as flow labels and give facilities for load spreading among different policy trees that might cross at the same points. On this matter, the author argues that divergent paths caused by link cost are not a sufficient requirement to introduce more complexity into the protocol and points out that an adequate network dimensioning should provide a much simpler solution.

PIM-DM/PIM-SM — The combination of PIM-DM and PIM-SM has been one of the first proposals for inter-domain multicast routing. The proposal consists of using PIM-DM as the intra-domain routing protocol and PIM-SM as the inter-domain routing protocol. Thus, PIM-SM will construct a shared tree (and source-rooted trees) that connect the source-specific trees maintained at every domain using PIM-DM.

The set of RPs is advertised inter-domain to all border routers in order to provide a mapping between each multicast group address and the respective RP.

This approach can be applied to a well managed corporate network but not to the Internet since the mechanism to advertise RPs and the maintenance of the soft state entries in PIM-SM will consume a large amount of control overhead. The amount of state entries to be maintained is also not feasible for an inter-domain protocol (one state entry for the shared tree and then as many as the number of source-specific trees available). Policy routing, which is an inter-domain routing requirement, is also not addressed by this approach.

PIM-SM/MSDP — The Multicast Source Discovery Protocol (MSDP) is the short-term IETF standard proposal for connecting shared trees without the need to construct an inter-domain shared tree [34]. It is envisioned that the Border Gateway Multicast Protocol (BGMP) is the long-term standard for constructing inter-domain shared trees. MSDP is applicable to shared-tree protocols such as PIM-SM and CBT, as well as other protocols that keep active source information at the borders (e.g., MOSPF or PIM-DM with domain wide reports [35]).

MSDP is used to connect multicast domains together. If the domains run PIM-SM, each PIM-SM domain uses its own independent RP(s) and does not have to depend on RPs in other domains to forward traffic to its own domain. MSDP is based on a different paradigm than protocols that construct an inter-domain tree between domains and then inter-operate with the multicast routing protocol intra-domain to make sure that connectivity is not broken at the border routers. Rather than getting trees connected, MSDP proposes to get sources known to all intra-domain trees. At first sight it appears not to be a scalable approach, but the authors point out that the trick is in the implementation.

MSDP works in the following way. An RP in a domain has a MSDP peering session with an RP in another domain. This peering session will be made up of a TCP connection in which control information is primarily exchanged. Each domain will have a connection to a logical topology where the nodes are the RPs in each domain and the links are the TCP connections between peering RPs. This virtual topology is congruent to the BGP paths between domains used to forward multicast traffic from one domain to another domain (these paths are advertised by BGP4+ or MBGP [33]).

The purpose of the virtual topology is to enable domains to discover multicast sources in other domains for internal multicast groups. If these sources multicast traffic for a group that has members in the domain, then the normal procedures in PIM-SM are used to send traffic from the source to these receivers (source sends the data encapsulated in a PIM-register message to the RP in the domain). The RPs that have active sources in their own domain send "Source-Active" (SA) messages and send them to their MSDP peers. The SA message contains the IP address of the source, the multicast group address, and the IP address of the RP that has sent the SA message. Each MSDP peer receives and forwards the SA message in a peer-RPF flooding fashion. This consists of checking if the MSDP peer has received the SA from a RPF peer (toward the originating RP). If so, the SA is forwarded to all its MSDP peers; otherwise it is dropped. The BGP4+ routing table is examined to determine which peer is the next hop toward the SA originating RP. A procedure similar to Split Horizon with Poison Reverse can be used to constrain the forwarding to only those children of the peer that advertise routes that use it as the next hop toward the originating RP.

When each MSDP peer (RPs for their respective domains) receive an SA, they check if they have any group members for the group. If so, the RP triggers an (*S*, *G*) join toward the source that sets a branch of the source tree to its own domain (*flood-and-join* model). If leaf routers decide to switch to the source-specific tree, they can do so using PIM-SM procedures since they already know the location of the source.

RPs that originate SAs do it periodically as long as there is data being sent by the source. However, intermediate RPs do not send periodic SAs on behalf of sources other than the sources located in their domain. The choice for the time for the refreshing period is the critical issue in MSDP in order to minimize the time it takes for a new receiver to join after a source has been active. A known problem is that associated with SA messages for sources that are very bursty and sending only for a very reduced amount of time. It might happen that by the time the RP joins, the source might just have resumed sending packets (*bursty source* problem).

Using MSDP, there is no shared tree built across domains. Therefore, each domain can depend solely on its own RP. SA state is not stored at all the MSDP peers, but the one that originated the SA. Data could already be encapsulated in SA messages for low-rate bursty sources. MSDP peers could cache SA messages and if they do, MSDP peers can get MSDP (S, G) state sooner and reduce join latency for new joiners.

The main advantage of MSDP is that it is easy to implement. Robustness is achieved using RPF checks prior to forwarding data. Multicast route policies are respected since BGP4+-advertised paths are followed. However, the maintenance of state inter-domain for every source that is not located in the same domain as the receivers is an issue hard to be ignored. Security aspects are similar to those discussed before for PIM-SM.

MASC/BGMP — The Border Gateway Multicast Protocol is an inter-domain multicast routing protocol that has been designed to inter-operate with any multicast routing protocol deployed intra-domain [36]. BGMP is associated with another protocol, the Multicast Address-Set Claim (MASC) protocol to form an architecture for inter-domain multicast routing [36].

MASC forms the basis for a hierarchical address allocation architecture. MASC temporarily and dynamically allocates multicast address ranges to domains using a "listen and claim" approach with collision detection. In this approach, child domains listen to multicast address ranges selected by their parent, select sub-ranges from their parents' range and propagate the claims to their siblings. The "claimers" wait for a suitably long period to detect any collision before communicating the acquire range to (1) the domain's multicast address allocation server [37] and to (2) other domains, through BGP, as multicast-specific routes. MAAS can then allocate, from its multicast address range, individual multicast addresses to groups initiated in their domain.

BGMP requires that each multicast group be associated with a single root or core and constructs a shared tree of domains, similarly to other shared tree protocols (e.g., PIM-SM and CBT). However, in BGMP, the root is an entire domain rather than a single router. BGMP is based on two main assumptions:

- That a rendezvous mechanism, whereby members get to know the identity of the sources without the need for global broadcast, is the most convenient for inter-domain multicast routing.
- That specific ranges of the class D space are associated (e.g., via MASC) with various domains. Each of these domains is chosen to be the root of the shared tree of domains for all groups whose address is in its range. This is because the root domain is very likely to be the domain initiator of the multicast group.

The actual number of multicast addresses, claimed by a domain using MASC, is a trade-off between two competing factors:

- If the number of multicast addresses available is high, the domain will become the root domain for a large number of groups.
- If the claimed address range is sufficiently large, groups initiated locally can get multicast addresses from the domain's range, thereby becoming locally rooted.

This is an area that is still under investigation.

The choice of the root of a shared tree in inter-domain routing has implications both in terms of policy and performance as it relates to end-to-end delay. In the intra-domain case, any router can be entitled to become core for the group. This is because the emphasis in the intra-domain case is on load sharing and the penalty on non-optimally located cores is not significant. The same cannot be said in the inter-domain case, that is, all possible root domains cannot be treated as eligible candidates. In inter-domain routing there are administrative issues concerning the ownership of the root domain and a greater risk from poor delay performance due to the location of the root. This is the reason why in BGMP the root domain has been chosen to be the domain of the group initiator in the hope that this domain will source a significant portion of the multicast data.

BGMP uses the routes advertised by BGP to construct the multicast trees for active multicast groups. Since inter-domain routing involves the use of resources in autonomously administered domains, the routing policy constraints of such domains need to be accommodated. BGMP follows policy for multicast traffic using the selective propagation of group routes in BGP4+ (multicast extensions of BGP4, RFC 2283 [36]).

BGMP runs on domain border routers and constructs a *bi-directional shared tree* that connects individual multicast trees built in a domain. Hence, the border routers also run protocols for multicast routing intra-domain (e.g., PIM-DM, PIM-SM). Such intra-domain routing protocols are also referred to as Multicast Interior Gateway Protocols or MIGPs. The modulo of the border router that runs an MIGP is referred to as the MIGP component; the modulo running BGMP is referred to as the BGMP component. It is up to the MIGP component to inform the BGMP component about group membership in the domain. This triggers BGMP to send "Joins" and "Prunes" border router to border router until the root domain or a border router that is already on the tree.

In BGMP, the receiver domain is allowed to build *source-specific uni-directional* inter-domain distribution *branches*. However, such branches are not allowed to collide with the shared tree, for the sake of loop avoidance and possible introduction of duplicate packets. The need to construct such branches arises when the shortest path from the current domain to a source domain does not coincide with the bi-directional shared tree from the domain. This feature is very useful for domains running MIGPs, such as DVMRP and PIM-DM which support only source-based trees within the domain and only accept source traffic if it arrives from the shortest path back to the source (RPF check). The trick used by the ingress border router is to encapsulate the packets to the appropriate RPF-compliant border router, from where the packets can be injected into the domains' MIGP. If a source-specific branch is constructed, data is sourced into the domain via the appropriate border router avoiding the data encapsulation overhead. Source-specific branches differ from source-specific shortest path trees built by some MIGPs in that the source-specific branch stops where it reaches either a BGMP router on the bi-directional tree or the source domain. In shortest path trees, the source-specific state is set up all the way back to the source. It is also assumed that, since the inter-domain topology is sparser than the intra-domain topologies, the traffic concentration aspects related to the shared trees are not too much of a penalty for the protocol.

In order to ensure reliable control message transfer, BGMP runs over TCP but uses a different TCP port than BGP's. BGMP routers have TCP peering sessions with each other for the exchange of BGMP control messages (e.g., "Join" and "Prunes"). The BGMP peers for a certain group are determined via BGP. It is assumed also that BGP's route selection algorithm ensures that one border router, among the border routers of the domain, is chosen as the best exit router for each group route. This router has an external peer as its next hop toward the root domain of the group and the other border routers have the best exit router for each group route.

Data packets are forwarded in BGMP on a combination of BGMP and MIGP rules. Routers forward data packets to a set of targets according to a matching source-specific entry (*S*, *G*) if it exists. If not, a matching shared tree state entry for the group is checked. If neither is found, then the packet is sent natively to the next hop peer for *G* that is the best exit border router for the root domain according to BGP rules (this is the case for a non-member sender). If a matching entry was found the packet is forwarded to all other targets in the target list. If a target is a MIGP component, then forwarding is subject to the rules of the MIGP protocol.

Satisfying policy constraints for an autonomous system's (in this text also referred to as a routing domain) multicast traffic and considering heterogeneous routing domains can often translate into an increase of group state maintenance and delivery quality. BGMP goes around this problem by aligning multicast domains with autonomous systems and thus obtaining efficient policy support following the routes defined by BGP. Still, policy control is restricted to the policy constraint support of BGP's underlying hop-by-hop routing paradigm and path vector concept. This implies, for example, that network-specific policies cannot be supported. Furthermore, accepting traffic from "come-from" interfaces might not be discriminatory enough as a policy mechanism. This is because traffic barriers imposed by autonomous systems may be bypassed if a source is covered by a prefix that is homed to more than one domain [38].

Due to bi-directional forwarding, BGMP is not adequate for asymmetrical routing environments. Moreover, BGMP can only support source-specific delivery criteria in limited cases, for the sake of reducing the complexity of the protocol. BGMP has been designed with the aim of being able to be used in heterogeneous multicast routing domains and to be independent of the MIGP deployed intra-domain. Thus, for a globally available multicast routing solution, the use of BGMP implies solving interoperability problems specific to whichever MIGP is in use. This has not proved to be an easy task and, in same cases, encapsulations cannot be avoided. This is the case when the MIGP protocol is suitable for regional deployment but not for supporting multicast transit traffic (see also [39] for suggestions on interoperability between BGMP and the most currently deployed MIGPs).

Considering the above, it can be argued whether inter-domain multicast routing would not be better served with a unique routing protocol used intra-domain and inter-domain or an adaptation of an existing protocol that could then be applied both intra-domain and inter-domain. Examples of protocols that take such an approach are described below.

EXPRESS — The EXPRESS multicast protocol [40] is based on the EXPlicit REquested Single Source (EXPRESS) multicast model as opposed to the current IP Multicast model that allows any host to send to any IP multicast address at any time without prior notification (potentially any source, or PAN, multicast). The idea has been triggered by the observation that the PAN multicast model violates the current Internet service provider (ISP) charging model developed around unicast. An ISP charges for unicast based on the data rate from the customer into the ISP access point. This model implies that the ISP network dimensioning strongly depends on the assumption that a customer input rate of *R* imposes a delivery rate for the ISP of *R*, and not more than *R*. Multicast violates this property since, for a multicast group with *N* members, although there are delivery economies within the network, the packet eventually explodes out to *N* copies, imposing a delivery rate of approximately *N* * *R* on the ISP.

Another problem associated with PAN multicasting is that it appears to require a complex protocol in order to operate at large scale. This is induced by the fact that it is a PAN multicast requirement that the network layer be able to deliver a multicast packet sent by any host located anywhere in the Internet to all group members "without" any previous warning or notification from the source. The administrative costs associated with deployment aspects, such as configuring routers to interoperate with neighbor domains, are yet another disincentive for Internet multicast to be widely deployed.

Considering the above, Holbrook and Cheriton [40] proposed a multicast model that consists of defining multicast *channels* instead of multicast groups that are defined by a tuple (*S*, *E*) where *S* is the source address of the sender and *E* is an EXPRESS destination address. A host requests reception of data sent to a channel (*S*, *E*) by explicitly specifying both *S* and *E* to the network in a subscription request (e.g., using modified IGMP reports). To distinguish from the PAN multicast model we call the receiver host a *subscriber* instead of a member. The source *S* sends data to channel (*S*, *E*) by simply transmitting a datagram addressed to *E*. Only host *S* may send to (*S*, *E*).

The subscribe/unsubscribe process in EXPRESS is equivalent to join/leave in CBT, respectively. There are differences because the subscribe messages are propagated in the direction of the source, not the core, and a subscription entry at the on-tree routers contains the tuple (*S*, *E*) (the address *E* *per se* is insufficient to identify the multicast channel).

EXPRESS proposes the use of *authenticated* subscriptions. Thus, the host that wants to subscribe to a channel needs to know not only the addresses *S* and *E*, but also the channel key $K(_{S,E})$. This is different from CBT's join mechanism in the sense that the first subscribe message is propagated all the way to the source host, allowing the first hop router to check authorization against the source-supplied key. Subsequent subscribe messages stop at an on-tree router since the on-tree router has stored the key $K(_{S,E})$ for group (*S*, *E*) and can thus check the key supplied by the subscriber against the stored key.

As far as the host subscription protocol is concerned, EXPRESS needs a protocol that performs the function of IGMP. For this end, either

- IGMP could be extended in order to add the source address (and key for authenticated subscriptions).
- Simply use the same subscribe/unsubscribe mechanism that is used router-to-router also between a host and a router.

The EXPRESS model is not really interesting for multicast applications where sources may come and go during the life of the multicast session (e.g., distance learning applications). However, this kind of applications can be supported by EXPRESS using an application *session manager* (SM) that coordinates access to the session. The SM uses an EXPRESS channel (*SM*, *E*) to which each participant (source or receiver) subscribes. The role of the SM is similar to the role of the rendezvous point (RP) in PIM-SM, relaying traffic from multiple sources through a shared tree to the subscribers. However, because the SM has application-layer information, it can choose when to use a separate EXPRESS channel for sources that require a source-specific tree (high data rate sources) and when to relay via the SM's shared channel data from sources that send sporadically.

The EXPRESS model makes the multicast service more viable for an ISP in a number of ways:

- The network provider can provide assurance to a channel's owner that no other sources can send on the channel, and a content provider can assure that a subscriber can only receive traffic from a specific source on a certain channel.
- The EXPRESS model allows a router to determine the size of the downstream portion of the multicast tree expressed as the number of links. Link bandwidth is a costly resource for an ISP, and thus it can classify channels based on the number of links that are in use. The number of links in a tree provides the router nearest to the source (and all other routers) with the approximate size of the downstream multicast tree, giving the ISP the information necessary to charge the source of the channel based on its size.
- Since a multicast source is to be charged based on the size of the channel, it is necessary to provide the source with a mechanism to prevent unauthorized hosts from subscribing to its channel. Authentication subscriptions make it possible to do just this.
- The scarcity of multicast addresses is eliminated since each host has available the whole range of multicast addresses (2^{28} channels) that it can send to.

Simple Multicast — Simple Multicast (SM) [41] was inspired by the idea explored in EXPRESS of identifying a multicast group not just by its class D address but also by a unicast IP address (8 bytes altogether). However, SM defends the use of a single shared bi-directional tree rather than source-specific trees, and thus insists on the idea of a core router through whom all receivers join the multicast group.

Simple Multicast is based on two main ideas:

- SM identifies a group address with 8 bytes that comprises the core IP address, *C*, and the class D address of the multicast group, *G* (C, G). The extension of the size of the group identification address makes address allocation a trivial task. In fact, each distinct core could administer a complete class D space.
- An end-node (be it a host or a router) conveys the (*C*, *G*) parameters to an SM router, eliminating the need for a mechanism to advertise the set of cores, which is not scalable for a large domain and potentially suffers from slow convergence.

Simple Multicast features:

- Separation of multicast group and core address allocation from routing.
- Unification of intra-domain and inter-domain routing protocols in order to avoid complex interoperability issues between several (conceptually different) multicast routing protocols.
- Expansion of usable address space to avoid having to care for multicast address clashes and make possible the use of hierarchical address space for filters.
- Exploitation of state reduction for groups in which the majority of the sources are also members.
- Backward compatibility mechanism with existing multicast protocols to allow for incremental deployment. (See [41] for details on the tunnels proposed by SM.)

SM shares the first three features with EXPRESS; the last two are specific to SM. The adoption of bi-directional instead of uni-directional shared trees *minimizes the influence of core placement*, since traffic from a source distant from the core does not need to be sent forth to the core and then to the set of receivers. In fact, as soon as the traffic sent by the source meets an on-tree router, it can be forwarded on all interfaces that are on the tree (except the interface from where the data came). Thus, traffic is received directly from a source if the receiver can be reached in a branch that is downstream from the "fan-out" on-tree router. Traffic will be sent via the core for the remaining receivers.

Simple Multicast eliminates the domain-level control problem (also referred to as the third-party independence problem). The idea behind this is that if SM is used both intra-domain and inter-domain, "Joins" from different parts of the domain might only converge outside the domain. However, it is not desirable for a domain to depend on another "third-party" domain for the distribution of internally sourced traffic to other internal receivers. It is therefore necessary to ensure that "Joins" from different internal receivers merge at a common point inside the domain. Since BGP-4 allows the egress/exit router from a domain to be specified for a particular route (or unicast prefix), "Joins" inside a domain can converge at the desired common point inside the domain.

Shared trees are not as flexible as source-based trees to support transit routing policies. This is because packets from a domain *A* to receivers in another domain *B* might have to be sent via third domain *C* if the core resides inside domain *C*. It can happen also that policy might prohibit packets from domain *A* to domain *B* to transit domain *C*. Simple Multicast proposes that in such a case the sender *S* in domain *A* sends a message to the core *C* announcing the creation of another group rooted at the source (*S*, *G*). The core in domain *C* will send the message to all receivers, which will prompt them to also join the new group (*S*, *G*). SM creates additional trees only in such cases when transit policies prevent the source to reach group receivers.

Simple Multicast is not "yet another protocol." In fact, it builds on the ideas of CBT for tree construction and maintenance and on the ideas of EXPRESS multicast to use the 8-byte addressing scheme. Because of its simplicity and scalability features, SM is suited to intra-domain as well as inter-domain multicast routing. An additional feature of SM's design is to allow the support of data-driven dense-mode multicast distribution based on a simple Reverse Path Multicast Forwarding algorithm (as used in PIM-DM) for groups with a very dense set of (local) members. It does so by inserting a core address of 0xFF:FF:FF:FF in the SM header of the data packet. More details about the protocol can be found in the IETF draft by Perlman *et al.*, 1999.

The following protocols are the first proposals for inter-domain multicast routing protocols. They consist of the application of a hierarchical routing model to existing multicast protocol proposals (e.g., DVMRP, PIM-SM and CBT). Of the proposals listed below, only HIP will be listed in Table 4 since it is similar to BGMP in its goals.

HDVMRP — Hierarchical DVMRP [42] has been proposed as a routing protocol that interconnects domains running any of the existing MIGP. HDVMRP floods data packets to the boundary routers of all regions, and boundary routers that are not part of the group send prunes toward the source network to stop receiving packets. This implies a large amount of overhead and, as in DVMRP, maintenance of state per source, even where there is no interest for the group. HDVMRP also requires encapsulating data packets for them to transit a domain, which adds additional undesirable overhead.

HPIM — Hierarchical PIM [43] builds on PIM-SM using a hierarchy of RPs for a group. A receiver would send "Joins" to the lowest level RP, which in turn would "Join" an RP at the next level, and so on to top of the hierarchy. The number of levels of the hierarchy is related to the scope of the multicast group addresses. Data flows in a bi-directional manner along the tree of the RPs (that is, the tree above the lowest level of the hierarchy). However, since HPIM uses hash functions to choose the next RP at each level, the tree does not perform well in terms of delays from source to receivers, especially in the case of local groups.

OCBT — The Ordered CBT [44] protocol is an extension to CBT that uses a hierarchy of cores in the same way as HPIM. OCBT functions similarly to CBT in the sense that a router wishing to join the tree sends a "Join" request toward the core, which is followed by a "Join" acknowledgement either by the core itself or an on-tree node. In OCBT, every core and on-tree router also maintains an integer-logical level. The level of each core is fixed and the level of the router is set by the returned "Join" acknowledgement, which is marked with the level of the core or on-tree router that was reached. When a lower level core receives a "Join" request and it is not already part of the multicast tree, it must "Join" to a higher level core and does so in the same way by sending a "Join" request toward the next highest core. The "Join" request is marked with the level of the core for which it is intended; if it reaches a branch of the tree of that level or higher, then the "Join" acknowledgement is marked with that level and builds a branch of that level back to the sender. If the request reaches a lower level branch, that branch breaks to allow formation of the higher-level branch. It is this labeling and breaking mechanism that ensures that OCBT remains loop free.

One of the criticisms associated with HPIM and OCBT is that they do not allow for an arbitrary MIGP to run inside a domain. They are dependent on a specific multicast routing protocol to be deployed intra-domain. In the case of HPIM, the MIGP is PIM-SM and, for OCBT, it is CBT. Furthermore, both OCBT and HPIM suffer in hierarchical application from the fact that it is very difficult to come up with a hierarchical placement of cores or RPs without extensive knowledge of the network topology and the receiver set. HPIM and OCBT do not support policy routing, either.

HIP — The Hierarchical Multicast Routing [45] protocol was the first protocol to propose the construction of a shared tree of domains. The goal is, then, to build a single distribution tree across all receivers in a way that the control information regarding the tree is limited to a particular domain or level. This type of tree can be defined as a "tree within trees," that is, a node of a higher-level tree actually contains a lower-level tree. In this type of hierarchical scheme, a single flat routing region is divided into several non-overlapping domains, each of which runs its own intra-domain multicast routing protocol. HIP uses OCBT as the inter-domain routing protocol in a hierarchy that can include any multicast routing protocol at the lowest level. Thus, it allows for heterogeneity of multicast protocols at the different levels.

HIP comes up as an answer to the problem, in OCBT and HPIM, to define each logical level of cores as an actual level of a hierarchy. HIP constructs a tree of trees introducing the idea of a *virtual router* (VR) that is formed by all border routers of a domain operating in concert to appear as a single router in the next level of the hierarchy.

HIP is similar to BGMP [39] in its goals; it differs in the way it accomplishes these goals. First, HIP allows multiple hierarchical levels, instead of the two levels implicit in BGMP, in order to adapt for scalability. Second, HIP defines methods for distributing the location of the center point, though these are not mandatory in the function of the protocol (MASC [36] can be used instead). The default location of the center point is the domain that contains the group creator. While this does not guarantee a good location with respect to the receiver set, it performs as well as BGMP, which in fact uses the same heuristic for center point location. Third, in contrast to BGMP, HIP does not allow per source branches. The idea behind this is that domains that choose to use RPF routing per source should not be allowed to pass the cost of their routing to domains in higher levels of the tree. Finally, HIP is easily extensible to provide security services for the multicast routing (see the recently proposed protocol in [46]).

In spite of the brilliant idea to improve the scalability of multicast routing in a large network, HIP suffers from the same criticisms as those of BGMP and all the previously described hierarchical approaches. This is because the mechanisms for center point location at each tree level are not related to the distribution of the receivers nor the network topology. This results in an increase of the packet delay between the source and the receivers when the receivers are not located in the same domain as the center point.

Other approaches to multicast routing more focused on a certain application (e.g., mobile hosts, very sparse groups) are described in the following sections.

Centralized Multicast — Centralized Multicast [47] is an approach that proposes to separate the forwarding of multicast packets from control operations, namely routing, resource reservation, and group management. In each domain there is a control element, the gateway, and control elements linking gateways, called root controllers. Because of the centralized control in this approach, it has been called Centralized Multicast.

The problems related to centralization, namely, control overhead from central participation in all decision making and failures due to non-response of the control elements, are addressed by:

- Creating a hierarchy of gateways and root controllers.
- Replication of gateways.
- Loose synchronization among root controllers.

Centralized Multicast is prone to long "Join" delays, in particular when the group that the receiver wants to join does not have a tree branch in the same domain. This is because "Joins" are directed to the gateway, which in turn will proceed to select the closest on-tree node and the nodes of the new branch to whom messages should be directed, stating the type of state entry to be added to the "forwarding" table.

Static Multicast — Static Multicast [48] is a proposal that addresses the scalability of sparse mode multicast routing protocols regarding the mechanism to advertise RPs or cores (in PIM-SM and CBT, respectively). The proposal consists of using DNS to find RPs or cores for a certain group given DNS's scalable and hierarchical mechanism that enables dynamic updating and security awareness. Static Multicast has been submitted to the IETF as a proposal for modifications of PIM-SM for Static Multicast [49].

DCM — Distributed Core Multicast [50] is a protocol proposed in the context of a large single domain with a very large number of multicast groups with small numbers of receivers. Such a case occurs, for example, when multicast addresses are statically allocated to mobile hosts, as a mechanism to manage Internet host mobility.

The DCM protocol is based on an extension of center-based tree protocols such as PIM-SM or CBt. It uses several core routers, called distributed core routers (DCRs) and a special control protocol among them. The most interesting feature of this approach is that it suggests how to:

- Avoid multicast group state information in backbone routers.
- Avoid triangular routing across expensive backbone links.

Basically these can be achieved either with "tree-based source routing" or "list-based source routing." Please refer to [51] for more details.

Connectionless Multicast — Connectionless Multicast [52] (CLM) is an approach directed to multicasting for groups with a small number of receivers (and sources), as is likely to be the case inter-domain. Multicast address allocation is not required since multicast D-addresses are not used. CLM encodes the list of member addresses in the data packets.

CLM does not require state to be maintained at routers, nor does it suffer from sub-optimal forwarding in the case of asymmetric routing conditions. This is because the path is constructed from the source to the receiver and not the other way around.

CLM enables fast reaction to topology changes since the next hop to a destination is determined solely via the unicast routing tables, that is, is not based on multicast state.

CLM eases security and accounting since the sources will know all receivers before forwarding data to them. Moreover, it is possible to control the identity of the senders via an out-of-band mechanism.

The price for all these advantages is an increased packet overhead and more complex header processing. However, the latter problems can be alleviated by a caching mechanism and by header compression. CLM was initially designed to be used as an inter-domain protocol in combination with Simple Multicast or PIM-SM intra-domain. CLM can also be used end-to-end for a limited number of receivers.

In the area of *multicast addressing*, and apart from the MASC architecture [36], EXPRESS [40], and Simple Multicast [41], with their proposal for 8-byte group address, other proposals exist, such as:

- Pejhan *et al.* [53] have proposed group addresses based on the IP address of the host and the port number of the application on the host that initiates the group. The resulting group address is six bytes long. Unless an incremental solution is used to support this, all routers have to be changed to recognize the extended addresses.
- Braudes and Zabele have outlined a hierarchical address allocation scheme in [54]. It consists of a query-reply response mechanism with a single root for the hierarchy.
- Levine and Garcia-Luna-Aceves proposed the Addressable Internet Multicast (AIM) architecture [55] which generalizes the IP Multicast architecture by introducing group-relative addressing information in multicast routing trees. This added information makes possible the provision of new sender- and receiver-initiated delivery services and allows higher-layer protocols to place packets into application-defined logical streams, so that hosts may prune the routing packets based on contexts meaningful to the applications.

DISCUSSION

The multicast routing protocols presented here are classified according to the parameters identified in Table 2 and the results are shown in Tables 3 and 4. These 13 multicast routing protocols have been selected because they have been implemented or are interesting to be considered for implementation or because they have been subject to performance studies (see also the studies done by Wei and Estrin [9] and Bilhartz *et al.* [23]). Performance studies are a scarce resource to obtain in the area of multicast routing protocols, because most of these protocols have not been implemented. Simulation studies are another possibility to test the performance of the protocol, but a widely accepted benchmark against which to test the protocol is not yet accepted by the research community.

In the following, a discussion will be presented on specific protocol taxonomic features that are particularly important from a network or application perspective. Wherever suitable, pointers to crucial requirements for the deployment of specific multicast applications will be given.

Scalability — The ability of a protocol to scale can be evaluated in terms of the overhead growth in the presence of a large number of groups or number of participants per group and groups for which the set of participants (receivers and senders) changes often over time. The multicast application for which scalability is crucial is distributed interactive simulation (DIS). DIS applications are characterized by hundreds of changes per second in the set of participants and by a large number of participants.

Overhead can be measured in terms of memory resources (in routers) as they relate to routing state entries maintained per group; bandwidth resources can be measured in terms of control or signaling messages per group and processing power. Mechanisms used to minimize *memory resources* include:

- Minimizing the routing state entries maintained at the node. For example, in DVMRP, state entries are aggregated per source network rather than one state per source.
- Partitioning the routing domain in "areas" and promoting the use of a hierarchy. For example, in MOSPF a two-level hierarchy is used to manage routing areas with a limited number of routers per area.
- Using a shared tree rather than source-specific trees for each of the active sources. For example, CBT maintains only one state entry per group rather than *N* (times the number of groups), where *N* is the number of active sources.
- Delegating the responsibility of managing group dynamics to an intermediate node, e.g., the RP in PIM-SM or core in CBT).
- Creating routing state for a group "on-demand," that is, only when the first packet from a source is emitted.

Mechanisms used to minimize *control message overhead* include:

- Pruning multicast tree branches whenever there are no members downstream.
- Using a receiver-driven approach to advert the join/leave of a participant of the group (as in PIM-SM and CBT) rather than flooding (as in MOSPF).
- Use "hard" states rather than "soft" states. Soft states need to be refreshed after a certain period by reception of a control message, which creates overhead, in particular when the number of receivers is large and sparsely distributed over the routing domain. Hard states do not need to be refreshed once they have been created.

In terms of *computational complexity* it is easy to identify some protocol design features that make the protocol suitable to be classified as complex. For example, MOSPF is a computationally intensive protocol because the multicast tree for a specific source and group must be recalculated at every router every time a new receiver joins or leaves the group. This is

FEATURES	DVMRP	MOSPF	CBT	PIM-DM	PIM-SM	MIP
Independent of unicast protocol	No (depends on RIP)	No (depends on OSPF)	Yes	Yes	Yes	Yes
RPF-based	Yes	Yes	No	Yes	Yes	No (shared tree)
Uni/bi-direct. trees	Uni-directional	Uni-directional	Bi-directional	Uni-directional	Bi-directional	Both
Multicast tree types	Source-specific trees	Source-specific trees	Shared tree	Source-specific trees	Shared and source-specific trees	Shared and source-specific trees
Multicast routing algorithm	Flood and Prune with RPF check	Dijkstra's algorithm	Shortest Path Tree	Flood and Prune with RPF Check	Shortest Path Tree	Shortest Path Tree
Core select. method	Not applicable	Not applicable	Out-of-band mechanism	Not applicable	Bootstrapping	On-tree node
Loop free	Transient loops can occur (during routing table updates)	Transient loops can occur (during routing table updates)	Loops can occur if more than one core is used (CBTv1)	Transient loops can occur (during routing table updates)	Transient loops can occur (during routing table updates)	Loop-free (diffusion algorithm)
Third-party dependent	No	No	Yes	No	Yes	Yes
QoS aware	No	Yes (TOS trees)	No	No	No	No
Security	No	No	Yes	No	Yes	Yes
Incremental deployment	Yes	No	No	No	No	No
Development stage	Deployed according to standard	Deployed	Not deployed	Deployed according to standard	Deployed	Research only
Idea brought forth	DVMRP tunnels	Two-level hierarchy	Center-based trees	Unicast Protocol Independence	Support of both shared and source-based trees	Receiver or source-initiated tree construction
Relevant assumptions	Symmetric routes	All routers are multicast-aware	Symmetric routes	Symmetric routes	Symmetric routes	Symmetric routes
Group management	Slow: flooding	Slow: flooding	Acceptable: core-based	Slow: flooding	Acceptable: core-based	Acceptable: distributed
Computational complexity	Acceptable	Complex	Acceptable	Acceptable	Complex	Complex
Latency	Small end-to-end; slow join	Small end-to-end; slow join	Max. 2x DVMRP's end-to-end delay; small join	Small end-to-end; slow join	Small end-to-end; small join	Small end-to-end; long join
Traffic concentration on links	No	No	Yes	No	No	No
Control message overhead	Heavy (refresh prune state)	Heavy (increases with frequency in membership changes)	Light (refresh only shared tree state entries)	Heavy (refresh prune state)	Heavy (refresh all state entries)	Heavy (diffusion query-reply)
Memory consumption	State entry per pair of *S* and *G* in all routers of the domain	State entry per pair of *S* and *G* in on-tree routers only	State entry per *G* in on-tree routers only; (*S*, *G*) state is built occasionally between the border router and the core (CBTv3)	State entry per pair of *S* and *G* in all routers of the domain	State entry per *G* and state entry per pair *S* and *G*, for certain *S*, in on-tree routers	Same as PIM-SM, but MIP maintains hard state rather than soft state entries
Scalability	No	No	Yes	No	Yes	Yes
Easy to implement	Yes	No	Yes	Yes	No	Yes
IP mobility	No	No	See DCM [47] for suggestions	No	See DCM [50] for suggestions	No
IP over ATM	Complex	Complex	Fits well with ATM forwarding scheme	Complex	Fits well with ATM forwarding scheme	—

Table 3. *Intra-domain IP Multicast routing protocols.*

computationally heavy for the router, particularly if the source remains active for short periods of time, the number of sources or the number of groups is high, or some of the groups have high group dynamics. Another example of a complex protocol is PIM-SM, which uses a system of timers to refresh routing state entries. The refreshing period needs to be carefully configured in order to be able to detect changes in either the routing topology (e.g., RP failures) or the membership of the group. Hence, long refreshing periods make the protocol insensitive to changes in the multicast routing environment, causing transient routing loops to occur during undetected failures or updates in unicast routing tables. Small refreshing periods cause a large transmission overhead to be experienced since many control messages must be transmitted and processed. In PIM-SM, the switch-over from a shared tree to source-specific trees is also a complex process that requires measurements to be performed at routers close to receivers.

Robustness — Robustness is there to ensure that the protocol is resistant to the formation of routing loops or the injection of duplicate packets in the presence of changes in the routing tables and in the membership of the group. Robustness also relates to the ability of the protocol to limit the influence of network topology failures to the neighboring nodes where the failure took place. Robustness in the presence of updates in routing topology and changes in the set of participants is achieved by:

- Promoting the use of *soft-states* (periodically refreshed states), as in PIM-SM, rather then hard routing state entries.
- Periodic flooding with RPF check. This will prevent the existence of cyclic routes, but duplication of packets may still occur during unicast routing table updates.
- Construction of an *identical multicast tree* at every node, that is, being able to always choose the same path at every node in the presence of equal-cost paths (e.g., in MOSPF).
- Using a *diffusion mechanism*, as proposed in MIP, for updating the multicast tree when a new member joins/leaves. Diffusion makes the multicast protocol independent of unicast routing tables since all possible paths are considered.
- For protocols that use a core or center point (such as PIM and CBT), the use of a *pre-configured set of cores*, rather than a single core, to lower the effects in case of core failure.

Latency — Join latency and end-to-end latency can be influenced by the design choices of the multicast routing protocol. The following are some examples of how latencies can be minimized:

- Use of *source-specific trees* rather than shared trees when the sources are high data rate sources (as in PIM-SM and MIP), because a source-specific tree offers the shortest path from a source to a receiver. In addition, multiple (source specific) trees are better at balancing the traffic load of several high data rate sources among the nodes of the domain than a single shared tree.
- *Select the core* among the participants of the group (as in MIP), based on the heuristic that if there is a group participant in a certain location it is very likely that other participants also exist (or will join) for that location.
- Use of *bi-directional trees* in protocols using a single shared-multicast tree per group (as in CBT and BGMP). Bi-directional trees minimize end-to-end latencies for receivers that are in the close neighborhood of sources since the data does not have to go to the core prior to being transmitted to the receivers. (Note that the "Join" latency is still dependent on optimal core placement since receivers will join via the core.)

Latencies are to be avoided for interactive applications that involve a two-way communication such as audio and video conferencing. Other applications, such as the delivery of stock information (Web based) or participation in on-line auctions are also delay-intolerant. On the other hand, streaming applications such as headline news, weather updates and sports scores are tolerant of end-to-end transmission delay.

Constrained Routing — The support of constrained routing involves considering either:

- Policy constraints that are domain-specific (policy routing).
- Constraints on loss, delay, jitter or other dynamic metric (QoS routing).

In IP Multicast, the only protocols that support constrained routing are YAM, QoSMIC and PTMR. However, there is no consensus on the Internet Engineering Task Force (IETF) regarding unicast- or multicast-constrained routing. This partly explains why multicast-constrained routing is a research topic, since the multicast tree is constructed based on the unicast routing tables.

The current solution for constrained unicast routing calls for a reservation of bandwidth for the unicast path via the Resource Reservation Protocol (RSVP). RSVP [56] is run by the end nodes and can traverse nodes that do not run RSVP. Thus, the service guarantees provided by RSVP are only as good as the weakest link in the path. RSVP does not scale inter-domain due to the high control message overhead inherent in the protocol. Thus, combined solutions using RSVP intra-domain and DiffServ inter-domain are currently proposed [57].

Issues related to constrained routing specific to IP Multicast routing involve adapting for heterogeneous receivers, that is, receivers with different service requirements in terms of, for example, delay or jitter. This involves either:

- The maintenance of different multicast trees associated with a certain type of service (as in MOSPF).
- A way of arbitrating which is the service requirement that prevails if different service requirements exist downstream from a certain tree branch (e.g., a receiver is happy with a "best-effort" type of service, whereas another specifies constraints on delay).

Constrained routing requires the use of dynamic routing metrics to track changes in values for time-dependent metrics (e.g., delay, jitter). If a hop-by-hop routing paradigm is to be followed (as is the case in IP networks) then an efficient way of advertising dynamic routing metrics to all the nodes in the domain is required. Link-state routing protocols such as OSPF can be used intra-domain, but such protocols do not scale inter-domain. MBGP [33] can be used inter-domain to advertise routing policies. However, it is not foreseen that dynamic routing metrics will be advertised inter-domain. This would simply not scale due to the high control overhead.

Another issue to take into account is that most of the currently proposed multicast routing protocols are based on Reverse Path Forwarding (RPF). RPF is based on the idea that the path from A to B is the same as the path from B to A (symmetrical paths). However, when routing constraints are introduced, there is no guarantee that this is the case. Hence, RPF will cause forwarding on a sub-optimal path (in QoS routing) or might even prevent receivers from receiving traffic from certain (or all) sources (in policy routing).

Policy routing is mandatory in inter-domain multicast rout-

FEATURES	QoSMIC	PTMR	MSDP/PIM-SM	BGMP	EXPRESS	SIMPLE M.	HIP
Independent of intra-domain multicast rout-. ing protocol	Yes (applied intra- and interdomain)	No (PIM-SM or CBT)	No (PIM-SM)	Yes	Yes (applied intra- and interdomains)	Yes (applied intra- and interdomains)	Yes
Interoperable with intra-domain multic. rout. protocol	No	No	No, only with PIM-SM	Yes	No	No	Yes
RPF-based	No	No	Yes	Yes	Yes	No	No
Uni/bi-direct. trees	Uni-directional	Uni-directional	Uni-directional	Bi-directional	Uni-directional	Bi-directional	Bi-directional
Multicast tree types	Both (used as in PIM-SM)	Both (as in PIM-SM)	Both	Shared-tree and source-specific branches	Source-specific trees	Shared tree	Shared tree
Multicast rout-ing algorithm	One-to-many join mechanism	Source routing	BGP-advertised shortest path	BGP-advertised shortest path	BGP-advertised shortest path	BGP-advertised shortest path	BGP-advertised shortest path
Core selection method	Designated router of first receiver	As in PIM-SM	As in PIM-SM	Root domain for *G* address (def. By MASC)	Not applicable	Out-of-band mechanism	Domain of group initiator
Loop-free	Yes	No, when tree branches merge	Yes	Yes	Yes	Yes	Yes
Third-party dependent	No	No	No	Yes (root domain)	No	Yes	Yes
QoS/policy aware	Yes	Yes	No	No	No	No	No
Security	—	—	Yes	Yes	Yes	Yes	Yes (KHIP)
Incremental deployment	No	Yes?	No	No	No	Yes, SM tunnels	No
Development stage	Research only	Research only	Soon to be deployed	Standardization process	Research only	Research only	Research only
Idea brought forth	Choice of alternate paths to join a tree	"Come-from" routing rather than "go-to" routing	Source address advertisement inter-domain	Shared tree of domains	8-byte address composed of source and group address	8-byte address composed of core and group address	Virtual router (trees within trees)
Relevant assumptions	Use of dynamic metrics	Asymmetric routes exist	Most of the sources are in same domain as receivers	Most of the sources are local to the root domain	Groups have a few sources	Most sources can also be receivers	Source-specific trees are not scalable
Membership management	Slow (via manager node)	Acceptable (same as PIM-SM's)	Acceptable for receivers; slow for new sources	Acceptable: only root domain knows source domain	Acceptable for receivers; slow for new sources	Acceptable: only core knows sources	Acceptable: only core knows sources
Computational complexity	Complex	Complex	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable
Latency	Small end-to-end; slow join	Small end-to-end; slow join	Small end-to-end; slow join	Small join; ?end-to-end	Small end-to-end; small Join	Small join; ?end-to-end	Small join; ?end-to-end
Traffic concentration on links	Acceptable	Acceptable	Acceptable	Heavy?	Acceptable	Heavy?	Heavy
Control mes-sage overhead	Heavy (join dis-covery process)	Heavy (marking cycles)	Heavy (soft state entries)	Light (hard state entries)	Light (hard state?)	Light (hard state entries)	Light
Memory requirements	Same as PIM-SM	Same as PIM-SM	State entry per pair of *S* and *G*	State entry per *G* and occasionally source-specific	State entry per pair of *S* and *G*	State entry per *G* address	State entry per *G* address

Table 4. *Inter-domain multicast routing protocols (continued on next page).*

FEATURES	QoSMIC	PTMR	MSDP/PIM-SM	BGMP	EXPRESS	SIMPLE M.	HIP
Scalability	No	No	No	Yes (?)	No	Yes	Yes
Easy to implement	No, uses dynamic metrics	No, needs changes in PIM-SM	Yes	No, needs MASC	No, host model changes	No, host model changes	No

Table 4 (cont.). *Inter-domain multicast routing protocols.*

ing. MSDP and BGMP focus on MBGP-advertised paths, but do not provide a clear solution in the presence of routing asymmetries. This might introduce an increase in the end-to-end delay that could have been avoided if an alternate route had been considered. PTMR proposes a solution based on source originating paths, that is, it is the router closer to the source that "marks" the path between itself and the targeted receiver. This enables a routing mechanism that does not consider how the discovered path has been established and what route selection criteria were involved. Moreover, source originating tree deployment is independent of the underlying policy model.

The support for constrained routing is crucial for multimedia applications, because these types of applications are particularly sensitive to a component of delay, the delay jitter, that tends to occur when traffic offered to a network is very bursty. The applications themselves need to be able to adapt to changing network conditions (see [58] for suggestions on methods for adaptive applications) but the network will need to provide a different type of service for applications where video and audio quality are essential for the communication (e.g., video conferencing).

Third-Party Dependent — Third-party dependence is to be reliant on a specific domain (e.g., ISP) to forward multicast traffic to the group. This issue is crucial in inter-domain multicast routing. In the intra-domain case, the choice of the root of the multicast tree is related to load sharing among the nodes of the domain, and thus the penalty on non-optimally located cores is not significant. The same cannot be said in the inter-domain case, that is, all possible root domains cannot be treated as eligible candidates. In inter-domain routing there are administrative and policy issues concerning the "ownership" of the root domain and a greater risk from poor end-to-end delay performance due to the location of the root.

The issue above is related to the fact that protocols, e.g., BGMP, address scalability by constructing a shared tree of domains rooted in a "root" domain. If, due to policy reasons, traffic from a certain domain A is not authorized to transit via the root domain, then a tunnel is required to reach another domain B. Otherwise, group members in domain B will never receive traffic sourced in domain A. BGMP addresses third-party dependency using bi-directional multicast trees that do not insist that traffic always circulate via the transit (root) domain. However, the problem is not clearly solved using bi-directional trees since there will still be some receivers that will be reached via the possibly non-authorized root domain.

Deployment Considerations — IP Multicast has just started to be deployed by big ISPs and in corporate networks. Ubiquitous deployment is the next challenge since at present a source in a domain cannot reach receivers in another domain without using tunnels. MSDP is the first solution for inter-domain multicast routing, followed by the MASC/BGMP architecture. Both of these proposals are in conformance with the IP service model proposed initially by Deering [1].

Approaches for multicast routing such as Simple Multicast and EXPRESS will need to provide for incremental deployment if they are to be deployed, since they imply changes in hosts. This is so because IGMP does not currently support multicast addresses 8 bytes long. However, and considering the simplicity in multicast address allocation obtained with SM and EXPRESS, there will be advantages in revising the current multicast model.

The ease of implementation plays a big role in the adoption of a multicast protocol these days. There are business factors that cannot be ignored and that have been manifested by the IP Multicast Forum [59]. There is also great interest in the research and engineering community to promote the deployment of multicast technology, which is vital for multimedia multi-party communications on the Internet. Constrained routing will be the object of increased interest once the technology has been ubiquitously deployed. Other outstanding issues crucial for multicast applications are security (see also ([18, 46, 60]) and reliable multicast ([61]). It is expected that more than one solution, each specialized to address particular application requirements, will emerge either in the case of secure or reliable multicast. Network issues that need to be further addressed are policy routing (see also [28, 33]), address allocation ([36, 41]), and network management tools ([62]) for IP multicast.

CONCLUSIONS

IP Multicast is particularly well suited for applications that send the same data to more than one receiver, that is, for applications that involve multi-party communication. Examples of these applications are: video-conferencing, shared workspace, distributed interactive simulation (DIS), software upgrading, etc. The reason for this is that IP Multicast puts the strain on the network rather than on the applications. It is up to the network to route the multicast traffic to the participants in the multi-party communication; for the application it implies no distinct effort to send the information to 10 or to 10,000 receivers. The application only has to send one copy of the data, rather than as many as the number of receivers.

In addition, IP Multicast makes it possible to lower the network load since, at each router, only one copy of an incoming multicast packet is sent per outgoing link, rather than sending one copy of the packet per number of receivers accessed via that link. However, IP Multicast traffic is not network-friendly in the sense that multicast packets are sent over UDP, and thus the source of packets has no feedback in case of congestion, as is the case with TCP traffic. This implies that network performance can be severely affected by a non-robust IP Multicast routing protocol: duplicate or looping packets can generate large amounts of traffic, which may eventually lead to network congestion. These negative effects become more acute as group size grows or group dynamics increase.

Multicast routing protocol design choices need to be pursued in order to provide:

- A simple but robust algorithm for tree construction.
- An efficient control mechanism to detect changes in the set of participants and trigger the appropriate route update actions.

Many protocols have been proposed for multicast routing. This article surveyed more than 13 intra-domain and inter-domain multicast routing protocols. This is not presented to be an exhaustive listing of all proposed protocols, since the area is expanding constantly. Thus, it was useful to present a taxonomy to classify the features, successful design choices, and context of application of the proposed multicast protocols. The taxonomy presented in this article focuses not only on design choices to support scalable, robust and efficient routing, but also on network performance features such as control message overhead, memory consumption, computation complexity, end-to-end and "Join" latencies, and traffic concentration.

IP Multicast is about to be re-born out of the ashes. In 1988, Deering's work was aiming at resource discovery applications. Today, video and audio streaming and conferencing applications are approaching the technology as a means of reaching a larger consumer database. IP Multicast is already available from a few ISPs, and an inter-domain solution will soon follow. Concentrated efforts are being directed toward the ubiquitous deployment of IP Multicast (see IPMI [63] and Internet2 [64]). It is expected that management tools for IP Multicast networks will emerge as soon as multicast traffic starts to increase in ISP and backbone networks. There are still a few issues that require further research, including multicast address management, constrained routing, and reliable and secure multicast, but the horizon for the use of IP Multicast is starting to become clear.

Acknowledgments

Thanks to B. Sales, H. De Neve, and D. Ooms for manifesting the need for this document and making comments on the written version. The section on multicast routing algorithms has been inspired by Prof. Dr. ir. P. Van Mieghem of Delft University of Technology; the expertise on IP over ATM is based on the work of P. Dumortier. Special acknowledgments are also due T. Ballardie and Liwen Wu for feedback on the Simple Multicast section; A. Banerjea for feedback regarding QoSMIC; and C. Shields on HIP and OCBT.

References

[1] S. Deering, "Multicast Routing in Internetworks and Extended LANs," *SIGCOMM '88*, Stanford, CA, Aug. 1988, pp. 55–64.

[2] W. Fenner, "Internet Group Management Protocol, IGMP, version 2," *Xerox PARC*, IETF's Appendix I of RFC1112.

[3] C. Diot, W. Dabbous, and J. Crowcroft, "Multi-point Communication: A Survey of Protocols, Functions and Mechanisms," *IEEE JSAC Journal*, vol. 15, no. 3, 1997, pp. 277–90.

[4] E. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerical Mathematics*, 1959.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, McGraw-Hill, 1995.

[6] L. Kou, G. Markowshy, and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica 15*, pp. 141–5, 1981.

[7] D. Wall, "Mechanisms for Broadcast and Selective Broadcast," PhD thesis, Stanford University, Technical Report no. 190, 1980.

[8] P. Winter, "Steiner Problem in Networks: A Survey," *Networks*, vol. 17, no. 2, 1987, pp. 129–67.

[9] L. Wei and D. Estrin, "The Trade-Offs of Multicast Trees and Algorithms," *Int'l. Conf. Comp. Commun. and Networks*," Sept. 12–14, 1994.

[10] P. Dumortier *et al.*, "IP Multicast Shortcut over ATM: A Winner Combination," *Globecom '99*, Sydney, Australia.

[11] T. Pusateri, "Distance Vector Multicast Routing Protocol," IETF Draft, update to RFC 1075, draft-ietf-idmr-dvmrp-v3-06.txt, 1998.

[12] Dalal and Metcalf, "Reverse Path Forwarding of Broadcast Packets," *Communications of the ACM*, Dec., 1978.

[13] J. Moy, "Multicast Extensions to OSPF (MOSPF)," Proteon Inc., RFC 1584, 1994.

[14] J. Moy, "OSPF, version 2," Proteon Inc., RFC 1583, 1994.

[15] A. J. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT)," *Proc. ACM SIGCOMM '93*, San Francisco, CA, 1993.

[16] C. Shields and J. J. Garcia-Luna-Aceves, "The Ordered Core Based Tree Protocol," *Proc. INFOCOM 1997*, Kobe, Japan, 1997.

[17] A. J. Ballardie, "Core Based Trees (CBT, v2) Multicast Routing: Protocol Specification," IETF Draft, work in progress, 1997.

[18] A. Ballardie and C. Crowcroft, "Multicast Specific Security Threats and Counter-Measures," *Proc. Symp. Network and Distributed Systems (NDSS '95)*, Feb. 1995.

[19] A. J. Ballardie, "A New Approach to Multicast Communication in a Datagram Inter-Network," PhD. dissertation, University College of London, 1995.

[20] K. Calvert, E. W. Zegura, and M. J. Donahoo, "Core Selection Methods for Multicast Routing," *ICCCN '95*, Las Vegas, Nevada, 1995.

[21] S. Deering *et al.*, "Protocol Independent Multicast (PIM), Dense Mode Protocol: Specification," IETF Draft, work in progress, 1994.

[22] S. Deering *et al.*, "Protocol Independent Multicast (PIM), Sparse Mode Protocol: Specification," IETF Draft, work in progress, 1995.

[23] T. Bilhartz *et al.*, "Performance of Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols," *IEEE JSAC*, vol. 15, no. 3, 1997.

[24] M. Parsa and J. J. Garcia-Luna-Aceves, "A Protocol for Scaleable Loop-Free Multicast Routing," *IEEE JSAC*, vol. 15 no. 3, pp. 316–31, 1997.

[26] E. Dijkstra and C. Scholten, "Termination Detection for Diffusing Computations," *Inform. Process. Letters*, no. 11, vol. 1, pp. 1–4, 1980.

[26] J. Garcia-Luna-Aceves, "Loop-Free Routing Using Diffusing Computations," *IEEE/ACM Trans. Net.*, no. 1, vol 1, pp. 130–41, 1993.

[27] M. Parsa and J. J-Garcia-Luna-Aceves, "Scaleable Internet Multicast Routing," *Proc. ICCCN'95*, pp. 162–6, 1995.

[28] H. Hodel, " Policy Tree Multicast Routing: An Extension to Sparse Mode Source Tree Delivery," *SIGCOMM Computer Communications Review*, vol. 28, no. 2, Apr., 1998.

[29] V. Paxson, "End-to-End Routing Behaviour in the Internet," *Proc. ACM SIGCOMM*, Stanford, CA., Aug. 26, 1996.

[30] D. Zappala, D. Estrin, and S. Shenker, "Alternate Path Routing and Pinning for Inter-domain Multicast Routing," Tech. Report USC CS TR 97-655, U. South California, 1997.

[31] K. Carlberg and Jon Crowcroft, "Building Shared Trees using A One-to-Many Joining Mechanism," *Computer Communications Review, ACM SIGCOMM*, vol. 27, no.1, Jan. 1997, pp. 5–11.

[32] M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: Quality of Service Sensitive Multicast Internet Protocol," *ACM SIGCOMM '98*, Vancouver, British Colombia, Sept. 1998

[33] T. Bates *et al.*, "Multi-Protocol Extensions to BGP-4," IETF Draft, work in progress.

[34] D. Farinacci *et al.*, "Multicast Source Discovery Protocol," IETF Draft, work in progress, 1998.

[35] W. Fenner, "Domain Wide Multicast Group Membership Reports," work in progress, 1997.

[36] S. Kumar *et al.*, "The MASC/BGMP Architecture for Inter-Domain Multicast Routing," *SIGCOMM '98*, 1998.

[37] M. Handley, D. Thaler, and D. Estrin, "The Internet Multicast Address Allocation Architecture," IETF Draft, work in progress, 1997.

[38] D. Meyer, "Some Issues for an Inter-domain Multicast Routing Protocol," IETF Draft, work in progress, 1997.

[39] D. Thaler, D. Estrin, and D. Meyer, "Border Gateway Multicast Protocol (BGMP): Protocol Specification," IETF Draft, work in progress, 1999.

[40] H. Holbrook and D. Cheriton, "EXPRESS Multicast," submitted for publication, 1999.

[41] A. J. Ballardie *et al.*, "Simple Scalable Internet Multicast," Feb. 5, 1999 (received via private mail), submitted for publication.

[42] A. Thyagarajan and S. Deering, "Hierarchical Distance Vector

Multicast Routing for the Mbone," *Proc. ACM SIGCOMM*, Cambridge, Massachusetts, Aug. 1995.

[43] M. Handley, J. Crowcroft, and I. Wakeman, "Hierarchical Protocol Independent Multicast," University College London, Nov. 1995.

[44] C. Shields and J. J. Garcia-Luna-Aceves, "The Ordered Core Based Tree Protocol," *Proc. INFOCOM 1997*, Kobe, Japan, 1997.

[45] C. Shields and J. J. Garcia-Luna-Aceves, "The HIP Protocol for Hierarchical Multicast Routing," *Proc. 7th Annual ACM SIGACT-SIGOPS, Symp. Principles of Distributed Computing (PODC '98)*, 1998.

[46] C. Shields and J. J. Garcia-Luna-Aceves, "KHIP: A Scalable Protocol for Secure Multicast Routing," to appear in the *Proc. SIGCOMM '99*, 1999.

[47] S. Keshav and S. Paul, "Centralized Multicast," submitted to the *IEEE/ACM Trans. Net.*, Apr. 1998.

[48] M. Sola, M. Ohta, and T. Maeno, "Scalability of Internet Multicast Protocols," *Proc. Internet Society Inet '98*, Geneva, July 1998.

[49] M. Sola and M. Ohta, "Modifications of PIM-SM for Static Multicast," IETF Draft, work in progress, 1998.

[50] L. Blazevic and J.-Y. Le Boudec, "A New Multicast-Based Architecture for Internet Host Mobility," *Proc. ACM Mobicom '97*, 1997.

[51] L. Blazevic and J.-Y. Le Boudec, "Distributed Core Multicast (DCM): A Routing Protocol for IP with Application to Host Mobility," Technical Report no. SSC/1999/001, Jan. (from http://lrcwwwepfl.ch/~blazevic), 1997.

[52] D. Ooms and W. Livens, "Connectionless Multicast," IETF Draft, draft-ooms-cl-multicast-00.txt, work in progress, 1999.

[53] S. Pejhan, A. Eleftheriadis, and D. Anastassiou, "Distributed Multicast Address Management in the Global Internet," *IEEE JSAC*, Oct. 1995, pp. 1445–56.

[54] R. Braudes and S. Zabele, "Requirements for Multicast Protocols," IETF RFC-1458, May 1993.

[55] B. N. Levine and J. J. Garcia-Luna-Aceves, "Improving Internet Multicast with Routing Labels," *Proc. IEEE Int'l. Conf. Network Protocols (ICNP '97)*, Oct. 1997, pp. 241–50.

[56] R. Branden *et al.*, "Resource ReSerVation Protocol (RSVP): Version 1 Functional Specification," RFC 2205, Sept. 1997.

[57] Y. Bernet *et al.*, "Interoperation of RSVP/Int-Serv and Diff-Serv Networks," Nov. 99, IETF draft, work in progress.

[58] "Adaptive Applications," report from the IP Multicast Initiative (IPMI), Stardust Forums, Inc. available from www.ipmulticast.com, 1999.

[59] K. Almeroth, "The evolution of Multicast: from the Mbone to Inter-Domain Multicast to Internet2 Deployment," public report from the IP Multicast Initiative (IPMI), Stardust Forums, Inc. available from http://www.stardust.com/multicast/whitepapers/interdomain.htm , 1999.

[60] R. Canetti and B. Pinkas, "A Taxonomy of Multicast Security Issues," *Proc. Infocom '99*.

[61] S. Floyd *et al.*, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *ACM SIGCOMM '95*, Aug. 1995, pp. 342–56.

[62] K. Almeroth, "Managing IP Multicast Traffic: A First Look at the Issues, Tools, and Challenges," IP Multicast Initiative Summit, San Jose, California, USA, Feb. 1999.

[63] The IP Multicast Forum (IPMI) is a division of Stardust Forums, Inc. (see also www.ipmulticast.com)

[64] Internet2 is the University Corporation for Advanced Internet Development (see also www.internet2.edu)

Biography

MARIA F. RAMALHO (maria@starlab.net) is chief scientist at Starlab N.V. in Belgium. She joined Starlab in the summer of 1999. Before that, she had been working for the Corporate Research Centre of Alcatel Telecom, where she was a consultant on IP Multicasting and participated in IETF standardization activities. Prior to that, she obtained her PhD degree at Queen Mary and Westfield College, Univ. London (U.K.) on the application of fuzzy logic techniques and genetic algorithms for connection admission control in ATM networks. Dr. Ramalho graduated from the Univ. of Coimbra (Portugal) with a degree in computer science. Her main research interests are computer communications, evolutionary computing (fuzzy logic and genetic algorithms), and multimedia applications for pre-school education.