Chapter 2

Hierarchical Multicast Protocols with Quality of Service

2.1. Introduction

Multimedia applications vary radically from the traditional applications of data transfers such as email or file transfer. Indeed, these applications generally concern only two users: a source and a destination. In addition, the transmission delays do not influence the service. On the contrary, multimedia applications can imply more than two users (a videoconference for 100 people for example). In addition, these applications need short delays and flow guarantees in order to ensure a continuous playback of the multimedia flow. However, the current networks generally provide only a service, which is “at best” and point-to-point. Particular protocols and mechanisms must be developed at both the transport and application levels. Some mechanisms already exist: they enable the applications, using the multicast over Internet, to operate generally in satisfactory conditions. Hence, it becomes necessary to provide guarantees in terms of quality of service for this type of multimedia application.

The quality of service (QoS) refers to the way a packet is delivered. Thus, it is defined by the following parameters:

– the delay, which characterizes the end-to-end transfer time;

– the jitter, which represents the variation of the communication delay;

Chapter written by Abderrahim BENSLIMANE and Omar MOUSSAOUI.

– the bandwidth, which corresponds to the possible throughput between two end entities; it is limited by the throughput of traveled physical links, but also by the concurrent flows and the equipment capacity;

– the reliability, which is the average error ratio of various communication supports and equipment.

Several studies were suggested in order to guarantee a certain QoS [BRA 97], or at least to differentiate the services [BLA 98]. However, there is still a lot left to be done in order to deploy the multicast protocols while taking into account QoS.

Multicast routing algorithms and protocols have been largely studied in the literature [BAL 97, BIS 00, DEE 99, DID 97, EST 95, EST 98]. However, most of them are not scalable and do not manage the QoS efficiently. In addition, there were several studies advocating the importance of QoS in multicast routing [CAR 97, CHE 00, FAL 98, SRI 98]. These works mainly deal with the research of paths from the new members towards the multicast tree by considering the parameters of QoS, which is done by taking into account either a single path or multiple paths. However, with a single path, QoS is not necessarily considered.

On the other hand, with multiple paths, there is an overloading of the network (i.e. reservation cost, flooding), which does not enable scalability. The development of multicast routing sensitive to QoS drew less attention, even though it is indispensable for multimedia applications. As for the scalability of multicast routing protocols, the solutions based on "hierarchical trees" are very promising [THY 95, HOF 96, PRA 01]. These protocols decompose the global multicast group into separated sub-groups. Each group consists of participants of a same region or same domain. However, no protocol explains how to perform this decomposition (static or dynamic) or with what QoS parameters. In addition, for each sub-group, a representative is chosen.

In this chapter, we will present in detail a hierarchical communication architecture for large scale multicasting, which takes into consideration the QoS. Indeed, in [BEN 03], the authors used hierarchical trees, rendezvous points and criteria of QoS in order to connect the multicast group members which are scattered in the network. The global multicast group is decomposed into sub-groups based on QoS parameters. In each sub-group, a server is chosen to manage the communication within its own group. The communications among the sub-groups go through the servers of the sub-groups. The latter are connected by using either shortest path trees or a shared tree with rendezvous points. Rather than using static rendezvous points, as in [EST 98], the authors suggested using rendezvous points dynamically.

The rest of the chapter is structured as follows. In section 2.2, we will introduce the multicast routing principles and algorithms. In section 2.3, we will present the multicast routing protocols that exist in other works. Section 2.4 presents how QoS is taken into account in multicast routing. In section 2.5 we will present some hierarchical multicast routing protocols. Then, in section 2.6 we will describe a construction method of hierarchical structure for multicasting.

2.2. Multicast principle

Multicast routing is a particular technique which makes it possible to considerably reduce the transmission costs for group communications (a sender towards several destinations or several senders towards several destinations). After presenting the advantages of multicast routing, we will describe the hierarchical multicast routing algorithms and protocols. Without explicitly mentioning it, we will deal only with IP networks, the only ones currently having real multicast capacities.

2.2.1. *Advantage of multicasting*

Multicasting was introduced along with the advent of multiparty applications (Internet videoconferencing, etc.) and collaborative work applications (shared simulations, etc.). In other words, the word multicast is related to the concept of group communications, this technique having been introduced in order to reduce the communication costs for this type of applications.

With a traditional "unicast" technique, a communication that implies several recipients requires the successive sending of the same information as many times as the number of recipients. It is easy to understand the waste, in terms of bandwidth, that these repetitions of strictly identical information cause (see Figure 2.1(a)). When the nodes of the network have copy capabilities, the source can send only one message which will be copied and sent on the various branches of the multicast tree, when this is necessary (Figure 2.1(b)).

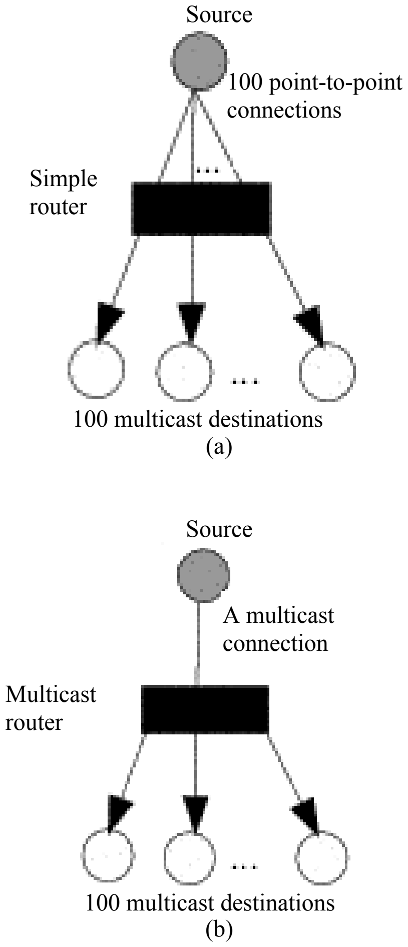

Source
100 point-to-point connections
Simple router
...
...
100 multicast destinations
(a)
Source
A multicast connection
Multicast router
...
100 multicast destinations
(b)

Figure 2.1. *Interest of multicast*

From the point of view of signaling, multicasting also introduces new facilities and new concepts. In the unicast case, the various recipients must announce themselves to the source, one by one, so that the source can send data with the address of each sender. This can also be the case for point-to-multipoint connections with Frame-Relay or ATM, where again the source must manage the group. This “source based” signaling is thus adapted to small groups or when the source wants to limit the entry in the group. For large-scale broadcasts, this technique is not doable because the group management could saturate the source.

Thus, with the new multicast techniques, recipient[1] based signaling was introduced. In this case, the source should not at all be concerned about the dimension of the group: the entire management of the tree is done by the network[2]. Hosts that wish to take part in the communication use thus specific messages to join the group. The network has mechanisms that make it possible to add a new branch leading to this destination.

2.2.2. *Technological constraints*

We have rapidly presented the main motivations which made it possible to suggest multicast routing. This technique requires implementing additional capabilities in the network:

– *copy capabilities*: we have seen the interest (in terms of economy of the bandwidth) to introduce duplication capabilities in the network. This multiplication can take place at two levels:

- *at the level of multiple access networks*: the majority of LAN segments have broadcasting capabilities which can be used in order to avoid resending the information several times,

- *at the level of the routers*: the majority of current switching matrices make it possible to duplicate an information element towards several outgoing interfaces.

In summary, the current architectures generally enable the duplication of information. The introduction of multicasting in the network is most often a software problem and not a hardware one;

– *addressing*: multicast routing requires defining group addressing[3]. To each multicast communication is associated a specific address. The machines that want to take part in the multicast session must join the address of this session. We shall note that the introduction of these addresses imposes behaviors that could not exist in the case of unicast communications:

- several stations may have the same multicast address. Indeed, all members of the group have the same multicast address, irrespective of their location on the networks (irrespective of their unicast address),

[1] This is the case in IP networks.

[2] With certain IP protocols, the source ignores even if there is a recipient in the multicast group it created.

[3] Group addresses are addresses reserved in the addressing plan. More precisely, we have here class D addresses (224.0.0.0 to 239.255.255.255) in IPv4.

- a host may have as many multicast addresses as desired, according to the number of sessions that it wants to join;

– *protocols and signaling*: it is necessary to carry out specific processes in order to build the broadcast trees. There are two types of interfaces using different procedures:

- *host-network*: messages must be defined in order to enable the hosts to inform the network of their wish to join or leave a given group. In the Internet network, multicast signaling between a host and its default (or designated) router is ensured by the IGMP (*Internet group membership protocol*) [CAI 02],

- *network-network*: procedures must be implemented in order to build or modify the multicast tree according to the requirements of the hosts. This is the role of multicast routing protocols.

2.2.3. *Main types of trees*

Before studying in more detail the various multicast routing protocols, it is necessary to present the various types of multicast trees that it is possible to build. Indeed, one of the major difficulties in the development of multicast routing protocols is the choice of the type of tree: certain trees perform well in terms of end-to-end delay but they generally use many resources. Other types of trees use the least possible amount of resources but they lead to high end-to-end delays, etc. In this sub-section, we will present the main types of trees along with their advantages and disadvantages.

2.2.3.1. *Shared tree/specific tree*

Firstly, it is important to note that multicast trees can be divided into two main categories: the shared trees or the source specific trees. A specific tree is built based on a determined source, so that it is necessary to build several trees for the same multicast group if there are several senders. Hence, we consider unidirectional trees, from the source towards the recipients (Figure 2.2(a)). On the contrary, a shared tree is established to interconnect all the members of the multicast group. Hence, we have bidirectional trees, where there is no distinction between senders and recipients (Figure 2.2(b)).

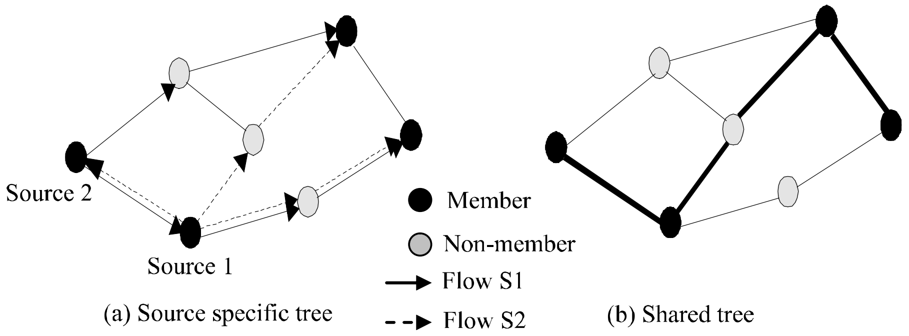


Figure 2.2. *Shared trees or source specific trees*

The difference between these two types of tree is the number of states to be stored in order to maintain the tree. In the case of a specific tree, there is a tree to describe for each source. In other words, the nodes of the tree must store a number of states of order Φ ($G*S$), where G is the number of multicast groups present in the network and S designates the average number of sources per multicast group. In the case of shared trees, on the other hand, the number of states to store does not depend on the number of sources; hence the complexity is about Φ (G).

2.2.3.2. *Shortest path tree (SPT)*

The shortest path trees (SPT) are the most frequently used trees nowadays. The construction of an SPT tree is the simplest: each leaf of the tree is connected to the source by using the shortest path defined by the underlying unicast protocol. Through construction, the transmission delays of this type of trees are minimal. From the point of view of used resources, the SPT trees are not very economical. Firstly, they are specific trees. Secondly, each branch is built independently from the others through unicast routing, without worrying about the possible proximity between the nodes of the same multicast group.

2.2.3.3. *Steiner tree*

A Steiner tree is a shared tree enabling the connection the members of the group through a given graph, and doing this by minimizing the resources used. The construction of a Steiner tree, which is a centralized construction, is a NP-complete problem, which makes it difficult to be carried out in a large size network. This is a very well known problem in graph theory and there are several works on this topic [BIS 00]. Numerous heuristic methods were suggested, the heuristic method KMB

[MAR 81], [WIN 87] being one of the most frequently used[4]. The majority of the heuristic methods lead to a problem of minimum spanning tree[5]. Through construction, the Steiner trees types are optimal in terms of cost (use of resources).

2.2.3.4. *Centered tree (CBT)*

We have seen the interest of shared trees for reducing the number of states to store at the level of network nodes. The Steiner tree makes it possible to build shared trees but the construction of these trees is very complex and requires the localization of all the members of multicast groups. The CBT (*core based tree*) algorithm [BAL 97] is an approach of the centered tree. The construction of a centered tree is extremely simple but it requires knowing the RP rendezvous point: a node is designated as RP or center. In order to connect a leaf to the multicast tree, it is sufficient to connect it to the RP rendezvous point through the shortest path (by using the underlying unicast routing). Like in the case of an SPT tree, the branches converging in the same point merge together. The performances of centered trees depend on the position of the rendezvous point. However, the rendezvous points are generally static (set once and for all by the administrator).

2.2.3.5. *Summary*

Table 2.1 summarizes the average characteristics of the three types of multicast trees we have studied. The results announced are drawn from what was said in the previous sections and from the well known simulation results in the Internet world [EST 94].

	SPT	**CBT**	**Steiner**
Complexity	Low	Low	High
Dynamic	Good	Good	Bad
Cost	High	Average	Low
State number	$\Phi\ (G{*}S)$	$\Phi\ (G)$	$\Phi\ (G)$
Delays	Short	Average	Average
Concentration	Low	Strong	Strong

Table 2.1. *Comparison of the main types of trees*

[4] Not in the real networks because this type of tree is not often used, but just for simulations. Indeed, the cost of a tree is systematically compared to the cost of a Steiner tree.

[5] It is a question of covering all the nodes of a graph by using the minimum amount of resources.

Steiner trees are particularly interesting from the theoretical point of view because they make it possible to considerably reduce the resources (number of links and states in the routers) used by a multicast group. Since they do not enable a dynamic management and they are not easy to build, Steiner trees are not currently used. The SPT trees are the most frequently used due to their simplicity and excellent performances for the multimedia flow (short delays and low concentration). However, this type of tree consumes a lot of resources. The CBT trees seem to be an intermediate solution: the delays can be quite good if the RP is well positioned and the use of resources is highly reduced compared to the SPT trees in the case of groups having several sources.

2.3. Multicast routing protocols

In this section, we will quickly describe the multicast routing protocols which are most frequently used. We will center our study on the Internet protocols since the other networks do not have such protocols[6].

2.3.1. *DVMRP*

The *reverse path forwarding* (RPF) algorithm is used in order to build a cover tree specific to a one group for each potential source of the sub-network [DAL 78]. Pruning techniques have been suggested in order to generate a multicast based-source tree from the *spanning tree* generated by RPF [DEE 90].

DVMRP (*distance vector multicast routing protocol*) [PUS 04] is a distributed algorithm which dynamically generates a multicast delivery tree for each pair (source *S*, group *G*) by using the RPM technique (*reverse path multicasting*) that is an improvement of RPF. The best known implementation of this algorithm is Mrouted under Unix. Since this implementation is very old, it is easy to realize that it is no longer fit for the dimension of the Internet of today and, consequently, it can be used only for small networks.

DVMRP consists of building a broadcast tree of datagrams sent by the source *S* to the group *G*. This tree is restricted to the branches linking the routers connected to sub-networks where the members of the group *G* are present. The DVMRP protocol is an extension of RIP (*routing information protocol*) [HED 88] which gives for each source sub-network a metric which is the evaluation of the route cost in number of hops.

[6] For example, in an ATM network there is no multicast routing precisely; the point-to-multipoint connections are opened branch by branch by the unicast routing.

A router accepts a datagram sent by *S* for the group *G* if it receives it on the interface through which the best route passes in order to go back to the source *S*, otherwise it destroys it. The router uses the routing table to determine the best route (shortest path) towards the source *S*. If the router accepts the datagram, it creates an entry (*S, G*) in its table and constructs the list of outgoing interfaces towards which it must resend this datagram. In order to prevent its neighboring routers from receiving duplicate datagrams, it resends the datagram (*S, G*) only towards the interfaces linking the routers upstream to the broadcast tree (*S, G*). The list of outgoing interfaces for the datagrams (*S, G*) can be reduced even more if the router detects that no active member of group *G* can be reached via an outgoing interface. This mechanism, which consists of limiting the broadcast tree (*S, G*) to the branches leading only to the active members of the group *G*, is called *pruning*. If an upstream router knows that it is a leaf router (because it did not receive any advertisement from *S* in reverse path), and that no member rejoined group *G* (IGMP *host report*) on all the sub-networks that it connects, then it will send a prune message towards all sources broadcasting to the group *G*, i.e. towards all routers upstream of each tree (*S, G*).

The prune messages can go step by step towards the sources *S*, thus pruning the broadcast trees. The router preserves in the entry (*S, G*) the "pruned" state of the outgoing interface through which it received a prune message and initiates a timeout associated with this state. When this timeout ends, the router modifies the state of the outgoing interface in the entry (*S, G*) and resends again the datagrams (*S, G*) towards this interface, until the possible reception of a prune message.

If, in a pruned sub-tree, a new member of the group *G* appears, the router connected on the same sub-network sends towards the upstream router a graft message. The graft messages, like the prune messages, go step by step towards the sources *S*, so as to rebuild the broadcast trees. Grafting makes it possible to graft again a pruned sub-tree, without waiting for the timeouts initiated by the previous pruning to expire.

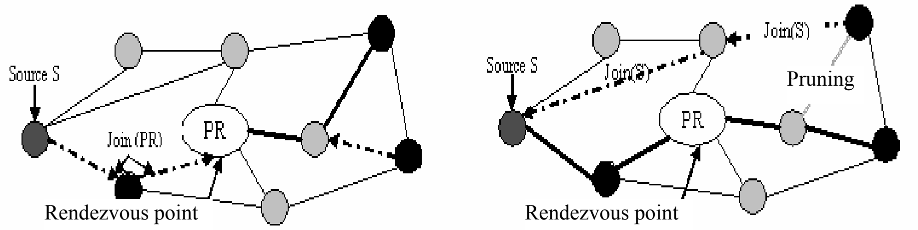
2.3.2. *PIM*

The PIM (*protocol independent multicast*) [DEE 99, EST 95, EST 98] operates on every underlying unicast protocol, contrary to its DVMRP. We have thus two distinct protocols that can be used in pairs:

– *dense mode*: PIM-DM [DER 99] is very close to the DVMRP protocol, but it uses the routing tables of the underlying unicast routing protocol. Thus, this protocol makes it possible to rebuild the SPT trees by broadcasting and then pruning the useless branches;

– *sparse mode*: PIM-SM [EST 98] builds CBT trees (by default, see Figure 2.3(a)) or SPT trees (on explicit demand, see Figure 2.4(b)). The information of a multicast group is sent towards a leaf only after the establishment of a performed branch. There is no broadcasting like in PIM-DM.

Figure 2.3 briefly synthesizes the mechanisms carried out for the construction of an SPT or CBT tree for the PIM-SM protocol. The mode of operation of PIM-SM, similar to the one of DVMRP, will not be described in this section. With the PIM-SM protocol, a centered tree is initially built, as shown in Figure 2.3(a). The rendezvous point is set by the administrator during the configuration of the routers. Then, the recipients can take the initiative to leave the centered tree and to explicitly join a given source (Figure 2.3(b)), if the traffic sent by this source is sustained (a threshold rate is set by the administrator).



(a) Centered tree (by default)

(b) Specific tree (on demand)

Figure 2.3. *Signaling of PIM-SM protocol*

The names "sparse mode"/"dense mode" are justified through the applications implied by each protocol: as shown in [EST 95], the sparse mode is more economical in terms of signaling exchanges or states to store when the group is lightly distributed in the network. On the other hand, the dense mode is more efficient when the members of the group are strongly concentrated in the network.

2.3.3. *MOSPF*

MOSPF [MOY 94] is an extension of the famous unicast routing protocol OSPF (*open shortest path first*). This protocol builds only SPT trees. The information on the multicast groups is exchanged in the network due to specific messages "*link state advertisements*". Each router has the topological view of the network and of the location of the members of the multicast group. Hence, it is possible for them (for example, the Bellman-Ford algorithm), through simple calculations, to build a representation of the SPT tree. Hence, no broadcasting or signaling exchange is required for the construction of multicast trees. Shared trees are not supported.

2.3.4. *IP multicast*

The IP multicast protocol is an extension of the IP protocol which makes it possible to send a packet to several hosts simultaneously. It is based on the IGMP (*Internet group management protocol*) [CAI 02]. This protocol enables a station to register itself dynamically in order to receive various types of IP multicast traffics. Any station which wants to have access to a group obtains the IP multicast address of that group and it belongs to this pseudo-network as long as it does not leave it. Indeed, the destination address of the IP packet determines the multicast group that wants to receive the datagrams. The address of the IP multicast group used is a class D IP address ranging between 224.0.0.0 and 239.255.255.255.

1	1	1	0	Identification of the group

IP multicast routing is performed according to the DVMRP protocol. This protocol proceeds by systematic broadcasting in order to reach the IP multicast recipients. The tunnel stations[7] that have the role of routers perform this flooding by using the reverse shortest path between a specific sender and its recipients. Mbone "*multicast back-bone*" is a virtual network on the Internet physical links. It consists of routers and stations linked by tunnels. The tunnel station knows if stations belonging to a given IP multicast group exist on the same network. The tunnel station which receives an IP multicast packet for a group to which no station belongs will return a message to the sender of the packet to notify it about this state. Each tunnel station records this state and forwards the message upstream to all IP multicast sources it has recorded. Hence, the next packets will not be forwarded to the network where there are no members for that group.

[7] Tunnel stations have a specific role on Mbone: they link the Internet network whose mode of operation is still unicast and the local networks on which packets need to be broadcasted.

2.3.5. *Limitations of the current multicast routing protocols*

The most frequently used protocols are indisputably the PIM and DVMRP protocols. However, these protocols have several limitations.

2.3.5.1. *DVMRP*

Periodic broadcast: in order to be able to be dynamically adapted to the changes in the multicast group (arrival or departure of leaves), it is necessary to rebuild the tree periodically. These reconstructions are guaranteed by a complete broadcast followed by new pruning messages.

Reliability: the integrated unicast routing protocol is a "*distance vector*" protocol that is basic and has convergence problems. In addition, the metrics used limit the surface of the network (maximum 32 hops).

Additional cost: this protocol operates with its own unicast routing protocol (distance vector type) independent of the underlying unicast routing protocol[8].

Finally, this protocol cannot build shared trees.

2.3.5.2. *PIM*

The PIM protocol does not depend on any routing protocol: hence, a PIM router cannot know if a neighboring router has a better route than it has towards a source. It is thus forced to resend the multicast datagrams on all the interfaces where multicast routers are present, except the RPF interface.

The PIM-DM protocol has almost the same limitations as the DVMRP protocol.

In the PIM-SM protocol, the rendezvous points are set by the administrator during the configuration of the routers.

Moreover, the big limitation of the existing multicast routing protocols is the fact that they are poorly adapted to large scale networks such as the Internet and that they do not guarantee any QoS.

[8] Hence, there are two unicast routing protocols which are performed in parallel, which represents a waste in terms of resource use.

2.4. Quality of service in multicast routing

Multimedia applications based on multicasting require stringent QoS conditions, such as a minimal end-to-end delay, limited jitter and efficient use of the bandwidth. As we have already presented, traditional routing protocols, such as CBT and PIM, were designed for "*best effort*" data traffic. They build multicast trees mainly based on connectivity. Such trees cannot meet the requirements of QoS because they lack resources. Recently, several multicast routing algorithms sensitive to QoS have been suggested in order to find conceivable trees. Certain algorithms provide heuristic solutions to the NP-complete problem of the Steiner tree which is the search for multicast trees with minimal cost delay constraints.

Some *single-path routing* protocols, such as the *delay-constrained unicast routing* (DCUR) [SAL 97] and the *residual delay maximizing* (RDM) protocol [SRI 98] were suggested in order to take into account the QoS. They are valid for both unicast and multicast routings. Typically, these protocols use delay and cost tables in order to make routing decisions during the establishment of a communication with QoS.

The RIMQoS (*receiver initiated multicasting with multiple QoS constraints*) protocol [FEI 00] was suggested. It supposes the existence of a unicast routing protocol with QoS which calculates the paths with QoS. A recipient router knows all the state information in its domain in order to be able to calculate an optimal route towards the source according to a cost function. The router that wishes to join the group sends a request message with the entire route it has towards the source.

Contrary to these unicast routing protocols, several multicast routing protocols were suggested in order to increase the guarantees of QoS by searching the best path among the multiple candidate paths which meet the requirements of QoS.

2.4.1. *SJP*

The SJP (*spanning join protocol*) [CAR 97] is a protocol that makes it possible to build a shared tree. It uses a type 1 – n joining mechanism which creates a source-based covering tree originating from the node which requested the joining. The covering tree used is built by an algorithm that uses broadcasting with RPF. In order to join the group, a new member forwards request messages in its neighborhood in order to find nodes in the multicast tree. When a node in the tree receives the message, it sends a response message to the new member. The new member can receive several response messages corresponding to different candidate paths. Each response message collects the properties of QoS of the path it crosses. The best path is then chosen.

Root cores are established indirectly when the first recipient joins the group. When a recipient joins a group, an intra-domain branch is set from the "egress" border recipient. In its turn, this node uses a subscription request, of 1 – n type, in order to try grafting its intra-domain branch to an existing tree. However, in the absence of the existing tree, the border node having initiated the inter-domain subscription request becomes, by default, the root core for this multicast group.

The next joining requests from the border nodes, in other domains, graft their intra-domain branches to the root core, established by the initial recipient of the multicast group.

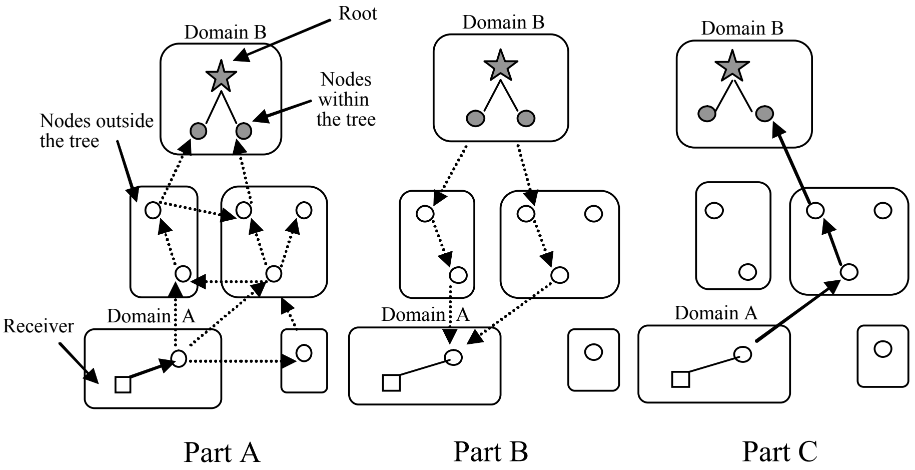


Figure 2.4. *Tree branch instantiation with one joining request and n responses*

Figure 2.4 provides an illustration in three parts in order to discover an existing tree and to establish a branch on the tree. In part (a), the leaf router (recipient) in domain A sends a *join-request* message to the border node which, in its turn, forwards the "*join*" message to all the other nodes. Part (b) shows that, when the "*join*" request reaches the nodes in the tree, i.e. in domain B, the responses are sent in unicast to the border node that initiated the broadcasting. Finally, in part (c), the border node chooses one of the paths and sends a confirmation message to one of the nodes on the tree which replied.

SJP is independent of the unicast routing protocol. However, because of the broadcasting, it overloads the network.

2.4.2. *QoSMIC*

In QoSMIC (*QoS sensitive multicast Internet protocol*) [FAL 98], the search for candidate paths is done according to two types of processes (a local search process and a search process in the multicast tree), which can be performed in parallel or sequentially.

In order to save the resources, the protocol begins by creating a shared tree. For QoS requirements, the recipients switch towards a source-based tree. In both cases, the protocol offers alternative paths for each connection. The local search is equivalent to SJP, except that only one small neighborhood is explored. The new router arches in its neighborhood a node that is already in the tree by using RPM. In order to limit the scope of request messages, a TTL is used. The search starting from the tree is done when the local search does not provide any result. That means that there is no node in the tree existing in the neighborhood controlled by the local search. An administrator node is introduced in order to manage a specific multicast group. For the search from the tree, a new member contacts the designated administrator node. The latter can command to the nodes in the tree to build paths towards the new member which then chooses the best path.

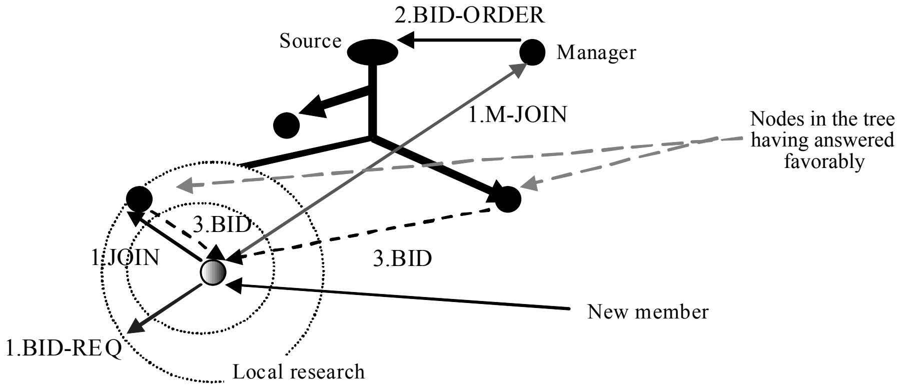


Figure 2.5. *QoSMIC with local search and research from the tree*

QoSMIC creates shared trees by default and source based trees when needed. Figure 2.5 shows the existence of several routes from the new member towards the nodes on the tree. The search for paths is done through two processes: one from the new member (local search in its neighborhood with BID-REQ message forwarding) and, if this does not succeed, another from the tree (search from the multicast tree controlled by an administrator required by the new member with the message M-JOIN). In order to launch the search from the tree, the administrator multicasts the message BID-ORDER in the tree, in order to select a sub-set of in-tree nodes. The nodes selected send BID messages to the new member. The paths of BID messages, determined by the underlying unicast routing protocols are candidate paths. The new member establishes the routing state through the selected path with the message JOIN. However, the undesired parts of the tree are pruned with the message PRUNE.

The two processes can be performed in parallel or sequentially, based on whether we want to gain time for the joining or to reduce the overload of control messages.

2.4.3. *QMRP*

QMRP (*QoS-aware multicast routing protocol*) [CHE 00] builds a shared multicast tree and can be used in an intra-domain or inter-domain routing. When a new member wants to join the group, it obtains the address of the core router of the multicast tree by requesting the session directory. Then, it sends a REQUEST message in unicast towards the core router. The REQUEST message transports the QoS requirements, for example the value of the minimal bandwidth. If a router in the multicast path does not meet the QoS requirements the request message goes back to the previous node which sent it to the core toward directions other than the one defined by the unicast routing path. When a router in the tree or the core router receives the request message, it sends an acknowledgement message towards the last router. In QMRP, two sequential processes are introduced: the single path mode and the multiple path mode, according to the conditions of the network. The protocol begins and continues with the single path mode until it finds a node which has insufficient resources to meet the joining request. When such a node is found, the protocol changes the mode.

Figure 2.6 provides an example of a multicast tree build through the QMRP protocol. We suppose that *c* is the core, and that the bold lines form the existing multicast tree. Suppose *t* is the new member and arrows form the paths taken by the REQUEST message. If each node on the path has sufficient resources, the path is a feasible branch of the tree and it is the only path searched by QMRP. If an intermediary node does not have the required resources, it will initiate the multiple path mode by sending a NACK message, back towards the previous node. Once the NACK message is received by the previous node, it diverts the REQUEST message by broadcasting it in different directions from the ones used by the unicast path.

Let us suppose that *j* does not have the sufficient resources, for example, it has no sufficient bandwidth on the link (*j, i*), in order to support the required QoS. This link is represented in the figure by a dotted line. Here, the lack of bandwidth is detected by *j* when it receives the REQUEST message and not by *i* during the sending of the request. Hence, *j* responds by sending the NACK message to *i*. When receiving NACK, *i* sends REQUEST messages to find several paths. In Figure 2.6, three paths are found, and all are feasible. When a feasible branch is detected, an ACK message is sent back through the branch to the node *t*. Three ACK messages converge then towards the node *i* which chooses the best path and rejects the others. In the general routing case, the research tree can branch itself on multiple nodes.

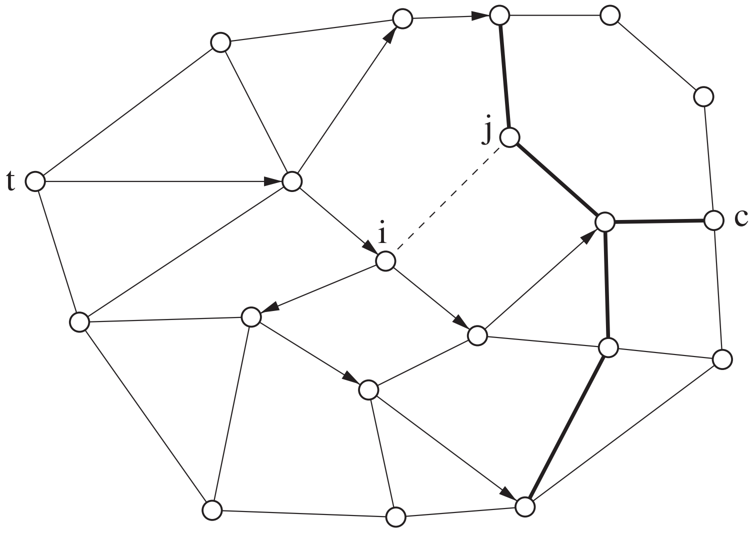


Figure 2.6. *Creation of the multicast tree with multiple paths*

2.4.4. *Conclusion*

The single path routing protocols determine only a path between the tree and the new member. Examples of such protocols are CBT and PIM. These protocols cause a low overload because the resources are reserved only on one path (usually the shortest) at one time. Hence, these protocols are suited for best effort traffic and not necessarily when a QoS requirement is considered.

The multiple path routing protocols determine several paths towards the multicast tree from the new member. In this case, the chances to find a path that meets the QoS requirements for the multicast tree are higher. However, the excessive resource reservation cost on multiple paths introduces an overload in the network. Certain protocols have been suggested: SJP, QoSMIC and QMRP.

All these protocols do not take into account all QoS requirements such as the scalability, the minimum end-to-end delay and the bandwidth. For example, SJP and QoSMIC do not enable Internet scalability because of a high overload related to message broadcasting. QMRP has the same problem in the multiple path mode. It also supposes a high join delay due to its sequential call of multiple path mode from single path mode.

2.5. Hierarchical multicasting

Apart from the QoS constraints, multimedia applications also require that the underlying multicast routing protocols should be scalable. In this section, we will enumerate certain protocols which were suggested in order to hierarchically organize communication and control.

A problem of the RP tree construction (*rendezvous point*) in PIM-SM is that the DR (*designated router*) on a LAN must be able to search for the address of the RP associated with the multicast group in order to be able to send “register” or “join” messages to the RP for the terminals. This RP search mechanism must also be able to operate on the gateways between the multicast routing protocols. Another potential problem with PIM-SM is that it requires the senders to be registered in the RPs. That requires the instantiation of a state for each sender. If the number of senders increases, like in distributed simulations, then preserving a state for each sender is not desired. The HPIM (*hierarchical PIM*) protocol [HAN 95] is based on PIM-SM and builds a shared multicast tree for the networks of *N* hierarchical levels. Like PIM-SM, HPIM does not need an advertisement message for the RP rendezvous point. Each level in the hierarchy has a candidate RP (with the secondary RP in case of failure of the primary RP) for each multicast group. Each router knows the address of the candidate RP in its level and each RP knows the address of the candidate RP of the above level. When a terminal node wants to join the multicast group, it joins the candidate RP in its level. Sequentially, each RP joins the RP of the above level until it meets an RP in the tree (this RP belongs to the multicast group).

2.5.1. *HDVMRP*

The HDVMRP (*Hierarchical DVMRP*) protocol [THY 95] and the multicast extensions to OSPF [MOY 94] are currently used in Internet for hierarchical multicasting. Mbone is organized like a single plate region where the majority of routers maintain explicit routing information for each sub-set in the network. However, HDVMRP divides the Mbone into a certain number of separate regions, hence creating two hierarchical levels. A single region identifier is assigned to each region. Figure 2.8 shows a network partitioned into four regions. Here, a region designates a cluster of routers, whereas a sub-set represents a region including one or more unicast routing domains. The intra-region multicast can use any protocol, whereas the inter-region multicast uses DVMRP for the routing between the border routers and the different domains.

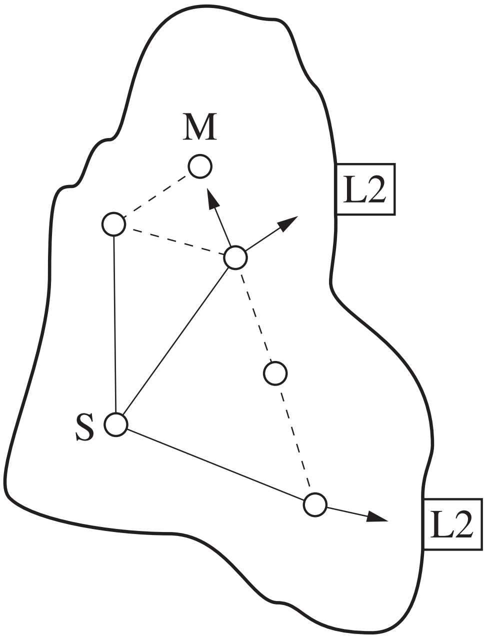


Figure 2.7. *Packet multicast routing in the region of origin*

Each region consists of one or more border routers that interconnect different regions and forward the multicast traffic among them. The multicast routers within a region carry out the level 1 (L1) multicast protocol and the border routers perform the level 2 (L2) multicast protocol in order to forward the inter-region traffic. The border routers support both protocols.

Figure 2.7 shows the intra-region routing of a multicast packet from a source *S* towards a member *M* of the destination group in region A. The dotted lines show the connectivity in region A, whereas the bold lines designate the exact paths taken in order to reach the destination members.

As for the routing of the packet outside the region of origin A towards other regions, the packet is firstly sent from the source *S* towards all the L2 routers related to region A. The packet is sent with the address of the source and the address of the group.

When a multicast packet is received by an L2 router sent from a router of one of the regions to which it is connected, it performs the following operations:

– it checks if the packet arrives from a source belonging to a sub-network of the region from which the packet arrived. If not, the packet is ignored;

– it labels the packet with the number of the region which initiates the packet;

– it sends a copy of the packet to each region to which tL2 is attached and to which it decides to send the packet. The set of regions represents an ABR multicast group "ALL_BOUNDARY_ROUTERS". The packet is encapsulated with a heading containing the address of the router L2 as source address, the ABR address as destination address and the tag representing the identifier of the source region of the packet. Then, the resulting encapsulated packet is routed through L1 routing from the sender border router towards all the other border routers of the region.

Figure 2.8 shows such a routing from the border router R1 by crossing region B and from the R2 router by crossing region C.

Finally, the packet received by the L2 routers crosses the destination regions in order to be delivered to the members of the group. Figure 2.9 shows the sub-trees formed for each one of the L2 routers related to the region C: R2, R3 and R4. We also notice that R3 prunes its sub-tree because there is no member in its downlinks.

MOSPF organizes the Internet in autonomous systems (AS), and each AS is divided into sub-groups, called *areas*. As mentioned in section 2.3, the Dijkstra algorithm is used in order to build the shortest path tree from the source, for an intraarea routing. The inter-AS and inter-area multicast communications are carried out by specific nodes: *inter-area multicast forwarders* and *inter-AS forwarders*.

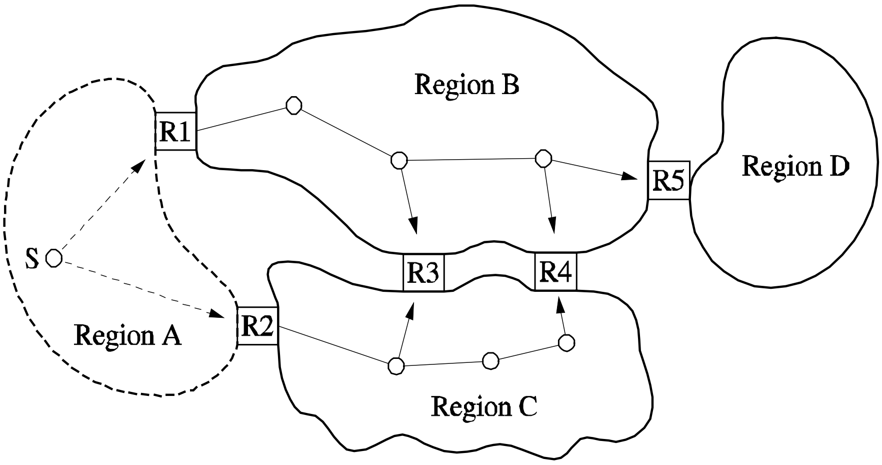


Figure 2.8. *Routing of packets among regions*

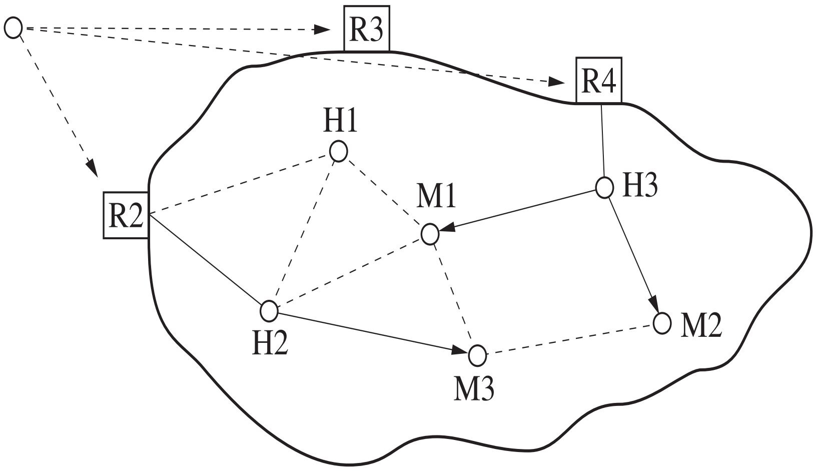


Figure 2.9. *Routing of packets within the destination region*

2.5.2. *LGC*

The *local group concept* (LGC) [HOF 96] is based on the best effort delivery model with multicast support. This requirement is in compliance with the IP protocol and the Mbone network. However, LGC is not restricted only to the Internet family protocols but it also operates for heterogenous networks, such as the satellite networks, ATM, IP, etc. It was suggested in order to overcome the scalability problem of point-to-multipoint services by taking into consideration the group sizes, the distances and the throughputs. LGC divides the global multicast group into separated groups. These groups must include the participants of the same local region, thus forming LG *local groups*. Each one of them is represented by a specific node, called local *group controller* (GC). These nodes perform the following functions:

– local transmissions: the GCs are capable of coordinating the retransmissions of lost and erroneous data in the sub-group. This reduces the delay caused by the retransmissions and decreases the overload of the source and of the network;

– local acknowledgement processing: the processing of acknowledgements by the controller makes it possible to reduce the state explosion problem in very large groups. The GCs evaluate the received control units and notify the multicast source about the status of local groups. This includes the error reports and the data flow control parameters.

In each local region, one of the recipients is determined to function as a group controller. The source itself is always defined as a controller. A controller must collect the status messages from all the members of its sub-group and route them towards the source in a single composed control message. The GCs are also in charge of the organization of local retransmissions. In order to designate a controller, LGC uses the *designated status protocols* (DSP) introduced in [PAU 94].

The concept of local group is based on grouping the recipients in a local region. Certain metrics are necessary in order to determine the distance between two nodes, but they depend on the application: delay, bandwidth, throughput, error probability, reliability, cost or number of hops. However, no article on this subject suggests a solution for distributing the recipients in the region. They do not explain how these regions are defined either.

In order to illustrate the concept of local group, we will describe the example of the following scenario. A multicast source communicates through a satellite link with four recipients, which are connected to a common switch. The satellite link is characterized by its high transfer delay and its high transmission cost. In this type of scenario, it is useful to regroup the four recipients into a single group. One of the recipients has the function of group controller. In this case, the local retransmissions do not cross the satellite link. This reduces the transfer delay and the overload of the satellite link.

2.5.3. *HIP*

HIP (*hierarchical multicast routing protocol*) [SHI 00] is an approach of inter-domain multicast routing. It introduces the idea of virtual router VR (*virtual route*), consisting of all the border routers, in order to organize the control of a full domain. VR appears as a single high level router in a shared tree. HIP performs unicast routing by using the OCBT (*ordered core based tree*) protocol [SHI 97]. In OCBT, when a branch of the tree is built, it is destroyed only when a failure of the link occurs on the branch, or when all recipients have left the branch. In addition, the protocol uses the unicast routing tables in order to make its routing decisions; the routers do not need to maintain separated tables to locate other multicast routers. OCBT operates similarly to the CBT protocol as long as a router which wants to join the group sends a "join" message to the closest core. When this message reaches this core or a router on the tree, the router replies with an acknowledgement message which crosses the reverse path of the request and establishes the branch on the tree. There is a single virtual router per hierarchical level.

HIP uses two types of addressing. An internal multicast address, called ABR (*all-border-routers*), is a simple multicast address which must deliver a single packet to all the border routers. A second address, defined for each high level domain, is called AVR (*all-virtual-router*). Any virtual router that contains the domain of the initial sender at the lowest level can subscribe to this address. Any domain that contains internal AVR recipients subscribes to the highest level AVR address. In order to avoid these confusions, a router that constitutes the root of the tree is designated by CP (*center point*) rather than core because the core is local to a single domain and does not have a scope on the entire tree. An OCBT domain is a section of the network under the administrative control of a single entity and with border routers well defined enabling the control and the external connection.

In order to enable the adhesion of the members belonging to different domains to the multicast group, HIP broadcasts the information on CP.

The following figures show the way in which HIP broadcasts the information on CP and the establishment of the shared tree. We have a virtual router A, which did not subscribe to AVR to receive the information on CP, and a virtual router C which contains a DVMRP domain that must be notified when a multicast group becomes available and then must subscribe to the AVR address. The virtual router B contains the physical CP of the group, making both E and F virtual center points for their levels. The establishment of the multicast group starts by distributing the information on the CP. The new CP announces its availability on the ABR and AVR addresses. The AVR announcement concerns only the domains from E to C because there is no subscription to the AVR address in the superior level F. The announcement on the ABR address will exit domain E towards domain D, where it is sent through the domain in unicast and routed to the exit router for the global domain F. The routing of the location message of CP is shown in Figure 2.10.

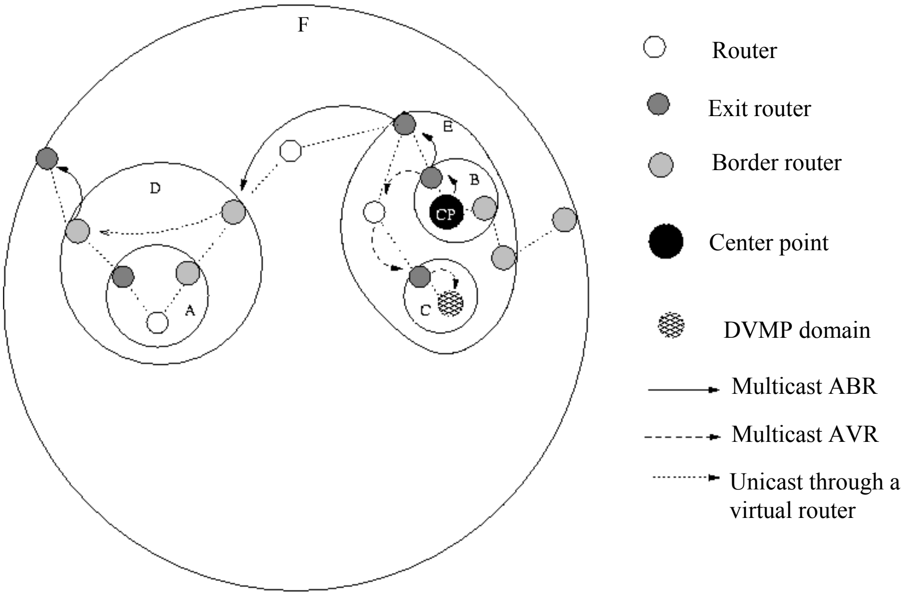


Figure 2.10. *Topology and broadcasting of the location of CP*

We shall note that domains A and D manage only the routing (and not the storage) and they do not store the information on CP because they are not ABR recipients. When an exit router of a DVMRP domain receives CP information, it broadcasts it in its domain. The controller of domain C sends a subscription request towards the CP. This request travels first through the exit router of domain C. Then, it is acknowledged and sent towards the exit router of domain B by crossing the exit router of E (superior level). This process is shown in Figure 2.11.

When the unique router in domain A receives an IGMP message from a sub-set that wants to perform multicasting, it selects the border router and subscribes to it. The controller of A must then search for the good CP for the group. The search is done because A neither read nor recorded the information on CP. In the example presented in Figure 2.12, we show the way A obtains the location information of CP. According to the example, the controller of domain F has this information. Indeed, A sends a location request of CP to the ABR group of its domain D. This request is then received by the outgoing router of its domain. Since the first location information of CP goes through D without being either read or recorded, the controller of domain D must also route a request to the superior level, i.e. domain F. In this case, the controller of domain F responds to this request by sending a message to the controller of domain D. When receiving this response, the controller of domain D realizes that the outgoing router must be changed in order to provide the shortest path towards the indicated CP. The location of CP is sent to A and the outgoing router of this group is changed. In turn, the controller of domain A realizes that the other border router provides the shortest path towards the CP. Figure 2.13 shows this subscription initiated by the recipient.

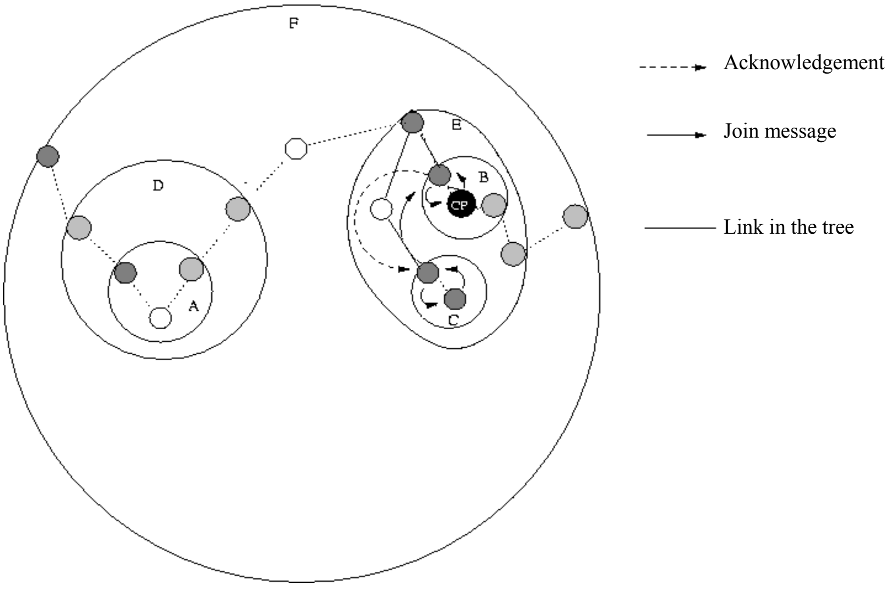


Figure 2.11. *Subscription requests in a DVMRP domain*

The main difference between HPIM and HIP is the location broadcasting process: either from the candidate RP or from the virtual router. Without exhaustively knowing the topology of the network and of the set of recipients, these protocols have the difficulty of placing the cores or the RPs in a hierarchical structure. However, the two protocols provide inter-domain protocols which can function with the intra-domain protocols such as DVMRP or MOSPF. On the contrary, they do not consider QoS.

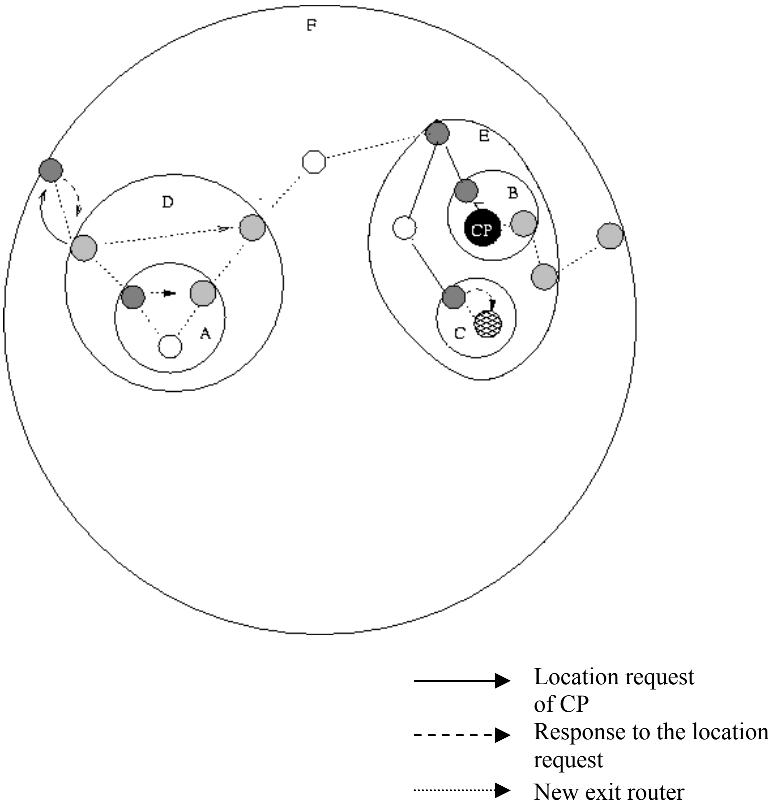


Figure 2.12. *Information request of the location of CP initiated by a recipient*

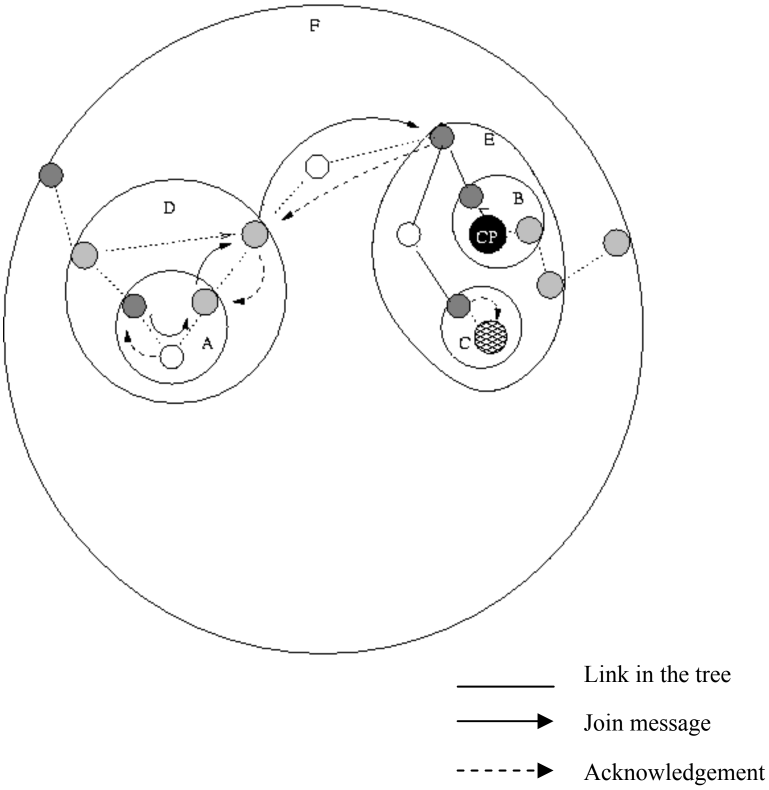


Figure 2.13. *Subscription process initiated by the recipient*

2.5.4. *QHMRP*

The QHMRP (*QoS-aware hierarchical multicast routing protocol*) [PRA 01] is different from QMRP; QHMRP uses a hierarchical network model and performs broadcasting. QHMRP uses an approach of complete meshing to organize the network in multiple levels where a domain is represented through its border routers. The concept of domain controller is used to coordinate the construction of shared multicast trees.

The controllers of various domains store information on the multicast trees and enable the operation of QHMRP. As for the rendezvous point in CBT and PIM-SM, the controllers do not take part directly in the tree. A controller of a domain has the list of all routers that are in the tree. The high level controller has the addresses of the controllers of sub-domains which own one or more routers in the tree. If there is a multicast tree, then there is at least one controller in each level which knows.

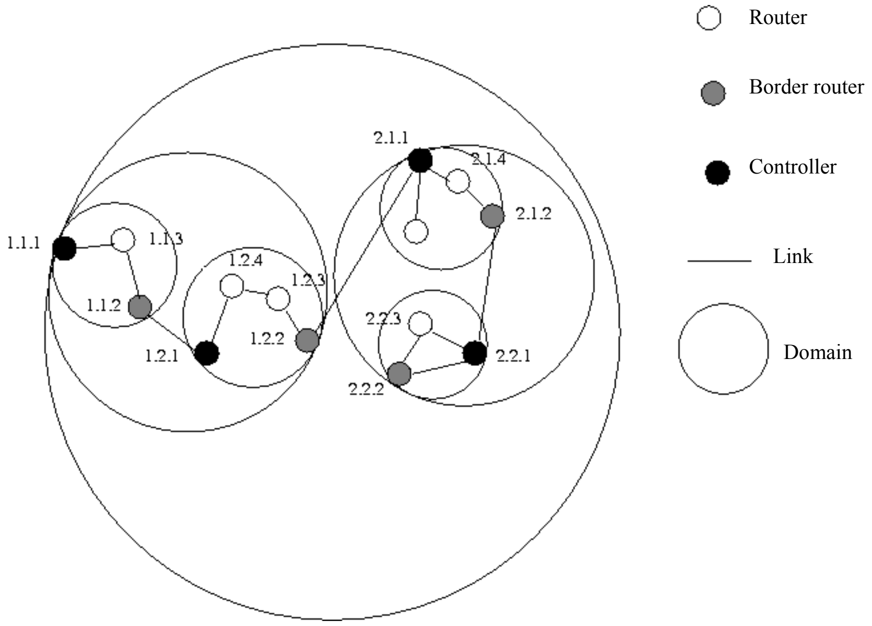
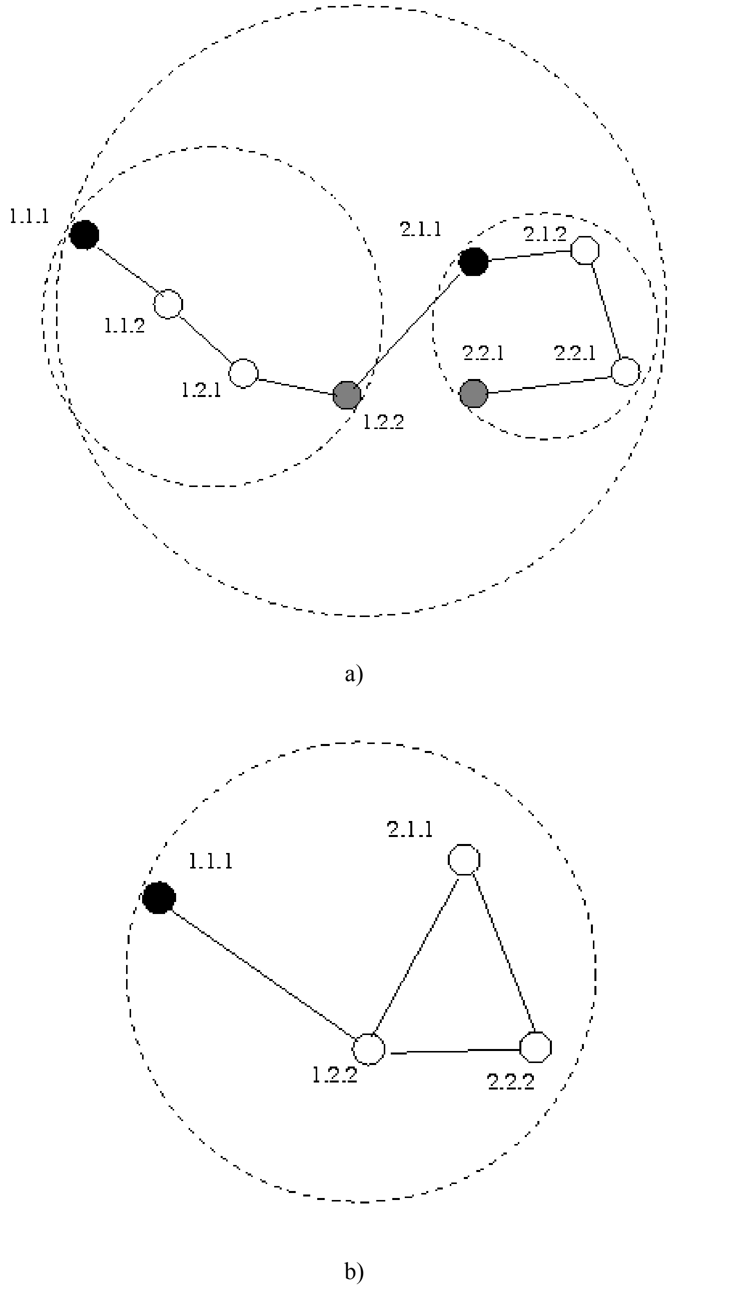


Figure 2.14. *Model of hierarchical network*

When a router wants to join the multicast tree, it sends a *JoinRequest* message to its parent controller. If the controller knows about the multicast tree, then it routes the request towards all the routers in the tree or towards all controllers that have routers in the tree. Otherwise, the controller routes the request to its parent controller.

If a multicast tree exists, then it is guaranteed that the JoinRequest message will reach a controller that knows about the existence of the multicast tree. The routers in the tree receiving the request send broadcast messages towards the end router. This broadcasting technique, from the routers of the tree towards the end router is called *reverse flooding*.



a)

b)

Figure 2.15. *Aggregated topology (a) level 2 domains and (b) level 3 domain*

An addressing system in the form of an n-tuplet is used only to identify a router in the network. The address of a router is expressed as (i_{L-1}. i_{L-2}. … i_3. i_2. i_1. i_0), where i, $j = 0, 1, 2, \dots, (L-2), (L-1)$, are non-zero positive integers. Here, i_j is the number of the sub-domain of the domain $(j+1)$ which the router belongs to. An example of three level hierarchical network showing the numbering system is presented in Figure 2.14. The size of the routing table of each router increases with the level of the domain in the hierarchy. For example, the highest level domain (level 3) in the example contains all the routers.

An aggregation of topologies is used in order to enable scalability by reducing the size of the routing tables. In this approach, each router stores several routing tables: one for each level it belongs to. For example, a router belonging to the number i domain has i routing tables for the levels 1, 2, …, i. For the aggregation of topologies, QHMRP uses a complete meshing where a sub-domain is represented by its border routers in the parent domain. The connection between two border routers in a sub-domain is represented by a logical link in the parent domain. The cost of the link is the minimal distance (in number of hops) between the two border routers. Figure 2.15 shows the aggregation of topologies of the network in Figure 2.14.

2.5.5. *Conclusion*

These protocols do not provide any mechanism that makes it possible to decompose the global multicast group into areas or sub-groups. In order to overcome these limitations, i.e. the scalability problems and the guarantees of QoS, the authors in [BEN 03] suggested a hierarchical structure enabling an efficient communication between the members of the multicast group supporting guarantees of QoS. This structure consists of regrouping the members, according to their location and their response to the QoS application requirements, into separate groups. These groups communicate with each other according to two different multicast methods. The first method consists of using the concept of centered trees. While the work in [EST 98] uses the static rendezvous points, the suggested method determines the rendezvous points dynamically. The second method consists of using the shortest path tree to connect the groups. Hence, this multicast architecture enables scalability and is sensitive to QoS.

In the following section, we will present this hierarchical multicast communication technique. The interest is to show in detail the decomposition of the global multicast group into sub-groups according to certain QoS parameters. Then, we will describe two methods which were presented in order to build the multicast trees between the groups. The choice of a technique over another depends on the multimedia application.

2.6. Hierarchical structure for multicasting

As we have already seen in the previous sections, trees or hierarchical structures offer interesting properties to support multicasting for large groups while taking into account QoS. Since the Steiner trees are too complex and they require information on the location of the members of multicast groups, in this section we will deal with the centered trees and with the shortest path trees.

2.6.1. *Context of the system*

We will consider a multicast group *M*, consisting of *N* processes[9] $\{P_1, ..., P_N\}$ distributed across various interconnected networks. These processes have different identities and communicate with each other the Internet network. Moreover, they take part in the same multimedia conference, as a videoconference. The communication between two processes P_i and P_j can be done by different paths.

2.6.2. *Construction of local groups*

Considering the high number of participants, it is interesting to divide the multicast group *M* into sub-groups according to the concentration of its members in various regions in order to ensure QoS (bandwidth, communication delay between processes, etc.). The construction of the groups was previously introduced in [BEN 02]., The authors suppose that the processes communicate, logically, between them via virtual channels: direct links. In [BEN 03], the authors use multi-hop paths, i.e. they use the physical links existing between the members of the multicast group. In order to build these local groups, they proceed in three stages.

2.6.2.1. *Construction of the neighborhood*

This first stage consists of building a neighboring group GV_i for each P_i process by using delay constraints and a time to live (TTL) message scope (expressed in hop number). Given a roundtrip delay threshold *D* and a TTL message scope, each process builds its neighboring group containing only the members of the multicast group. A process P_j belongs to GV_i if the TTL from P_i to P_j (decremented hop by

[9] The P_i process represents a terminal (source or destination).

hop) is non-zero, and if the roundtrip delay between P_i and P_j is inferior to a certain threshold *D*.

2.6.2.2. *Construction of transit groups*

Multimedia applications carry out data flows whose characteristics and requirements in terms of QoS (bandwidth, routing delay, error ratio, etc.) are very different. In the case of Internet, these flows generally cross a succession of autonomous networks, each of them being able to have its own QoS management policy. In this work, in order to build the neighborhoods of the processes, the parameters considered are the delay and the TTL. In order to guarantee more QoS, each process builds its own transit group by executing tests, initially presented in [BEN 02], on the capability of processing and storing media units:

– *test on the processing capability*: this basically means to verify if a process has a sufficient processing capability in order to process the data flows coming from its neighboring members. A process must be able to process all the media units generated at the same time by the neighboring processes within a period of time which is no longer than the processing time of a media unit;

– *test on the available memory space*: this consists of verifying if a process is capable of storing all media units coming from the various processes of its neighborhood in order to compensate for the delay variations (jitters) of these paths.

According to these two tests, each process builds its own transit group based on its neighboring group.

2.6.2.3. *Grouping and election*

Once the transit groups are built, each process knows the members of it own transit group and the paths linking them to their members. However, certain processes may belong to several transit groups at the same time. In order to solve this problem, a mechanism must be implemented in order to remove the useless connections. It consists of creating local groups based on these transit groups, provided each process belongs to a single local group. Indeed, in each transit group, all processes notify their own members by broadcasting the content of their group. Once these messages are received, each recipient selects the maximum of processes existing simultaneously in these transit groups. If a member belongs to several local groups, it is placed in the one having the smallest dimension. This makes it possible to balance the number of processes in these groups.

Hence, the multicast group *M* is divided into local sub-groups (Figure 2.16(a)). Each member of *M* belongs to a single local group (*GL*). Then each local group *GL* must elect its local server and its secondary server in order to represent it.. In other words, a process communicates with the other members of the multicast group only through the server of its local group. For example, if a process wants to send a message to the other members of *M*, then it sends it through the server of its local group. This server sends the received message to the other members of its *GL* and to the other local servers. When a local server receives a message from the exterior, it broadcasts it to the members of its own local group. The main role of the secondary server is to replace the local server in case the latter leaves the multicast group, or if an involuntary failure occurs.

The election of the local server can be done in several ways:

– to elect the process having the biggest memory available. Indeed, a server needs available memory, more than a simple process, in order to be able to store the media units coming from the members of its *GL* and from the other local servers;

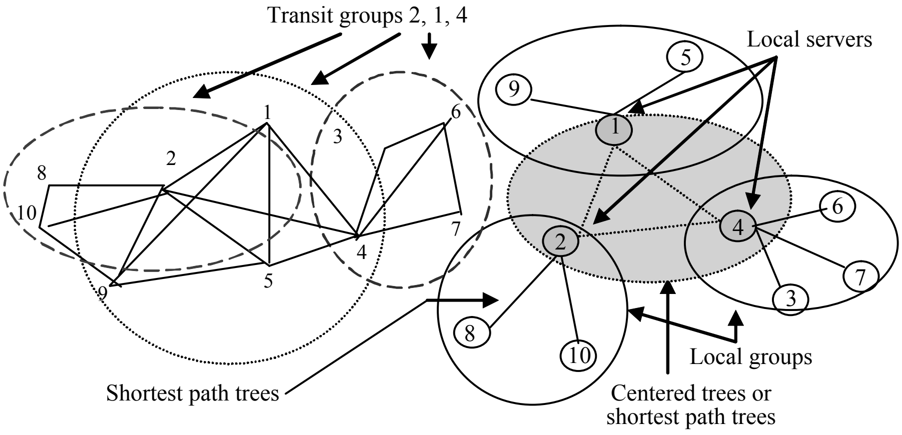
– to elect the process that minimizes the roundtrip average delay so that the QoS respects the delays;

– to elect the process that enables the exit to the outside.

The secondary server is the one that is classified immediately after the local server in the election process. Once a local server is elected, it sends a message to the members of its group in order to confirm the connection.

2.6.3. *Construction of hierarchical trees between servers*

Now, we shall suppose that *k* local groups were built and that each one of them has a local server (Figure 2.16(b)). Let $\Gamma = \{S_1, \ldots, S_k\}$ be the set of these local servers which, eventually, are numerous and scattered in various networks. In addition, in the case of videoconferences, there can be several sources in the same multicast group. Hence, we need to find a broadcast tree to link these servers in order to decrease the consumption of bandwidth and to minimize the end-to-end delay between the members of the multicast group. For this, two methods are suggested in [BEN 03]. We will present the centered trees in the first method and the SPT trees in the second one.



(a) Partition into transit groups (b) Partition into local groups

Figure 2.16. *Construction of local groups*

2.6.3.1. *Use of centered trees*

The method suggested in this section uses the centered trees simply and makes it possible to minimize the resources used. In addition, the average performances (i.e. cost and delay) are reasonable for the applications consisting of several senders and recipients. However, the choice of the rendezvous point and the associated problems make their implementation difficult in the large scale networks. Generally, and in other works, the rendezvous points are chosen statically (set once and for all by the administrator). In this work, the rendezvous points are determined dynamically. They enable the subscription to only local servers and not to all the other members of the multicast group. In addition, they are used in order to avoid the problem of traffic congestion in the neighborhood of a single rendezvous point.

PR\S	S_i	S_j	S_k	S_l
PR_1	dar_{1i}	dar_{1j}	dar_{1k}	dar_{1l}
PR_2	dar_{2i}	dar_{2j}	dar_{2k}	dar_{2l}
…				

Table 2.2. *Roundtrip delay matrix between the rendezvous points and the servers*

We assume that S_p is the server of the local group where there is the source which will be the first to broadcast the media units to the other members of the multicast group M. This server will represent the first rendezvous point PR_1.

Initially, PR_1 does not have any information on the addresses of the other servers. Consequently, it broadcasts hop by hop a message, called INIT (adrPR_1, adrG, TTL), containing the address of the rendezvous point PR_1, the address of the multicast group G and a TTL maximum scope. Each local server S_i that receives this message sends an acknowledgement to PR_1, by using a message called ACKNOWLEDGEMENT (adrS_i, adrPR_1).

When PR_1 receives the message ACKNOWLEDGEMENT (adrS_i, adrPR_l) from S_i, it calculates the roundtrip delay dar_{1i}. If this delay is inferior to the threshold D, then PR_1 selects the server S_i and adds it to its autonomous domain, a set called DA_1. Then, it places in a set S all the other servers which are not yet linked to the broadcast tree (S:=Γ- DA_1). After the construction of domain DA_1, PR_1 stores the delay calculated between itself and each server S_j belonging to S, in the MAT matrix so that it can determine the next rendezvous points. Then, it sends a message called SUCCESSOR (LS, adrPR_1, S, MAT), to these servers S_i constituting the list LS. These servers do not have to belong to DA_1 and must be the closest to PR_1, on separate paths, by taking as metric the roundtrip delay according to the MAT matrix. Table 2.2 illustrates the storage matrix of the delays calculated by the rendezvous points.

When a local server S_i of LS (in Figure 2.17, we have PR2 and PR4) receives the message SUCCESSOR (LS, adrPR_1, S, MAT), it will become a rendezvous point marked PR_2 and then it will play the role of initiator server for the set of members of S-LS by sending a JOIN message (adrPR_2, adrS_j) to each one of them.

When a server receives the JOIN message (adrPR_2, adrS_j), it sends the message ACKNOWLEDGEMENT (adrS_j, adrPR_2) to PR_2. Then, PR_2 calculates the minimal roundtrip delays between itself and the other members of the set S, and performs the following operations:

– selects all servers of S having a roundtrip delay inferior or equal to the delay threshold D and places them in its autonomous domain DA_2;

– deletes from S the servers which were selected. More exactly: S:=S-LS-DA_2;

– deletes the columns corresponding to the selected servers from the MAT matrix;

– adds in MAT a line to store the minimal roundtrip delays between PR_2 and all servers of S. In other words, the servers which do not belong to any domain;

– searches for the lowest values in MAT and takes the pairs (PR_2, S_q) corresponding to these values. Hence, the next rendezvous points S_q are determined dynamically in the list LS, among the servers of the set S whose parent is PR_2;

– sends the message SUCCESSOR (LS, adrPR_2, S, MAT) to the next rendezvous point S_q, marked PR_3;

– finally, PR_2 joins the local servers of its domain and its parent rendezvous point PR_1 through the shortest path tree (SPT) whose root is PR_2.

The algorithm continues with PR_3 and so on until the set S is empty. In other words, until all servers are linked to the multicast tree (Figure 2.17).

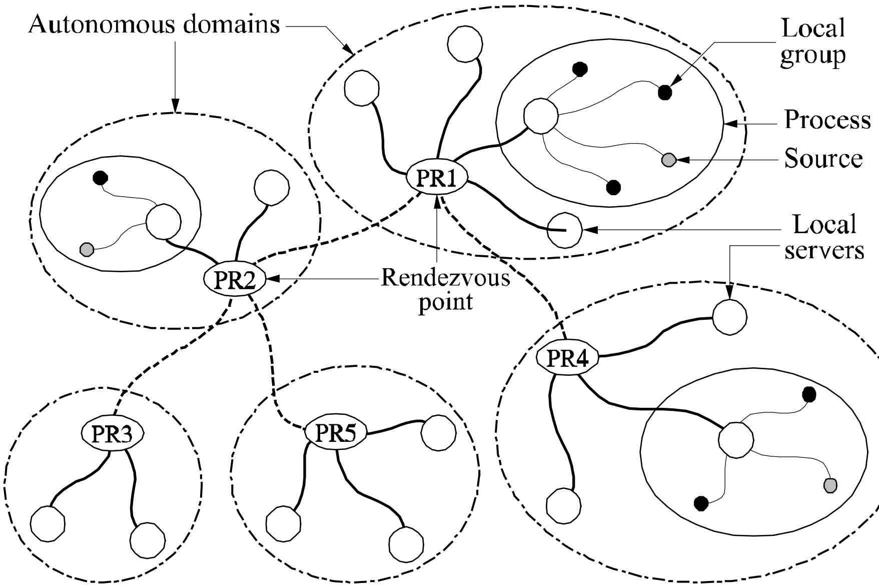


Figure 2.17. *Hierarchical structure of a multicast group with rendezvous points*

2.6.3.2. *Use of SPT trees*

In the second method, the SPT trees are used to link different servers. The SPT trees are the most frequently used due to their simplicity and excellent performances for the multimedia flows or interactive flows (low delays and low concentration). However, this type of tree consumes a lot of resources (number of states, number of used links). In order to overcome this problem, the SPTs are built on demand. In other words, we build the SPT tree for a local server S_j if and only if the local group of this server contains at least one source which wants to broadcast the multimedia flow to the other members of the multicast group.

As in the first method, we will assume that S_j is the local server of the group where there is the source which wants to be the first one to broadcast the media units to the other members of the multicast group M. S_j broadcasts hop by hop an initialization message INIT (S_j, adrG, TTL). Since this message contains the address of the multicast group, only the members of this group have the right to deliver the message. When receiving the message, all members of M which are not servers ignore it. On the other hand, each local server joins server S_j by the shortest path[10] by using the underlying unicast routing protocol. Once the SPT tree is built, the root server S_j sends a message INFO(S_j, Γ) containing the addresses of all local servers of the multicast group M to the child servers.

After receiving this message, each child server becomes aware of the others. For the moment, we have built a single broadcast tree consisting of the SPT tree between the local servers and their local SPT trees. Then, if a participant m_i of the multicast group M wants to become source of a flow, it must send to its local server S_k a REQUEST message (m_i, S_k). The server S_k joins all the other servers through the shortest path by using the underlying unicast routing protocol. Then, it broadcasts the flow coming from m_i to the other members of its own local group and to the other local servers by using the broadcast tree built.

2.6.3.3. *Comparison between the two methods*

In the first method, a single broadcast tree is built by using meeting points dynamically. This makes it possible to optimize the resources used and to reduce the consumption ratio of the bandwidth. In the second method, as many SPT trees as local group servers in which there is at least one broadcast source are built. Hence, the use of resources is very high in the case of several senders and recipients. However, the second method is more efficient for multimedia applications. Indeed, compared to the first method, it makes it possible to considerably minimize the communication delay and to efficiently reduce the traffic concentration. It would be better to use the first method for the multiparty applications that do not require strong temporal constraints.

[10] By taking as metric the delay.

2.6.4. *Management of the hierarchical structure*

Once the hierarchical structure is built, a mechanism must be implemented in order to manage the join and leave of the multicast group (M) members. This mechanism is described as follows:

– if a simple member wants to leave the multicast group M, then it is removed from its local group by all the other processes of the same group via the local server;

– if a new process P_n wants to join the multicast group M, it will broadcasts the JOIN message (adrP_n, adrG, TTL). With the help of this message, the process P_n requests to the servers of its region to take part in the multicast group M. During the receiving of a JOIN message (adrP_n, adrG, TTL) with non-zero TTL, all simple participants ignore this message. However, each local server performs tests on its processing capability and its available memory space. When receiving this message, any server capable of connecting the process P_n sends a reply message RECP (adrS_i, Bd_i, Nb), where adrS_i is the address of the sender server S_i, Bd_i is the available memory space of S_i and Nb is the number of members of its local group GL_i. If the process P_n receives several RECP messages, it will choose the local group of smallest size (number of members) or the one with a server having the biggest memory space available. Once the process P_n has chosen the local group, it joins its server through the shortest path (by taking as metric the delay);

– if a local server wants to quit the multicast group or if a failure occurs, then the secondary server of this group will play the role of the principal server. For this, the principal server informs the secondary server each time there is a change by using a particular information message which contains all necessary information on the management of the local group and the other local servers. Periodically, the principal server sends a control message so that the secondary server can detect if there is a problem or not. Once the secondary server is in place, it notifies the members of its local group that it is of the new local server. Then, it joins each member of its group through the shortest path. Then, a new secondary server is chosen by the members of the local group. In the case of the structure using dynamic rendezvous points, the new local server joins the rendezvous point of its autonomous domain through the shortest path. In the case of the structure using direct links between the local servers, the new local server joins each server of the local group containing at least one source of flow through the shortest path;

– if a new server wants to join the multicast group, then it broadcasts a message to the address of this multicast group. Each local server receives this message and it sends it an acknowledgement message containing its address. In the case of the structure with dynamic rendezvous points, this message contains also the address of the rendezvous point of the autonomous domain of the sender server. The new local server joins the closest rendezvous point through the shortest path. In the case of the structure using direct links between the local servers, the new server joins each server of the local group containing at least one source of flow through the shortest path.

2.7. Conclusion

In this chapter we have presented the main multicast algorithms and protocols. Very few of them take into consideration the QoS necessary to multimedia applications. Hence, we have presented multicast protocols which take into account the QoS. Then, we were interested to the scalability problem of the multicast protocols for which we have presented hierarchical multicast routing protocols. We believe that these protocols are more adapted to support a high number of participants. In order to illustrate this, we have presented a method to construct a hierarchical architecture for multicast. This architecture makes it possible to guarantee QoS. Indeed, it reduces efficiently of the occupation ratio of the bandwidth, the communication delay and the memory size necessary to store the media units dedicated to the synchronization in the multimedia applications.

There is still a lot to be done in order to deploy protocols that enable scalability and guarantees of QoS. However, the research progresses and we can hope for Internet multicast protocols soon.

2.8. Bibliography

[BAL 97] BALLARDIE A., "Core Based Trees (CBT version 2) Multicast Routing; Protocol Specification", *RFC 2189*, 1997.

[BEN 02] BENSLIMANE A., ABOUAISSA A., "Dynamical Grouping Model for Distributed Real Time Causal Ordering", *Computer Communications Journal*, vol. 25, p. 288-302, 2002.

[BEN 03] BENSLIMANE A., MOUSSAOUI O., "A Scalable Multicast Protocol with QoS guarantee", *Proc. IEEE/IFIP Net-Con 2003 Int. Conference on Network Control and Engineering For QoS, Security and Mobility*, Kluwer Academic Publishing, p. 1-13, Muscat, Oman, 2003.

[BIS 00] MUKHERJEE B., SAHSRABUDDHE L.H., "Multicast Routing Algorithms and Protocols", *IEEE Network,* p. 90-100, 2000.

[BLA 98] BLAKE S., *et al.*, "An Architecture for Differentiated Services", *RRFC 2475*, 1998.

[BRA 97] BRADEN R., *et al.*, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", *RFC 2205*, 1997.

[CAI 02] CAIN B., *et al.*, "Internet Group Management Protocol, Version 3", *RFC 3376*, 2002.

[CAR 97] CARLBERG K., CROWCROFT J., "Building Shared Trees Using a One-to-Many Joining Mechanism", *Computer Communication Review*, no. 1, p. 5-11, 1997.

[CHE 00] CHEN S., NAHRSTEDT K., SHAVITT Y., "A QoS-Aware Multicast Routing Protocol", *IEEE INFOCOM*, 2000.

[DAL 78] DALAL Y.K., METCALFE R.M., "Reverse Path Forwarding of Broadcast Packets", *Communications of the ACM*, vol. 21, no. 12, p. 1040-1048, 1978.

[DEE 90] DEERING S., CHERITON D., "Multicast Routing in Datagram Internetwork and Extended LANs", *ACM Transactions on Computer System*s, vol. 8, no. 2, 1990, p. 85-110.

[DIO 97] DIOT C., DABBOUS W., CROWCROFT J., "Multipoint Communications: A Survey of Protocols, Functions, and Mechanisms", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, p. 277-290, 1997.

[EST 94] Estrin D., WEI L., "The trade-offs of multicast trees and algorithms", *IEEE ICCCN'94*, August 1994.

[EST 95] ESTRIN D., WEI L., "Multicast Routing in Dense and Sparse Modes: Simulation Study of Tradeoffs and Dynamics", *IEEE ICCCN 95*, 1995.

[EST 98] ESTRIN D., *et al.*, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", *RFC 2362*, 1998.

[FAL 98] FALOUTSOS M., BANERJEA A., PANKAJ R., "QoSMIC: Quality of Service Sensitive Multicast Internet Protocol", *SIGCOMM 98*, 1998.

[FEI 00] FEI A., GERLA M., "Receiver-Initiated Multicasting with Multiple QoS Constraints", *Infocom*, vol. 1, p. 62-70, 2000.

[HAN 95] HANDLEY M., CROWCROFT J., "Hierarchical Protocol Independent Multicast (HPIM)", 1995, (available at ftp://cs.ucl.ac.uk/darpa/IDMR/hpim.ps).

[HED 88] HEDRICK C., "Routing Information Protocol", *RFC 1058*, IETF, 1988.

[HOF 96] HOFMANN M., "A Generic Concept for Large-Scale Multicast", *Proc. of Int. Zurich Seminar on Digital Communications IZS 96*, p. 95-106, Zurich, 1996.

[MAR 81] MARKOWSKY G., KOU L., BERMAN L., "A Fast Algorithm for Steiner Trees", *Acta Informatica*, vol. 15, p. 141-145, 1981.

[MOY 94] MOY J., "Multicast Routing Extensions for OSPF", *Commun. ACM*, vol. 37, p. 61-66, 1994.

[PAU 94] PAUL S., SABNANI K.K., KRISTOL DAVID M., "Multicast Transport Protocols for High Speed Networks", *Proc. of Int. Conf. on Network Protocols*, Boston, 1994.

[PRA 01] PRADHAN S., LI Y., MAHESWARAN M., "QoS-Aware Hierarchical Multicast Routing on Next Generation Internetworks" *Proc. of Int. Conf. on Performance, Computing, and Communications*, Phoenix, Arizona, 2001.

[PUS 04] PUSATERI T., “Distance Vector Multicast Routing Protocol”, *draft-ietf-idmr-dvmrp-v3-as-01, Internet-Draft*, 2004

[SAL 97] SALAMA H.F., REEVES D.S., VINIOTIS Y., “A Distributed Algorithm for Delay-Constrained Unicast Routing”, *IEEE Infocom*, 1997.

[SHI 00] SHIELDS C., GARCIA-LUNA-ACEVES J.J. “HIP – A Protocol for Hierarchical Multicast Routing”, *Computer Communications*, vol. 23, no. 7, p. 628-641, 2000.

[SHI 97] SHIELDS C., GARCIA-LUNA-ACEVES J.J., “The Ordered Core Based Tree Protocol”, *IEEE INFOCOM*, Kobe, Japan, 1997.

[SRI 98] SRIRAM R., *et al.*, “Preferred Link-Based Delay-Constrained Least Cost Routing in Wide Area Networks”, *Computer Communication*, vol. 21, no. 18, 1998.

[STR 02] STRIEGEL A., MANIMARAN G., “A Survey of QoS Multicasting Issues”, *IEEE Communications Magazine*, p. 82-87, 2002.

[THY 95] THYAGARAJAN A., DEERING S., “Hierarchical Distance-Vector Multicast Routing for the Mbone”, *Proc. of the ACM SIGCOMM 95*, Cambridge, p. 60-66, 1995, no. 9, 1998.

[WIN 87] WINTER P., “Steiner Problem in Networks: A Survey”, *Networks*, vol. 17, p. 67-129, 1987.