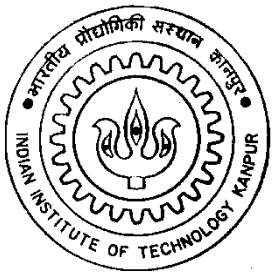
SCAMP : SCAlable Multicast Protocol for Communication in Large Groups

by
ATUL VADERA



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
January, 1999

SCAMP : SCAlable Multicast Protocol for Communication in Large Groups

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Atul Vadera

to the

DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

January 1999

CERTIFICATE

This is to certify that the work contained in the thesis entitled **SCAMP : SCAlable Multicast Protocol for Communication in Large Groups** by **Atul Vadera** has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Dheeraj Sanghi,
Assistant Professor,
Department of Computer Science & Engineering,
Indian Institute of Technology, Kanpur.

Abstract

This report proposes a new scalable multicast protocol for communication within a large group over Internet. There are many existing and new applications in which the groups involved are very large. An example would be Videoconferencing amongst a large and widely distributed group. The members of such a group can be spread across the entire Internet with no uniformity in their pattern of distribution. In such applications, there will be one or at most a few sources and a large set of destinations or receivers. The existing multicast routing protocols do not support such large groups efficiently. They incur large overheads and do not scale well. In this thesis we have proposed a new protocol called SCAMP (SCAlable Multicast Protocol) that incurs less cost and is scalable.

Most protocols either build source-based trees or shared trees. Source-based trees are those which have their root located at the node sending the data. In shared trees, a few routers in the Internet act as root of the trees and distribute the data among the members of the group which are on this tree. Source-based trees have the advantage that they support high data rate. The advantage of using shared trees is that they can support sparse groups. We have proposed a protocol which combines the advantages of both type of trees. Our protocol builds both the source-based and shared trees. The shared tree is used for signalling purposes. It conveys the information regarding the source to all the members of the group. Only one signalling tree is used for all the existing multicast groups. A multicast group is referred to a set of nodes across the Internet which can be identified by a common identifier. The source-based trees on the other hand are used for data delivery. The building of the source-based tree is initiated by the receivers. It exists as long as the members belonging to that group are present. This helps in reducing the route entries maintained by routers (referred as router state). Further it reduces the concentration of traffic in different parts of the network, the processing cost of the routers and also utilizes the bandwidth efficiently, by distributing the load.

ACKNOWLEDGEMENTS

I am grateful to my thesis supervisor, Dr. Dheeraj Sanghi for providing me an opportunity to work in my field of interest. His guidance and support throughout the thesis work inspired me to do this work. He gave me ample freedom to work my own way. The private discussions with him and his enthusiasm for networking kept me motivated. His ideas have been of great help in exploring the areas which otherwise would have been imposiible.

I thank all my friends, and especially Kapil, Manoj, Bepari, Nihal, Professor, Girish, Srikar and Gopi, for making my stay at IIT Kanpur a memorable experience. I thank Kshitiz and Atul for many heated conversations.

I thank my family for supporting me at every step and encouraging me to continue with my studies. I am grateful to all of them for their efforts in building my career.

I would also like to thank God for being kind to me and driving me through this journey.

Contents

List of Figures

Chapter 1

Introduction

There are three modes of packet delivery. They are unicast, broadcast and multicast. Unicast refers to the delivery of packets from a source to one destination. Broadcast refers to the delivery of packets to all the nodes within a particular network. There is one more possibility which lies between these two cases. This is the delivery of packets from a source to a group of receivers. This is called multicasting [17]. Figure 1.1 shows the three modes of packet delivery. In this figure host H1 unicasts a packet towards host H4. The packet travels the unicast path H1-R1-R2-R3-R4-R9-R11-H4. Further, host H1 broadcasts the packet to all the hosts present in network 1. For multicasting it has to deliver the packet to hosts H2 and H3. The part of the tree traversed by the multicast packet is covered by routers R1, R2, R3, R4, R5, R6, R7, R8.

With the advent of multicasting many applications have emerged in the Internet. These include Videoconferencing, Distributed Interactive Simulation (DIS) [32, 36] etc. Multicast in these applications can be either point-to-multipoint or multipoint-to-multipoint. This characterization of the applications is based on the fact they distribute data among multiple network hosts. A point-to-multipoint or one-to-many multicasting is the flow of data from a single source to many receivers simultaneously. In multipoint-to-multipoint or many-to-many multicasting, there are multiple sources. It can be viewed as several one-to-many data flows simultaneously.

1.1 Multicasting In IP Networks

Multicasting allows transmission of an IP datagram to a set of hosts that form a multicast group, where the members of a group can be spread across distinct physical networks. It utilizes the same best-effort delivery semantics as other IP datagram delivery.

Membership to a multicast group is dynamic, meaning that it is at the discretion of a host to join or leave a group at any time. The joining of a host to a group is subject to security (if implemented). This implies that before joining a group, the host may have to authenticate itself with some server. A host can be a member of any number of groups at the same time. If a host is a member of a particular

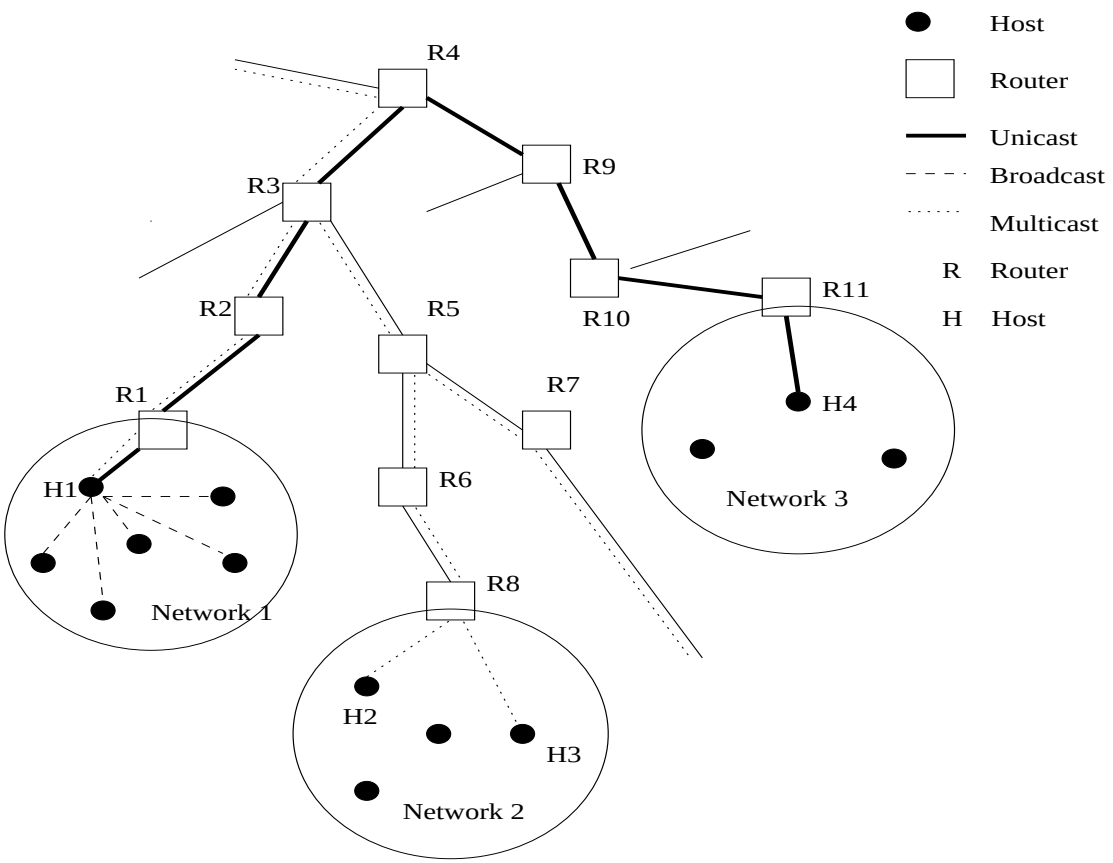


Figure 1.1: Modes of packet delivery

multicast group, it will receive all datagrams sent to that group. But to send a datagram to any group, it is not necessary for the host to be a member of that group. The host can be configured in three ways. Firstly, it can have no support for IP multicast. Secondly, it can have support for sending of datagrams but cannot receive multicast IP datagrams. Lastly, it can have full support for IP multicasting. In this mode, it can both send and receive the multicast datagrams.

Each multicast group has a unique multicast (class D) address. The class D address is identified by the first four bits which contain 1110. The remaining 28 bits are used for the identification of multicast groups. There is no fixed format as to how these 28 bits are to be specified.

Some IP multicast addresses are permanent and assigned by the Internet Assigned Numbers Authority (IANA). It is the responsibility of IANA to manage the multicast address space [27, 1]. Multicast addresses range from 224.0.0.0 through 239.255.255.255. The address 224.0.0.0 cannot be assigned to any group since it is reserved. These addresses are permanent in the sense that it is the address, not the membership of the group, which always exists [15]. A *permanent group* may have any number of members at any time, which can be even zero. Other multicast addresses are temporary in nature and the groups which they represent are called *transient multicast groups*. They remain in existence as long as there are members present in the group. If the membership count falls to zero, they are discarded. Apart from this, many proposals for the management of multicast addresses have

emerged [31].

There are two types of multicast groups with regard to the distribution of the group as shown in Figure 1.2. Each router keeps information about the groups on per network basis. Within a network, there can be several members belonging to the same multicast group. But as long as there is a single member belonging to any multicast group inside the network, the router maintains information about it.

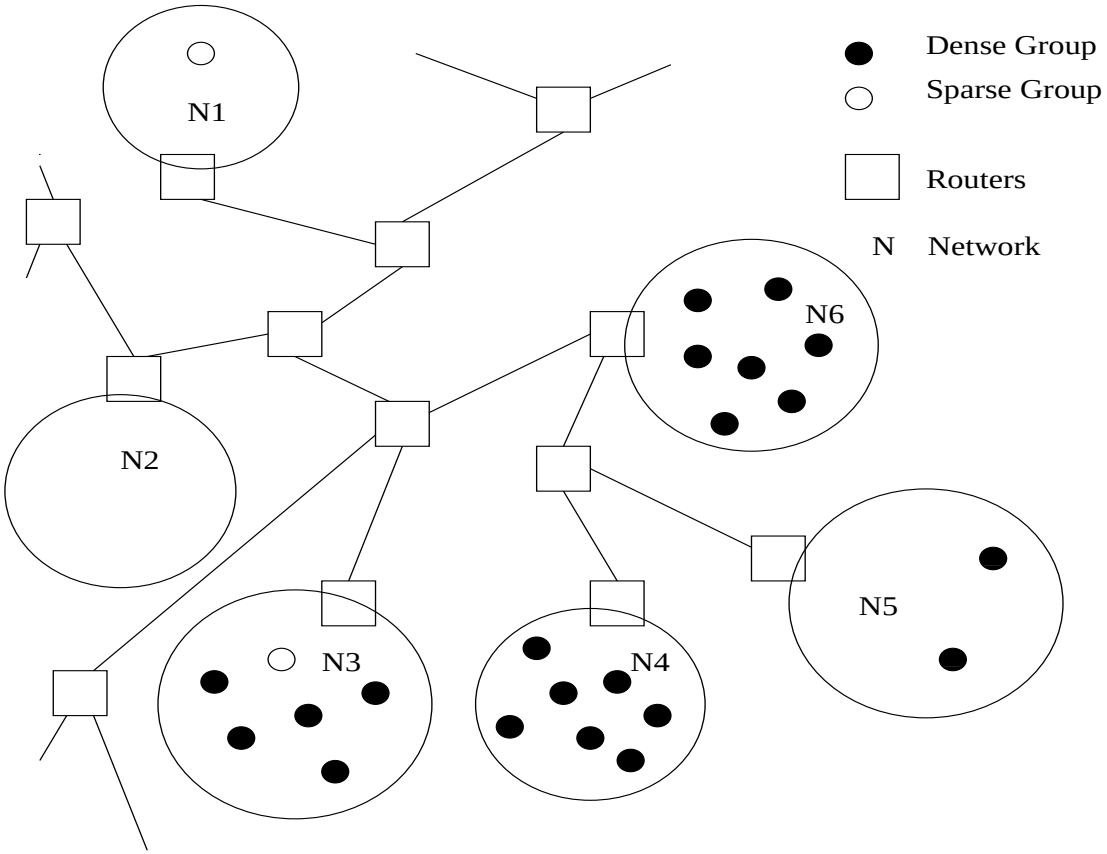


Figure 1.2: Types of multicast groups

If the distribution of the group is such that the number of networks involved is small, we call it a **sparse group**. On the other hand, if the distribution of the group is such that a large number of networks are involved, we call it a **dense group**. In Figure 1.2, we see that dense group is spread across networks N3, N4, N5, N6, whereas the sparse group is present only in networks N1 and N3.

IP multicasting may be used on a single physical network or throughout the Internet. All the routers in the Internet are not multicast capable. Routers which support multicasting are called multicast routers. These routers communicate among themselves through standard protocols and deliver the datagrams from the source to the destination. However, hosts inside the network are not aware about the multicast routers. The host transmits the datagrams using the local network multicast capability. The multicast router on receiving the datagram consults its internal *multicast routing tables* and then decides to forward the datagram to the appropriate outgoing interfaces.

When a host decides to join a particular multicast group, it conveys this information to the local multicast router i.e. the router of the network to which the host belongs. This router on receiving the membership information builds up the entry for this group and then propagates this information to other multicast routers across the Internet in order to establish the routes. For the local multicast router to gather multicast group membership information from within the network it implements Internet Group Management Protocol (IGMP) [24, 12]. When a host joins a group, it uses IGMP to send JOIN_REQUEST to the *all-hosts multicast address*. Further, due to the dynamic nature of groups (hosts may join and leave the group at any time) the multicast router uses IGMP to periodically check for the presence of groups inside the network. If the groups are present, it updates the corresponding entry but if after a certain number of polls, no membership information is obtained for a particular multicast group, the corresponding router entry is discarded. The routers are only concerned about the presence of multicast groups inside the network, no matter how many sources are transmitting to the group.

1.2 Requirements For Large-Scale Multicasting

Multicast applications typically have stringent Quality of Service (QoS) requirements. By quality of service we mean that each application expects a certain level of service from the underlying network. The network is characterized by multiple metrics such as bandwidth, delay loss probability etc. For instance, consider a multicast application which requires less end-to-end delay in delivering the data from the source to the destination. But if the underlying network is such that it incurs large delay, then it is not able to provide the desired service to the application. In this thesis we will not discuss issues like how to reserve resources, how to reduce the amount of congestion in the network, etc. What is really important to us is the requirements of large-scale multicast applications [5]. We are building a multicast protocol which makes efficient use of the existing network resources.

For example, consider videoconferencing. It requires that the network should support high data-rate, the end-to-end delay of data delivery from the source to the destination should be moderate and the delay variance should be low. But providing QoS is beyond the scope of multicast protocols. Further the establishment of the group should be done prior to the delivery of data, members can join the group at any time and as far as the scalability is concerned, it should be scalable to large number of networks and support large number of receivers per group.

1.3 Multicast BackBONE (MBONE)

To send information to a set of receivers, the method employed earlier was to duplicate the data packets and send it to each receiver. This wasted a lot of bandwidth. Further, the processing cost of routers was also very high. To overcome these overheads, Deering [17] proposed another mode of packet delivery called multicasting. IP supports multicasting, but only a few routers were made multicast capable. This was so because the set up was such that it became difficult to convert all routers to support multicasting overnight. To support multicasting, a virtual network called MBONE was developed to run on top of the physical Internet [20, 35]. As a first experiment on MBONE,

IETF conducted its first two *audiocast* in which live audio and video were multicast from the IETF meeting site to destinations all around the world.

There are a large number of networks in the Internet which directly support IP multicast. Each such network has a router, called IP multicast router (A router supporting IGMP and one or more multicast routing protocols). This router provides connectivity to the outside multicast network. This router runs an instance of *mrouted* multicast routing daemon [16]. The routers of different networks are interconnected among themselves through virtual point to point links, called *tunnels*.

The multicast router on the source end of the tunnel encapsulates the datagram and then forwards it. Encapsulation means that it prepends another IP header, with the destination address as the unicast address of the multicast router at the other end of the tunnel. The intermediate routers (the routers which lie on the path from the source to the sink of the tunnel) view this packet as normal unicast datagram and forward it according to the information in their unicast routing tables. The destination multicast router at the other end of the tunnel on receiving the datagram strips off the outer encapsulated header and then forwards the packet as appropriate. For this router, the datagram appears to come from one of its neighbor multicast router. The intermediate path taken by the datagram is hidden from this router. This path is just like a tunnel which has only two openings, one for entry and the other for exit. The entire procedure is depicted in Figure 1.3. In this figure, the router R2 in order to forward a multicast packet to router R5 encapsulates it and then forwards it to router R3. This multicast packet travels the unicast path R3-R7-R8-R5. But for router R5 it appears to come from router R2.

1.4 Multicast Scalability

Many proposals for improving the scalability of multicasting have been made [3, 4]. But we are primarily concerned with the routing aspects of multicast scalability. From this point of view, we consider the various network resources being consumed by the multicast routing protocols.

We have seen that routers support both unicast and multicast routing. For unicast routing, the routers maintain information about the individual networks. Each entry in the routing table is short. It contains a metric associated with each network which gives a measure as to how to reach that network. On the other hand, for multicast routing, the routers maintain more information. They not only have to keep information about the individual networks but also the multicast groups. Each entry specifies the source sending to the multicast group, the multicast group address, the interface on which the data arrives and the interfaces on which it has to be forwarded. It also has to maintain the state of each interface with respect to the presence of groups. In addition to this, some routing protocols require timers associated with each entry. Thus, we see that multicast routing requires much more resources (i.e. router memory) than unicast routing. Further in unicast routing, the router maintains one entry for each network but in multicast routing, same network can have several entries, each for a different multicast group. The multicast groups can be spread across the entire Internet. With every appearance of a new multicast group inside a network, the number of entries in the router increases. As the group size becomes large, the router memory usage is multiplied which is not the case in unicast routing. Thus, the usage of network resources is one of the key factors in determining the scalability

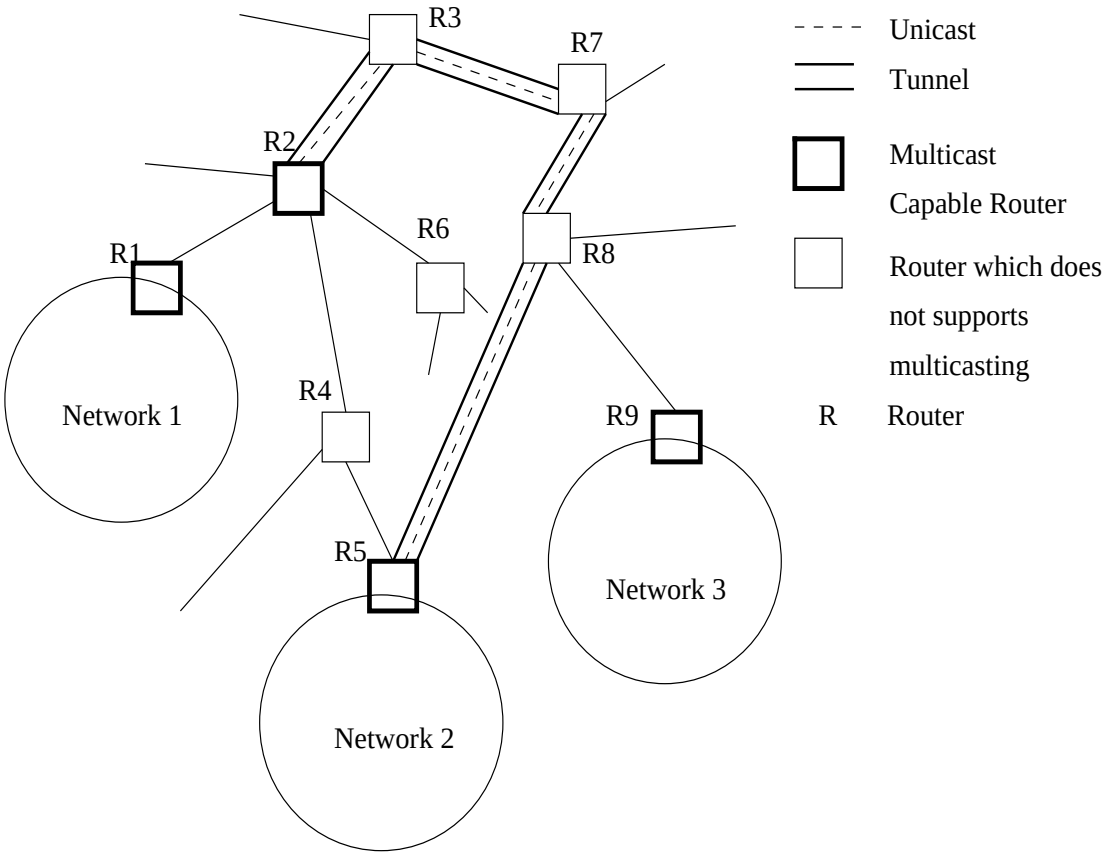


Figure 1.3: Tunnelling

of multicast protocols.

Presently MBONE [20, 14] provides support for multicasting. As the popularity of MBONE is increasing, many new multicast applications are emerging, each application having its own QoS requirements. Earlier there were only a few receivers or members involved in a particular session but with the growth of MBONE, more and more people are becoming aware, as a result of which the member set is increasing at a fast rate. The increase in member set from a small set to very large groups further demands more resources from the underlying network. Further, the utilization of resources depends on the distribution of members across the Internet. There are many problems faced in scaling the existing structure to large groups with regard to the resource utilization in the Internet. It has been observed that in going from small to large groups, a lot of network resources are being wasted which if utilized properly can increase the efficiency of the Internet. In our dissertation we have analyzed the existing protocols from the point of view of scalability.

There are many factors to be considered while measuring scalability of multicast protocols [8]. They are

- Network state maintenance by routers
- Router processing cost

- Bandwidth utilization and efficiency

Besides these the protocol complexity i.e. whether the method employed by the protocol is easy to implement or is very complicated in nature, sender set size, end-to-end delay and the wide- area distribution of group members also have an impact on scalability.

1.5 Our Contribution

Our goal is to design a multicast routing protocol which scales well and incurs less cost (with regard to the network resource utilization) as compared to the existing multicast routing protocols. We have studied the existing multicast protocols. We have analyzed each protocol to determine its strength and weaknesses. Based on this study we have proposed a new multicast routing protocol. We have compared our protocol with other protocols and showed that our protocol can support large groups efficiently.

With the wide spread of Internet, new collaborative computing tools are emerging where *Video-conferencing* plays an important role [38, 39]. This is the reason we selected videoconferencing as our sample application. The model which we are considering is that there is one source, or at most a few sources with a large number of receivers in the group.

1.6 Organization Of The Report

The rest of the report is organized as follows. In **Section 2**, we review the work that has been carried out in this field. We present a survey of existing multicast routing protocols and discuss each protocol's strengths and weaknesses.

In **Section 3**, we discuss the need for a new protocol and provide a design for a new multicast routing protocol.

In **Section 4**, we present an analysis of our protocol, including a comparison with the other existing multicast routing protocols.

Finally, we conclude the report in **Section 5** and give suggestions for future work in this area.

Chapter 2

Related Work

In the literature, several multicast routing protocols have been proposed [19]. Broadly, these protocols can be classified into two types, **dense mode protocols** and **sparse mode protocols** .

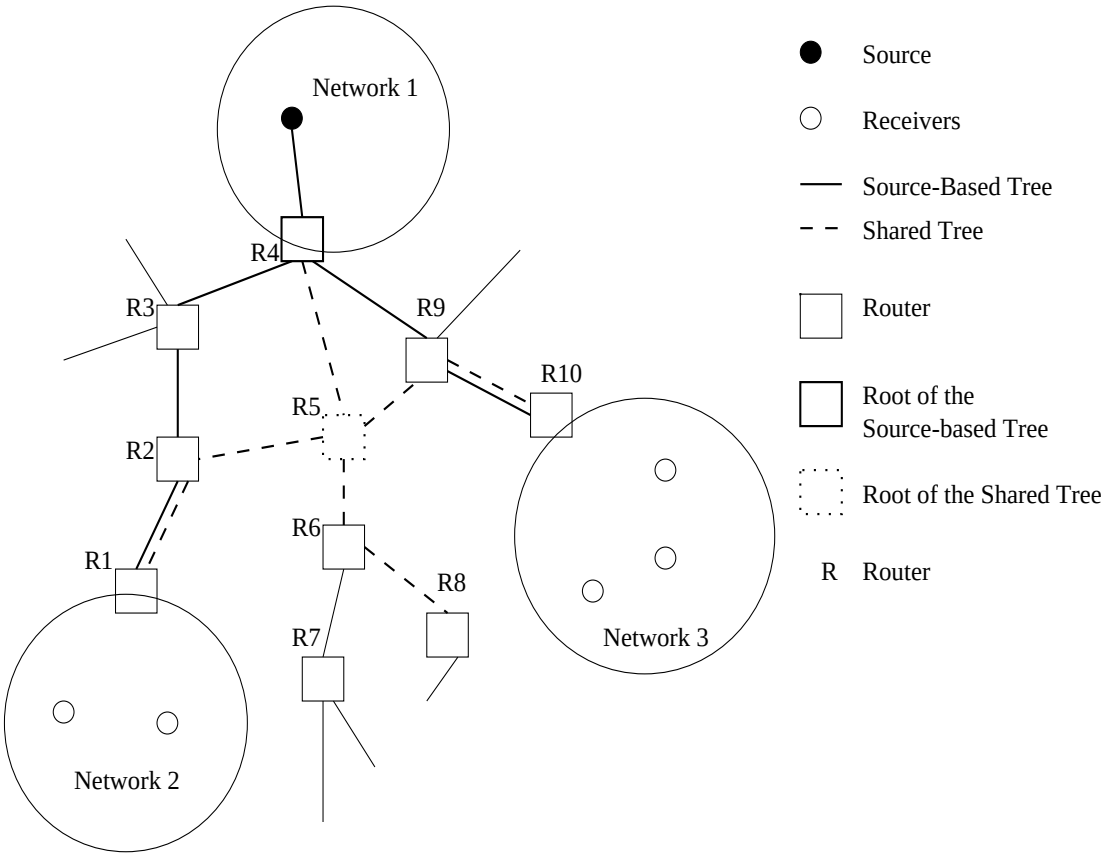


Figure 2.1: Source based and shared trees

The purpose of the multicast routing protocols is to help the routers in building route entries which connect all the members belonging to a multicast group. A separate entry is created for each multicast group. The entries in routers form a logical tree. Each router with an entry for a group represents a vertex of the tree. The two adjacent routers form an edge of the tree. The protocols differ in the type of trees which they build, and the method that they employ for building these trees. There are basically two types of trees i.e. *source-based trees*, in which the root of the tree is located at the network which contains the source (host transmitting the datagrams), and *shared trees* [13], in which the source networks (i.e. those networks which contain the sources) depend on some selected routers in the Internet, which act as the root of the trees for all the multicast groups which they support, for the delivery of datagrams. Both the trees are shown in Figure 2.1. The tree rooted at router R4 with the left child as router R3 and the right child as router R9 is the source-based tree. Router R5 acts as the root of the shared tree. Its downstream children are routers R2, R4, R6 and R9.

Next we study the various protocols.

2.1 Distance Vector Multicast Routing Protocol (DVMRP)

DVMRP supports the formation of source-based trees [34, 33, 37] i.e. the tree is rooted at the network which contains the source sending the data. The routers maintain a consistent view of the underlying unicast topology by periodic exchange of route reports which contain unicast routing tables. All downstream routers convey to the upstream router that they will use the upstream router as the next hop while unicasting data towards a particular source. As a result of these, the upstream router maintains a table of pairs i.e. the source network and a list of dependant downstream routers. This technique is called *Poison Reverse Technique*. For example, consider the topology of Figure 2.2. The routers R2, R3 and R4 tell the router R1 that while unicasting data towards the source (i.e. network N1) they will use it as the next hop towards it. The router R1 maintains this information in a table. The formation of trees is *data-driven*. A router starts building the router entries for any (source, group) pair only after it receives the first datagram belonging to this pair. In the beginning, the source starts forwarding the data on all outgoing links without any knowledge of the topology of the group.

Each router on receiving the packet belonging to a particular (source, group) pair performs a check whether the packet has arrived on the same interface which it uses to forward unicast data towards that source. For example, in the topology of Figure 2.2 router R3 on receiving a data packet from S1 checks whether it receives it on the interface on which it is connected to R1 (the first router towards the source S1). This check is called *Reverse Path Forwarding*. If the check succeeds, the packet is forwarded to a list of interfaces obtained through *Poison Reverse Technique*, and those which have directly attached networks with group members. The router also creates an entry in its internal tables. A typical entry consists of the source network address, the multicast group address, the incoming interface identifier, the forwarding or outgoing interface list and a few timers such as the age timer for the entry, the prune timer etc. This way the datagram is flooded to the entire network. Now, the *leaf routers* i.e. those routers which have no downstream multicast routers and no directly attached networks with group members, on receiving the initial datagram sends a prune message for

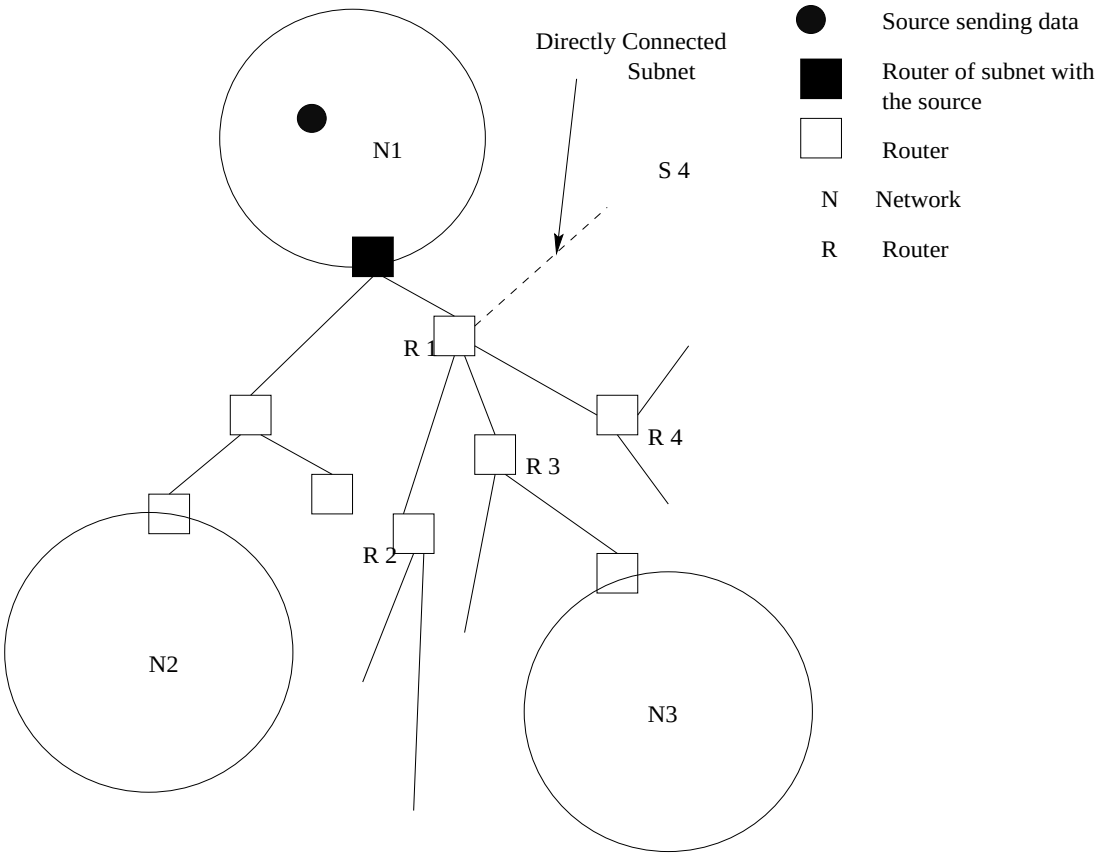


Figure 2.2: A sample topology

this (source, group) pair upstream. On receiving a prune message, the interface-id on which this prune message is received is deleted from the router entry. If there are no directly attached networks or downstream dependant routers for this (source, group) pair, then this router can also be pruned. It will forward the prune message upstream. This way the branches of the tree are pruned off. Pruning has been made a necessity in DVMRP employed over MBONE [25]. Again consider the topology of Figure 2.2. Suppose that R3 and R4 send a prune for a (source, group) pair towards R1. On receiving these prunes R1 checks whether any members are present on the directly attached network (i.e. S4) or on any downstream dependant router (i.e. R2). If no member is present it itself sends a prune upstream.

There is a timer associated with prune state in a router. After the timer expires, the prune information is deleted, and any future packets will be forwarded on all outgoing interfaces. This is done to account for the fact that group membership is dynamic and the members can can join the group at any time. On receiving these datagrams, the downstream router has to send a prune message again in order to disable the branch. Thus there is an alternate period of flooding and pruning.

When a new network wants to add itself to the tree, the corresponding router would not send a prune message next time. However, if the prune timer is large, this will delay the addition. To join the group immediately, there is a provision of *graft message*. This message is sent to the upstream

router, towards the source. Each upstream router acknowledges the graft message and then further sends a graft upstream if it itself is in the prune state for that particular (source, group) pair. This way the branches are added to the tree.

2.1.1 Protocol Analysis

We notice that there are four messages or packets involved i.e. the Neighbor Probe message (exchanged by routers to check their neighbor's liveness), Route Reports, Prune packet and Graft packet. Of these four packets, the first two have always to be exchanged even if there was no existing group. They are needed for unicast routing. Therefore we study the routing protocol cost due to the prune and graft packets only.

Initially all the routers receive datagrams and the forwarding cache (i.e. cache containing the router entries which is used by IP for forwarding the datagrams) entries are created on demand [16]. An entry is created only if the router receives a datagram with (source, group) pair for which there is no existing entry. A separate entry is created for each (source, group) pair. This is because different sources belonging to the same multicast group may use different paths to deliver data to the members of the group. Therefore, the scaling factor for the router state maintenance is O(S*G) , where S is the average number of sources for each multicast group and G is the number of multicast groups.

The number of prunes received and processed by routers depend on the following factors.

1. Number of incoming interfaces. (See Figure 2.3.)
   Incoming interfaces are the ones on which the data packets arrive. Different sources may be present on different incoming interfaces. for every source present on an incoming interface, the router has to process prunes if there are no members present in that group, downstream. In Figure 2.3, consider that for a source present on the incoming interface 1, no member is present downstream on the outgoing interface 4. Then the router will have to process a prune for this source received on this interface. Similarly, if for a source present on interface 7, no member is present on any interface, the router will have to process a prune.

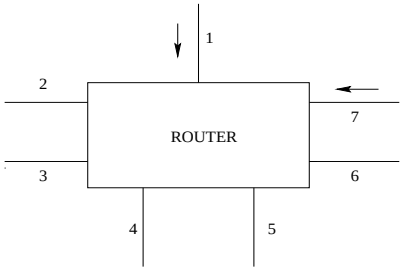


Figure 2.3: Router showing number of incoming interfaces

2. Number of outgoing interfaces per incoming interface. (See Figure 2.4.)
   If no member of a group is present on any outgoing interface, a prune will be received by the router. Similarly, prunes can be received on all outgoing interfaces if no members are present

on them. For a source present on an incoming interface 1 in Figure 2.4, no members can be present downstream on any number of downstream interfaces (6 in this case). Number of prunes processed depend on the number of such interfaces (may be 1, 2 or even 6).

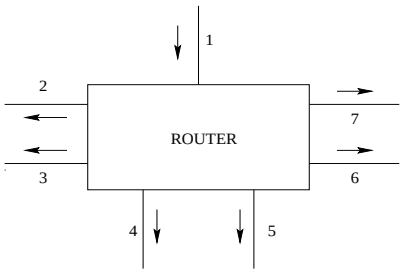


Figure 2.4: Router showing number of outgoing interfaces per incoming interface

3. Number of source networks on an incoming interface. (See Figure 2.5.)
   Since the prunes are source specific, therefore the number of prunes processed depend on the number of source networks. For example, any number of sources can be present on an incoming interface 1 in Figure 2.5. For every source present, the first two cases apply.

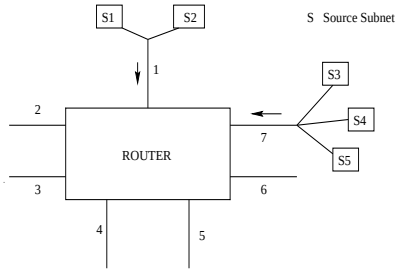


Figure 2.5: Router showing number of source networks per incoming interface

4. Number of groups per source on an incoming interface. (See Figure 2.6.)
   It may be that sources of different groups are present within a single network . Each prune is (source, group) specific, so the number of prunes processed increases as the number of groups increases. Each source network can contain sources for any number of groups. For example, in Figure 2.6, source S1 has senders for groups G1 and G2. Therefore, if no members are present for both (S1, G1) and (S1, G2) on outgoing interface 5, then prunes corresponding to both are received by the router.

5. Number of groups present but not active on outgoing interfaces per source. (See Figure 2.7.)
   For every inactive group on an outgoing interface, a prune is received by the router. By inactive group we mean that members belonging to that group are not present downstream on that interface. In the previous case, we considered that for any number of groups present on an incoming interface, the same number of groups are absent on some outgoing interfaces. Here we consider that the number of groups absent on an outgoing interface can vary. The number of such groups can be any subset of the total number of groups present on incoming interfaces.

We varied each parameter one by one keeping the others constant to realize the cost on the routers.

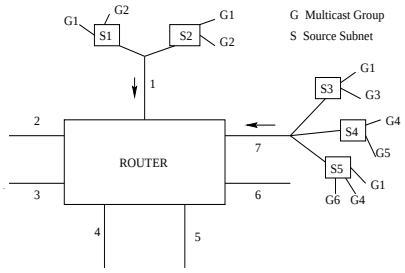


Figure 2.6: Router showing number of groups present per source network per incoming interface

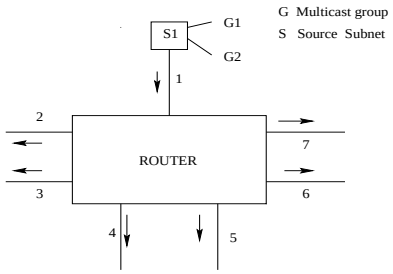


Figure 2.7: Router showing number of inactive interfaces per source network

We identified some five cases which are

- Ideal Case i.e. one incoming interface and the rest outgoing interfaces:
  We assume that only one of the interfaces of the router acts as an incoming interface on which the packets belonging to a single (source, group) pair arrive.

  Number of prunes = Number of inactive outgoing interfaces

- Now, vary the number of incoming interfaces
  We increase the number of incoming interfaces, making sure that only packets of one (source, group) pair arrive on each interface. But the set of outgoing interfaces for each (source, group) pair or incoming interface can be different.

  Number of prunes = Number of inactive interfaces * Number of Incoming Interfaces (II)

- Now, sources present on each incoming interface can also vary
  We assume that there is only one multicast group. But the number of sources present on each incoming interface can be different. The packets received on an incoming interface will be of the type $(source_1, group), (source_2, group)$, etc. Further, the set of outgoing interfaces for each (source, group) pair can be different. For example, consider Figure 2.5. The packets arriving on interface 1 will be of the type (S1, G) and (S2, G). The outgoing interfaces for (S1, G) are 2 and 5. Similarly, the outgoing interfaces for (S2, G) are 2,4 and 6. Therefore, prunes for (S1, G) are received on interfaces 3, 4 and 6 and that for (S2, G) on interfaces 2 and 5.

  Number of prunes = $\sum_{j=1}^{NumberofIIs}$ { Number of inactive interfaces * S_j}

- Now, number of groups per source per incoming interface can also vary
  Here we consider that though the number of (source, group) pairs can vary on incoming interfaces,

but for the outgoing interfaces they remain constant. In other words, the same number of (source, group) pairs are either present or absent on an outgoing interface. For example, consider Figure 2.6. The packets arriving on interface 1 will be of the type (S1, G1), (S1, G2), (S2, G1) and (S2, G2). Let (S1, G1) and (S2, G2) be present on interfaces 2, 3, 4 and 5. Similarly, let (S1, G2) and (S2, G1) be present on interface 6. Therefore, the prunes for (S1, G1) and (S2, G2) will be received on interface 6 and that for (S1, G2) and (S2, G1) will be received on interfaces 2, 3, 4 and 5.

Number of prunes = $\sum_{k=1}^{NumberofIIs} \sum_{j=1}^{S_{ik}}$ { Number of inactive interfaces * G_{jk}}

- General Case i.e. number of groups per source on each inactive outgoing interface can vary
  This case represents the full dynamic behaviour of the router. The number of prunes processed depend upon the nature of the groups with respect to each outgoing interface.
  Number of prunes = $\sum_{k=1}^{NumberofIIs} \sum_{j=1}^{Sik} \sum_{m=1}^{NumberofOIs} G_{jkm}$

Thus we see that the number of prunes being processed in the most general case depends upon the presence of groups downstream. If the groups are so distributed such that every outgoing interface of every router is utilized, then the number of prunes received and processed by the routers will be zero. Also no bandwidth will be utilized by the prune packets. On the other hand, if the group is very sparsely located, then much cost is incurred by those parts of the network which do not have group members. The off-tree routers (Routers can be classified into two types. Those which forward the datagrams to the members downstream are called on-tree routers. Routers which do not forward datagrams but maintain router entries for the multicast groups are called off-tree routers.) will have to maintain prune state and process the timers associated with this state. Further, bandwidth is wasted due to the periodic flooding of datagrams. For example, consider the topology of Figure **??**. If the group is so distributed that all the outgoing interfaces of all the routers from R1 to R12 have members downstream, then the routers do not maintain any prune state. On the other hand, if the group is distributed only in that part of the network depicted by double lines (i.e. R1, R2, R3, R6 and R9), then a large number of routers (R5, R7, R8, R10, R11 and R12) will have to bear the cost of maintaining the prune state. If the nature of the group is very dynamic, then for each source specific prune, a graft has to be sent upstream which further adds to the bandwidth utilization and router processing cost. Number of prunes and grafts depend on the number of (source, group) pairs present in the worst case.

For example, consider the case of MBONE. At present there are about 20,000 networks in the MBONE. If we consider a group of spread across 1000 networks, then the ratio

Number of networks with group members : Number of total networks = 1:20

We see that about 95 percent of the network incurs extra cost to support a multicast group which has membership in only 5 percent of the network. Therefore, DVMRP is not suitable for sparse groups.

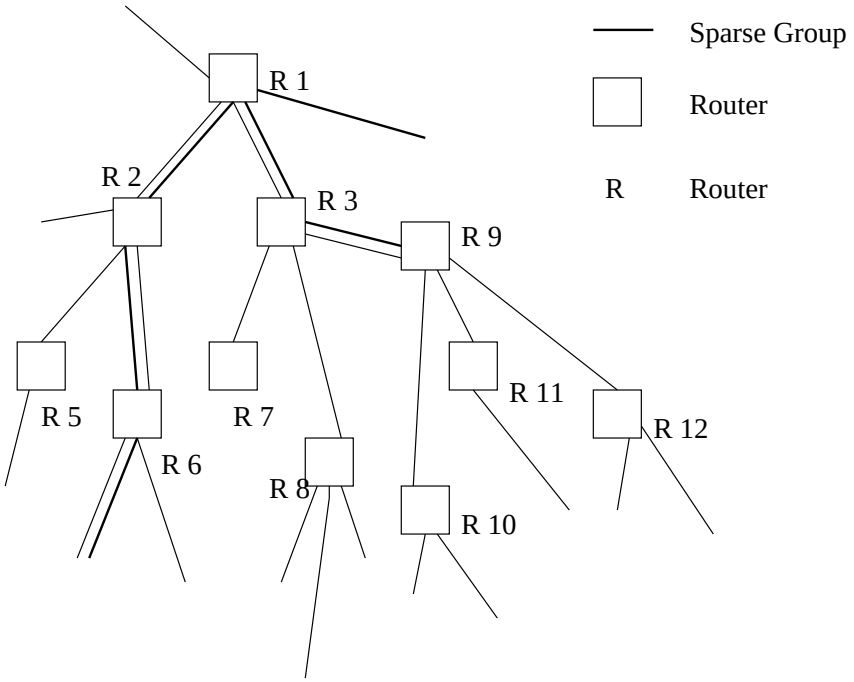


Figure 2.8: disgroups Dependence of prunes on the distribution of groups

2.2 Multicast Extensions to OSPF (M-OSPF)

Unlike DVMRP which is a distance-vector protocol, MOSPF [30] is a link state protocol. It supports source-based trees. The former refers to the set of protocols in which the routers maintain a list of destination networks along with the distance to reach that network. This list is built from the routing messages (containing routing tables) received from the directly attached routers. The distance is the shortest of all the distances reported by the neighboring routers (routers which share a common link) to reach the destination network. This distance can be any metric. For example, one of the metrics can be the number of hops. It is defined as the number of routers that a datagram passes through along the path from the source to the destination.

In link state protocols, each router maintains a complete knowledge about the entire topology. Each router periodically determines the status of all the neighboring routers (whether communication is possible with them or not). It then floods this information to the entire domain. This message is received by all the routers within the domain. Each router then updates its view of the topology. To reach any destination network the router first computes the routes by using Dijkstra's shortest path algorithm.

Before going through the protocol let us define two terms. An Autonomous System (AS) is a set of networks and routers which are controlled by an administrative authority. An AS can be further divided into sub-parts called areas.

Each router in an area maintains a local group database which contains information about the

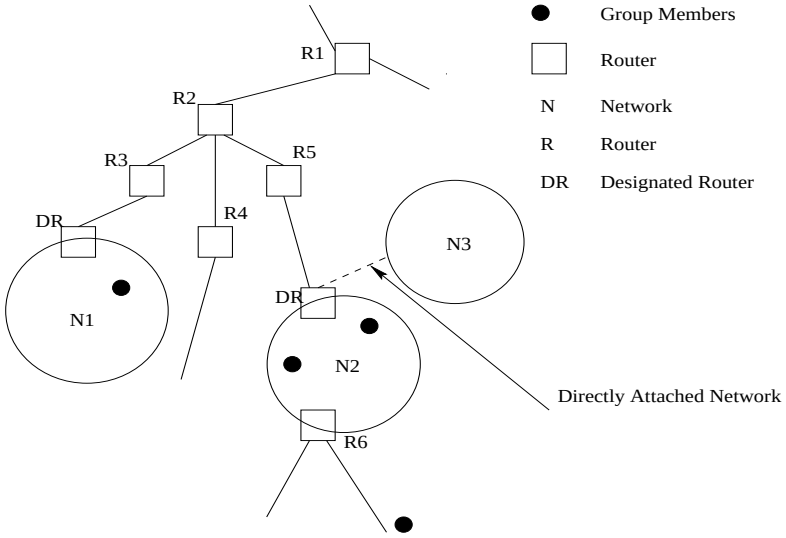


Figure 2.9: A sample topology

groups present on all the attached networks for which this router acts as *Designated Router* (DR). Normally, the DR sets up multicast route entries and sends corresponding JOIN messages on behalf of directly connected receivers and sources, respectively. The DR may or may not be the same router as the IGMP Querier. The entries in the local group database are indexed on (multicast group, attached network). Each router floods Group-Membership-LSA (Link State Advertisements) for each multicast group in its group database, throughout the area. These advertisements list the router which acts as the DR for the network with group members, transit networks but not the stub networks. Transit networks are those networks which are used to forward datagrams to downstream networks. For example, in Figure 2.9 the network N2 is the transit network. Stub networks are those networks which do not have group members present inside them or if they have group members then they don't act as transit networks for downstream members. In Figure 2.9, the network N3 is a stub network. The routers which receive these advertisements keep a record in their link-state database (maintained by OSPF [29]). The forwarding decision for the data packets is based on the contents of a data forwarding cache. This cache is built with the help of local group database and the shortest-path tree constructed from the link-state database. The tree constructed is data-driven. This means that the entry is created only when the first datagram for a (source, group) pair is received. When the router receives the first datagram for a (source, group) pair, it builds the shortest-path tree for this particular source using the Dijkstra's algorithm. Then it identifies its position on the tree.

The position of a router on the tree is specified by the upstream node and the downstream interfaces. Consider the topology shown in Figure 2.9. The position of the designated router of network N2 is specified by the upstream router R5 and downstream interfaces which have networks N2 and N3 attached to them. Using its position on the tree and the local group database it builds a forwarding cache listing the incoming and outgoing interfaces. Group-Membership-LSAs are sent or flooded periodically. Whenever there is some change in the area's topology, it is reflected in the LSAs. As a result, the routers clear their entire forwarding cache and the whole process of building the cache is repeated.

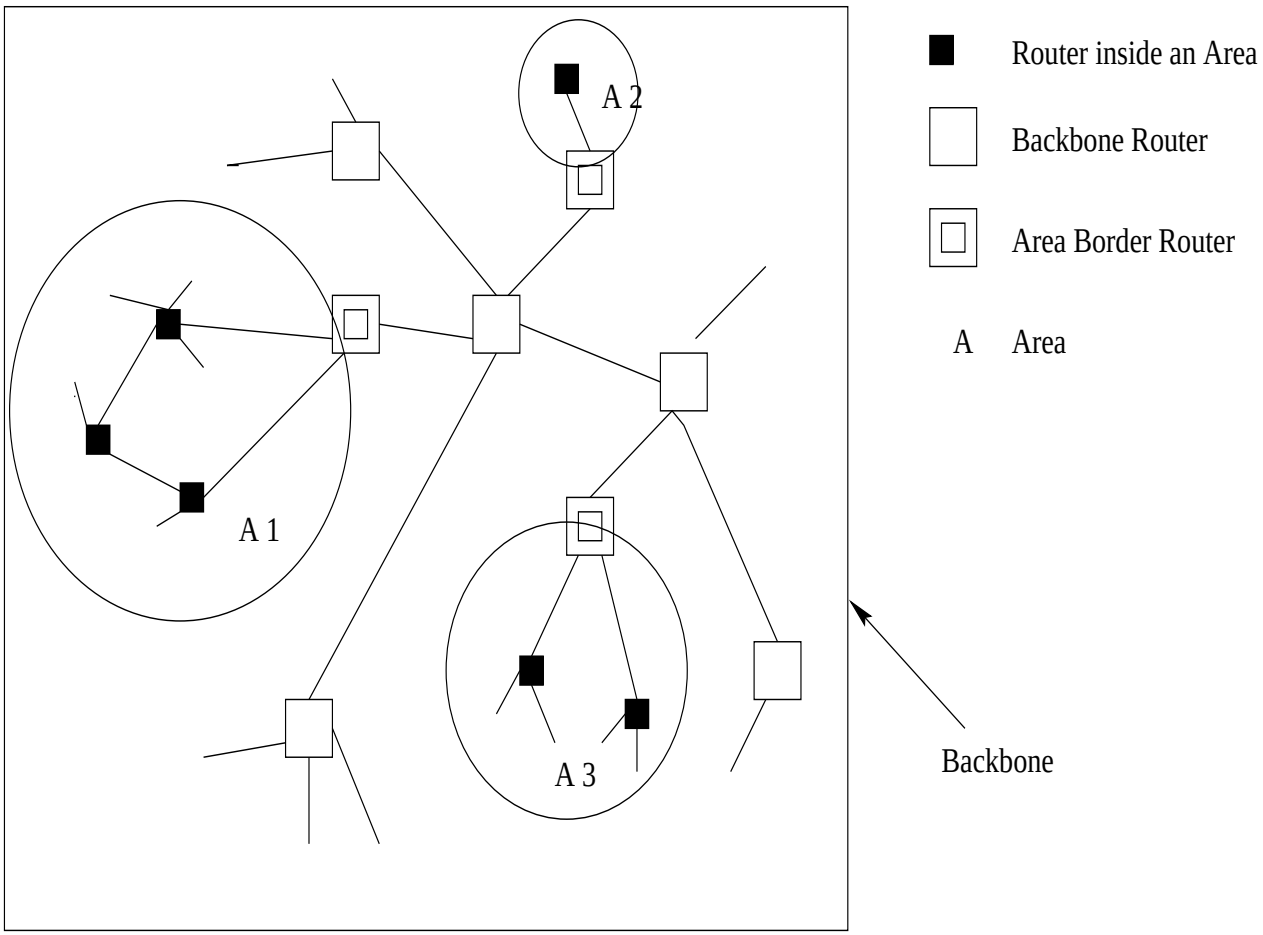


Figure 2.10: Inter area forwarding

Each IP datagram header includes a field that allows the sender to specify the type of service desired i.e. low delay, high throughput and high reliability. It is a hint to the routing algorithm that helps it choose among various paths to a destination based on its knowledge of the hardware technologies available on those paths. The router may have several outgoing paths to a particular network, where each path may have a different characteristic. A path may be best suited for datagrams requiring less delay but it may incur huge data loss. On the other hand a path may be very reliable. Depending upon the value of the TOS field, the router may choose a particular path. If the version of MOSPF also supports *Type Of Service* (TOS) forwarding, then separate trees are built for each TOS value.

For inter-area case, a few Area Border routers act as inter-area multicast forwarders and also as wildcard receivers. They determine what all members are present in the area and inject a Group-Membership-LSA into the backbone area. The backbone area having received LSAs from all non-backbone areas has complete knowledge of the groups present in each area. Since the area border routers act as wildcard receivers, they receive all the data packets originating within the area which they forward to the backbone area. Now it is the responsibility of the backbone area to deliver the data packets to all those areas which have members belonging to this group. For example, consider the topology shown in Figure 2.10. All the areas (A1, A2, A3) advertise their LSAs to the backbone.

Thus the backbone has complete knowledge of various groups present inside each area. The double boxes are the area border routers. Now, the area border router of area A1 acts as a wildcard receiver. For all the sources inside area A1 it forwards the datagrams to the backbone. The backbone routers having complete information about the distribution of groups further forward these datagrams only to those areas which have members belonging to them.

2.2.1 Protocol Analysis

The forwarding cache entry is indexed on (source, group) pairs [28]. The state maintained by the router increases by the order O(S*G), where S is the average number of sources in each multicast group and G is the number of groups. Further if it supports TOS forwarding, then the state or required memory size is further multiplied by the number of TOS values supported. But not all routers have to maintain this state since the forwarding cache building is data-driven and once the router finds that it does not lie on the tree for this group it stops forwarding the data packet.

The major cost of this protocol is due to the flooding of Group-Membership-LSAs and the Dijkstra's shortest-path tree calculation for each source, performed by the routers. The storage required for the maintenance of group membership information as provided by membership LSAs is one of the reasons that this protocol does not scales well. The flooding of membership LSAs consume a lot of bandwidth. Further any change in the topology of the AS requires the router to restart the tree building process. But not all routers are burdened with the task of tree calculations due to the fact that routers stop forwarding data packets once they find that they are not on-tree.

The router processing cost with regard to the shortest-path tree calculations is the product of the average number of sources per multicast group, number of groups and number of TOS values.

MOSPF is so obviously poor compared to DVMRP that we wont consider it any further.

2.3 Core Based Trees (CBT)

In CBT, the tree formed is the shared tree [10, 8, 9, 7, 6]. A set of routers are designated to act as cores, where one is the Primary Core and the others Secondary Cores. The Primary cores can be different for different groups.

The tree building process starts when a group member appears on any network . The network's Designated Router (DR) on receiving the group-membership reports (or a join request from the host) sends a JOIN REQUEST packet towards the target core (primary or secondary). The information about the target core is statically configured. This join request travels hop-by-hop towards the target core. The core confirms it by sending a JOIN ACK message. The Ack message travels the same route downstream (or towards the network's DR) as the request message travels upstream (i.e. towards the core). The request message creates a temporary state in all routers along its path. This state or entry is made permanent by the ack message traveling the same path in the reverse direction. A typical entry contains the group identifier, the parent address (the address of the router towards the

core), the interface on which it is connected to the parent, number of children and the address and interface associated with each child (downstream router). The target core, if it is not primary, then joins the primary core. The request message need not travel all the way to the target core. If the state for this group is already present in any intermediate router, then the request message is terminated at this router and the corresponding Ack is sent back. This way the trees are formed rooted at the primary core. If no members belonging to a particular group are present downstream, the router sends a QUIT REQUEST message upstream which is acknowledged (QUIT ACK) by the upstream router. This is similar to pruning except for the fact that it requires explicit acknowledgement. Further, this is not soft state. After sending the QUIT REQUEST message upstream the router removes the parent information (or the fields in the entry which give the parent address) immediately.

If any router or a link goes down, the downstream router which used this router as the next hop towards the core will have to rejoin the tree on behalf of each group present on their outgoing interfaces individually. They can perform an aggregate join if the groups share a common core-list. Each core supports a certain number of multicast groups. In other words any member joining these multicast groups can use any one of these cores. The list of cores which support a single group is called core-list for that group.

Further, during re-configuration a situation can occur where a router finds that the next hop towards its target core is the router which is downstream to it. Such a situation is depicted in Figure 2.11. Here router R1 finds that in order to join its target core it has to send a join request towards R2 which is downstream to it. R1 sends a FLUSH-TREE message downstream to teardown the tree i.e. to break the branch from R1 to R2. The downstream routers then perform explicit Rejoin if they have members present on them.

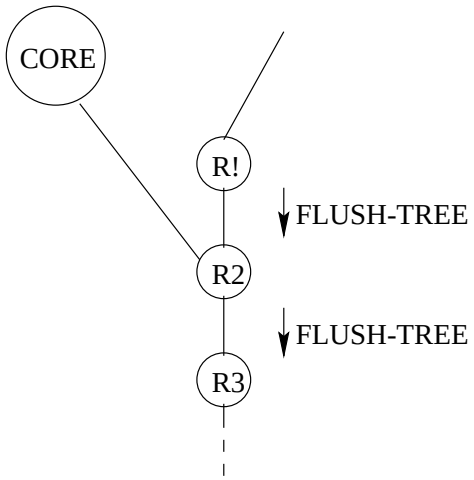
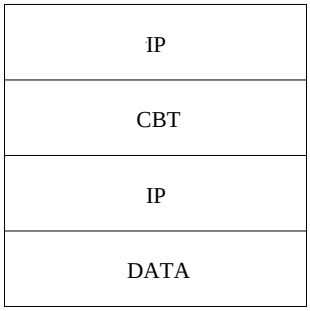


Figure 2.11: Flush tree message

The trees once built cannot be changed even if the underlying unicast topology undergoes a change i.e. the state is hard and does not reflects the underlying changes.

There are two modes of data forwarding, the native mode and the CBT mode. These modes are shown in Figure 2.13. A data packet originating inside a network is IP multicasted to all local members. When the network 's DR receives this packet, it multicasts one copy to all other CBT

routers present on this network called CBT multicasting. It also forwards one copy of datagram on all directly attached networks for which it acts as a DR called Native mode forwarding. Further, it forwards an encapsulated packet on all outgoing links. The encapsulation is as shown in Figure 2.12.



IP Internet Protocol

CBT Core Based Tree

Figure 2.12: Encapsulation by CBT routers

The downstream routers check the liveness of their parent by periodically sending ECHO REQUEST message towards the parent, which is confirmed by the parent through ECHO REPLY message. Only one such message is exchanged by routers per link. These messages help in tree maintenance.

2.3.1 Protocol Analysis

The packets involved in this protocol are JOIN REQUEST, JOIN ACK, JOIN NACK, QUIT REQUEST, QUIT ACK, FLUSH TREE, CBT-ECHO-REQUEST, CBT-ECHO-REPLY and CBT-BR-KEEPALIVE (used by border routers). We see that all these packets are required for the existence of groups. We analyze the cost due to each of them.

The router state being created by JOIN message is made permanent by the ACK message. Routers maintain a separate entry for each group. Since sources depend on a core router for distribution of data to the group members, the routers need not keep information about the source networks. Thus the forwarding cache reflects the parent-child relationship per group. From the state maintained, it is clear that it is of the order of O(G), where G is the number of groups.

Initially during the tree set-up phase a lot of JOIN messages are transmitted. But once the tree has been established this traffic dies down i.e. the bandwidth requirement is only for a short period.

Then, as the members quit from the group there is traffic due to QUIT messages. The number of such messages depend on the number of members leaving the group. This overhead is very less. Also, there is no state and timers maintained in routers for pruning like DVMRP. The router entry is discarded. No periodic refreshing is required and since the state is removed it does not result in unnecessary forwarding of datagrams. Thus the dynamic behaviour of groups does not pose much processing cost on routers. If in the future the members appear again, the grafting of tree branches is

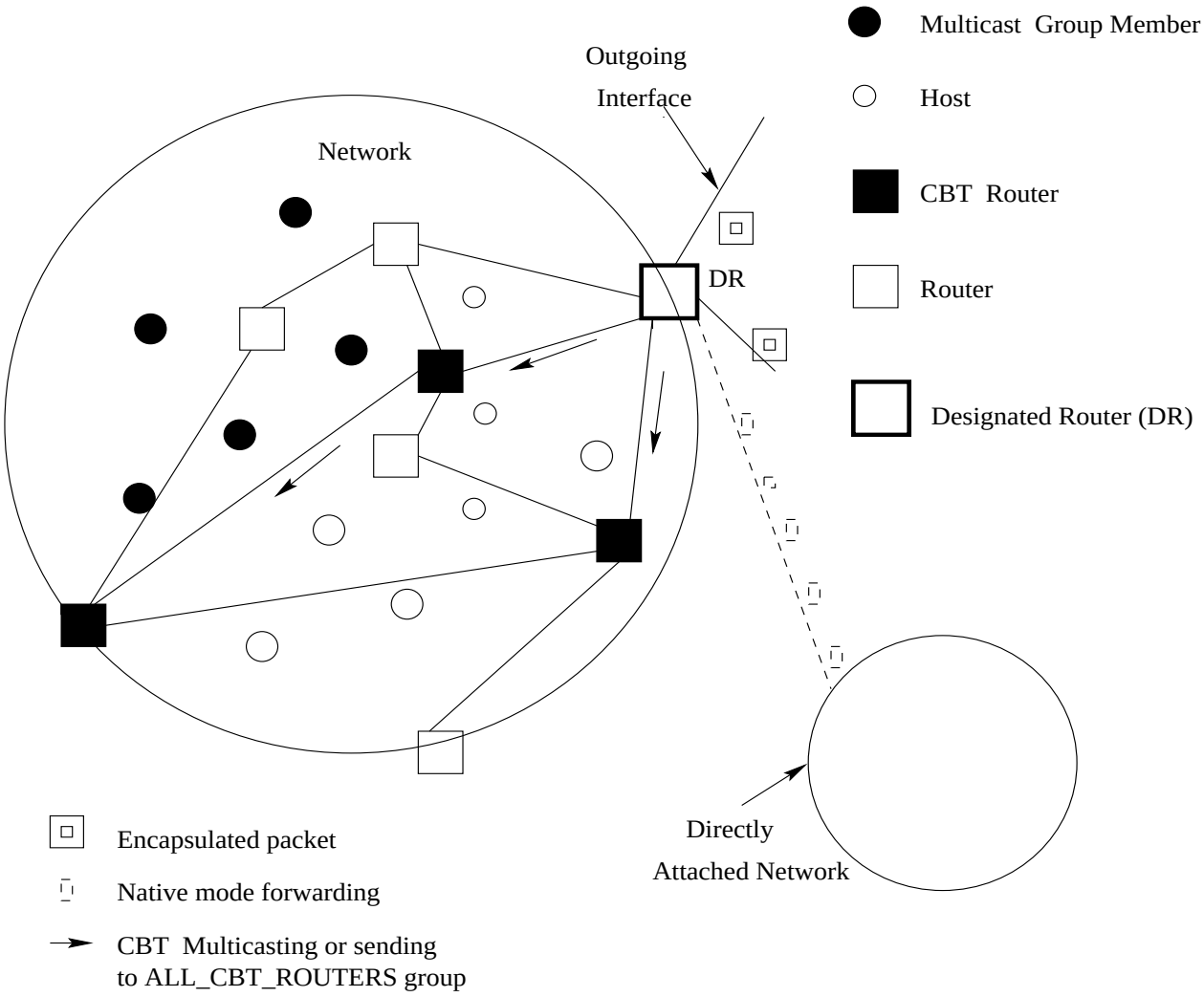


Figure 2.13: Data forwarding in core based trees (CBT)

through JOIN REQUEST and JOIN ACK. The number of such messages depend on the number of members joining the group.

In case of tree tear down, the cost of bandwidth usage and router processing due to flushing of branches and rejoining process is significant. It depends on the number of groups present downstream, but it is also short-lived.

One may note that the data and the routing protocol messages remain confined to only those parts of the network which have members belonging to different multicast groups i.e. no extra cost is incurred in that part of the network which has no members.

Since the data packets are encapsulated within a CBT header, the bandwidth required will be more than that if the packet was forwarded without encapsulation.

The major overhead comes due to the periodic exchange of KEEPALIVE messages (ECHO REQUEST and ECHO REPLY) between the routers.

Since all the trees are rooted at a single router (or a small set of routers) i.e. the primary core, the traffic concentration or link utilization will be very high in that part of the network which covers only the primary and secondary cores. Such routers will have to process a large number of packets. If these routers are not able to process packets at the rate at which they arrive, long queues are built up. This adds to the end-to-end delay.

2.4 Protocol Independent Multicast - Shared Mode (PIM-SM)

This protocol supports both the shared and source-based trees. A set of routers are designated to act as Rendezvous Points (RP) [21, 18, 23]. These routers act as the root of the shared trees being formed.

Each domain elects a Domain BootStrap Router (BSR) from among a set of routers. There is also a set of routers which act as candidate-RP routers. These routers convey to the Domain BSR a set of groups which they will support in Candidate-RP-Advertisement messages. The bootstrap router collects these massages and then sends a complete list of (RP, groups) pairs in bootstrap messages, which are forwarded hop-by-hop. The DRs on the receipt of these messages store them in an internal table. They utilize this table when looking up the RP associated with a particular group. These messages are exchanged periodically.

The tree building process starts when a group member appears on any network . On receiving a join request from any host inside the network, the network's Designated Router (DR) looks up the RP associated with this group from the table of (RP, groups) pairs maintained. It creates a wildcard entry i.e. (*, G) and sends a JOIN message towards the RP. All routers on the path create a wildcard entry for this group and forward the JOIN message to the RP. This way the tree is formed. When the host has data to send, it forwards it to the network's DR which encapsulates the data packet in Register Message (RM) and unicasts it towards the RP. The RP decapsulates the packet and multicasts it along the multicast tree. The entire process is depicted in Figure 2.14. Each DR keeps count of the number of packets received per group. When this count crosses a threshold value, it starts keeping count of the packets from particular sources belonging to this group. If the count of packets for a particular source crosses a threshold value, this DR initiates the switching over to source-based tree by sending a JOIN message towards the source. Only the RP and the last hop routers i.e. DRs can initiate the process of switching over from shared to source-based tree. Other routers just create (S, G) entries on the receipt of source specific JOIN messages. By the time the source-based tree is being formed the network initiating the switch continues to receive data on the shared tree. Then, when it starts receiving data on the source-based tree, it prunes the shared tree by sending a PRUNE message towards the RP, provided the interfaces of the two trees are different. The shared tree is used for the formation of source-based tree. When the RP has no downstream receivers for the shared tree or it has joined the source-based tree, itself it sends a REGISTER STOP message towards the source in order to stop the source from sending encapsulated data for a period of REGISTER SUPPRESSION TIMER.

The state maintained by routers is not permanent i.e. whenever there is a change in the underlying unicast topology it is reflected by performing an RPF check on all active router entries, which are updated accordingly. This means that if the unicast topology changes such that the shortest path to a particular destination also changes, the entries are updated to reflect this change.

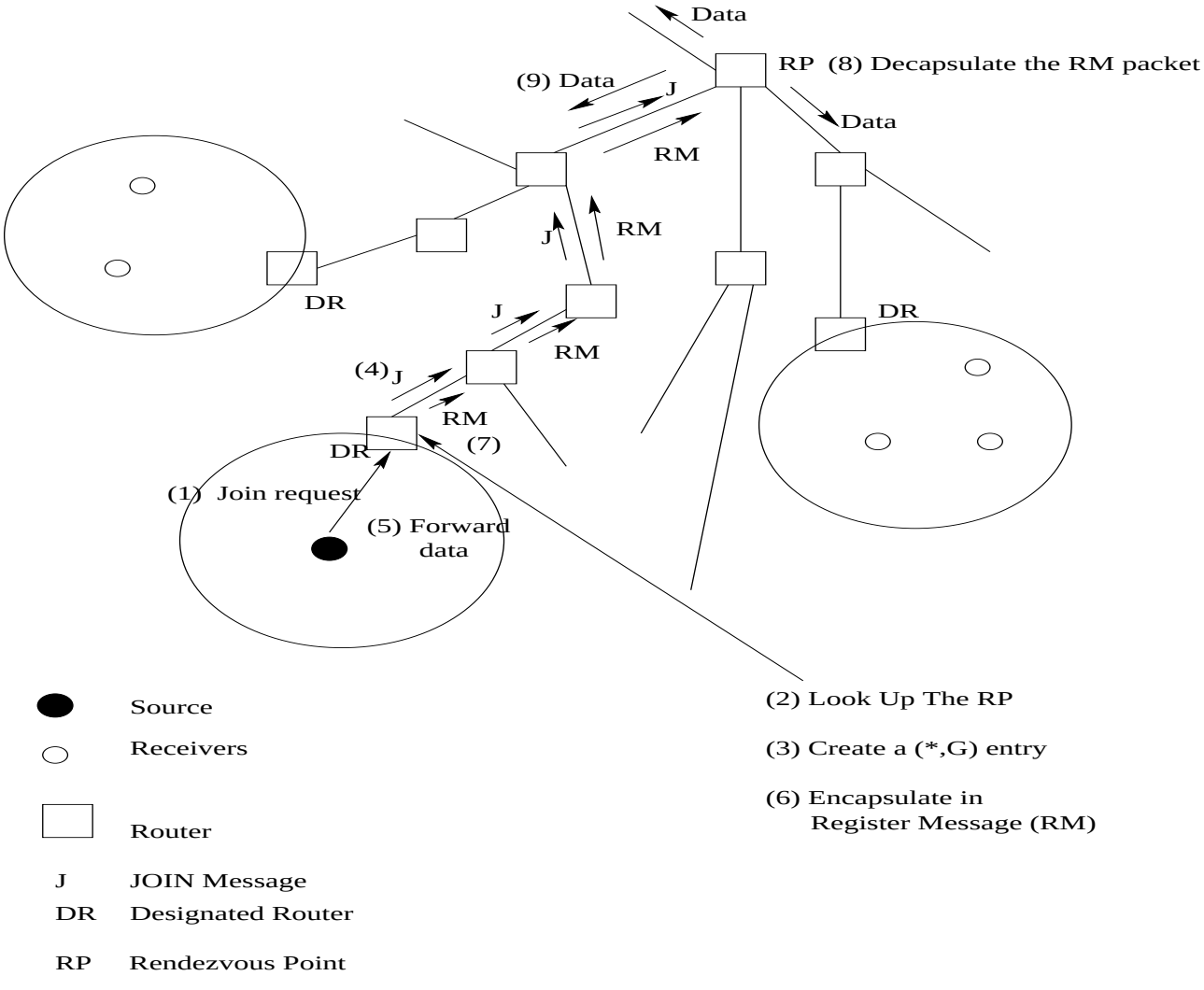


Figure 2.14: Tree formation and data forwarding in PIM-SM

There is one message which carries information about the groups to be joined and pruned simultaneously. This message contains a list of sources that have joined and sources that have pruned for a particular group address. These messages are exchanged periodically or event driven (i.e. whenever a new entry is created or the outgoing interface list goes from null to non-null or non-null to null) between adjacent routers, the direction being from the downstream router to the upstream router. They are not (source, group) specific. Rather an aggregate message reflecting the state of the router at that moment is exchanged every fixed interval.

A lot of timers are involved in this protocol i.e. those related with router entries, those related with neighbor discovery and those related with RP information maintenance.

2.4.1 Protocol Analysis

The packets involved in this protocol are HELLO (exchanged by routers for the election of DR), Register, Register-Stop, Join/Prune, Bootstrap, Assert (used in multi-access networks for the election of a single forwarder) and C-RP-Adv. The first packet has to be exchanged anyhow even if there were no groups in existence. We study the routing protocol cost due to the other packets.

Initially the routers maintain wildcard entries i.e. (*, G) along with the RP associated with this particular group [26]. Then, as the trees switch over from shared to source-based, they also maintain (source, group) specific entries. With the building of source-based trees, the routers also have to maintain state information about the individual source networks and the groups which they support. Thus the state maintained by the router increases from O(G) to O(G*S), where G is the number of groups and S is the average number of sources per group.

During the initial tree set-up phase, there is significant bandwidth requirement and router processing cost, since initially there is no existing entry for any group in the router and so aggregate JOIN/PRUNE cannot be used. For each new entry created, a separate JOIN/PRUNE is triggered. Once the trees are established, the incremental cost of establishing the trees is low.

Since the packets are encapsulated in Register Message and unicasted to the RP from where they are distributed to the group members, this adds to the end-to-end delay.

Whenever the switch-over takes place, there is bandwidth usage and router processing cost due to the (source, group) specific JOIN messages but this is also short-lived. During the switch over, duplicate data packets may exist in the network. Once the source-based trees are fully established, the source starts transmitting on them. But duplicate packets may still exist if all the receivers have not joined the source-based tree.

The important fact here is that the data and routing protocol packets are confined only to that part of the network which have the group members present and no extra cost is incurred in the remaining part of the network. For instance, the routers which do not lie on the trees do not incur the cost of processing the data or control packets as is the case in DVMRP.

Each router has a single JOIN/PRUNE timer. After the expiration of this timer, the router sends an aggregated JOIN/PRUNE message upstream. The major cost as far as the bandwidth utilization and router processing is concerned is due to these periodic messages and also other periodic messages (C-Rp-Adv etc.) and the timers associated with these messages. Thus, maintenance traffic is the major cost incurred in this protocol.

In this protocol, since a set of RPs are hashed to groups, the roots of the shared-trees and therefore the traffic is distributed. As a result of this the load on the links and routers is reduced to a great extent.

2.5 Protocol Independent Multicast-Dense Mode (PIM-DM)

This protocol is similar to DVMRP in all respects [22] except that it does not depend on the underlying unicast protocol to decide the list of dependent downstream routers. As a result, some duplicate packets may exist in the network. Otherwise, the cost incurred is the same as in the case of DVMRP.

Besides these protocols, there is another multicast routing protocol called Mobile Conference Steiner Multicast (MCSM). It utilizes close to optimal *Steiner trees* for the conference session [2]. Steiner trees are shared trees with the property that the sum of the weights of its edges is minimum. It is not easy to find such trees, but an approximation can be made. Such trees are called close to optimal Steiner trees.

Chapter 3

Design Of Our Protocol

3.1 Need For a New Protocol

We have seen dense-mode protocols like DVMRP, MOSPF and PIM-DM and sparse-mode protocols like CBT, PIM-SM.

We can broadly classify the protocols depending on two factors, one being the sparseness or denseness of the group and the other being the data rate.

We have divided the various protocols into four categories, each being assigned to the category for which it is well-suited. Source-based trees are good for dense groups. For sparse groups, there is too much overhead of tree maintenance in these trees. Shared trees are good for low-data rate. For high data rate applications, the core may become a bottleneck. Neither approach is entirely satisfactory for sparse groups with high data rate.

Data Rate	Dense/Sparse Group	Protocol
Low	Sparse	CBT, PIM-SM (Shared Tree)
Low	Dense	DVMRP, MOSPF, PIM-DM
High	Dense	DVMRP, MOSPF, PIM-DM

In DVMRP, M-OSPF and PIM-DM, the building of the tree is source initiated i.e. the source starts sending data without any prior knowledge of the group topology. The data is flooded to every part of the network. On receiving this data packet the routers create (source, group) entries. Thus, the address of the source belonging to this group is provided by the source itself. Though the downstream receivers may know the existence of group but they have no prior knowledge of sources which send data to this group. This is why flooding is used.

In shared-trees such as in CBT and PIM-SM, since the receivers have no prior information of the sources, they depend on a Core or Rendezvous Point for the delivery of data. The tree building process

is receiver-initiated.

In PIM-SM, the source-based tree is built using the shared tree. In this, since the receiver is already on the shared tree and receives data packets, therefore it has the information about the source sending to this group, which it utilizes to switch-over to source-based tree.

We cannot employ flooding, since our goal is to eliminate the cost incurred in that part of the network which has no downstream group members. For this, the building of the tree should be receiver-initiated. One approach can be that if somehow the receivers are provided with the source information, then they can explicitly join the sources.

Our protocol sets up a common signalling tree to convey to all receivers the address of the source sending to a particular group. The receivers then initiate to join the source, i.e. the paradigm is that of the formation of receiver-initiated source-based tree. The protocol supports both the shared and source-based trees simultaneously.

3.2 Protocol Overview

The protocol we have proposed is called SCAMP (SCAlable Multicast Protocol). In this, a set of routers are designated as *Critical Routers*. These routers help in the formation of shared signalling tree. In other words one of them acts as the root of the signalling tree being formed. All the hosts are required to join a critical router which puts them on the common signalling tree shared by all the groups. When a host decides to join a particular group, it sends a join towards the Designated Router (DR) of its network. On receiving this join request, the DR creates a wild card entry for this particular group i.e. (*, G), if an entry for this group does not exists already. Also, if this join request is the first ever request from inside the network, the DR proceeds to join the shared tree. It creates a (*, PG) entry where PG stands for *Permanent Group*, calculates the address of any critical router and sends a join request towards that Critical Router. This way the shared signalling tree is formed. All the DRs which have joined the signalling tree, periodically decide for which groups they have sources within the network and send this information towards the critical router, which distributes it on the entire shared tree. We call these messages as *Signal Messages*. Thus all the DRs which have joined the shared tree receive these 'Signal Messages'. On the receipt of these messages, the DR matches all the entries in the received message with those in the wildcard-entry table and if the group part of the entries match, then creates a (S, G) entry in the forwarding cache table, if such an entry does not exists already. Also, it sends a join request towards the source for which it creates a (S, G) entry. This is the receiver-initiated formation of source-based tree. (S, G) entries are also created by the DR on the receipt of data packets from within the network, but these entries do not trigger the sending of join requests. As the DRs receive the join requests, the outgoing interface lists are formed and it starts forwarding data packets on these interfaces.

3.3 Detailed Protocol Description

3.3.1 Introduction

We consider an environment in which there is any underlying unicast protocol, defining the unicast topology of the Internet. Within a network, the group members announce their membership to a group using Internet Group Management Protocol (IGMP) [24]. Further, a set of routers are designated as Critical Routers (CR) and one of them acts as the root of the shared tree being formed. The formation of trees is influenced from Core-Based Tree (CBT).

3.3.2 Hosts Joining a Group

When a host decides to join a group, G, it conveys its membership information using IGMP to the Designated Router (DR) of its network. Thus the host becomes a member of this particular group.

When DR receives a membership report for a group G, it checks whether it already has an entry for this group or not. If an entry exists, it just ignores the report. Otherwise, it creates a wildcard multicast route entry for the group which is referred to as (*, G).

3.3.3 Formation of Shared Signalling Tree

When a DR creates a wildcard entry for any group, it checks whether any other entry exists or not. If this is the first entry, it calculates the address of the associated Critical Router (CR). Then it creates a wildcard multicast route entry (*, PG) for the shared signalling group, where PG stands for *Permanent Group*. Triggered by this entry, it creates a JOIN_REQUEST message and sends it towards the CR. This message contains the address of the critical router in a special field. The interface on which this message is sent is included in the incoming interface list. It is not necessary for this message to travel all the way to the Critical Router. If an intermediate router has already joined the group, it terminates this message and sends an acknowledgement i.e. JOIN_ACK message towards the source of the REQUEST message. If the JOIN_REQUEST message travels all the way to the CR, the CR acknowledges it. The ACK message travels back the same route which the request message travels in the reverse direction.

When an intermediate router receives the REQUEST message it creates a wildcard route entry (*, PG) and includes the interface on which the message arrived in the outgoing interface list. The interface on which this message is forwarded is included in the incoming interface list.

This way the shared signalling tree is formed, which connects all the DRs of networks with members of any group present inside them. The formation of the shared tree is shown in Figure 3.1.

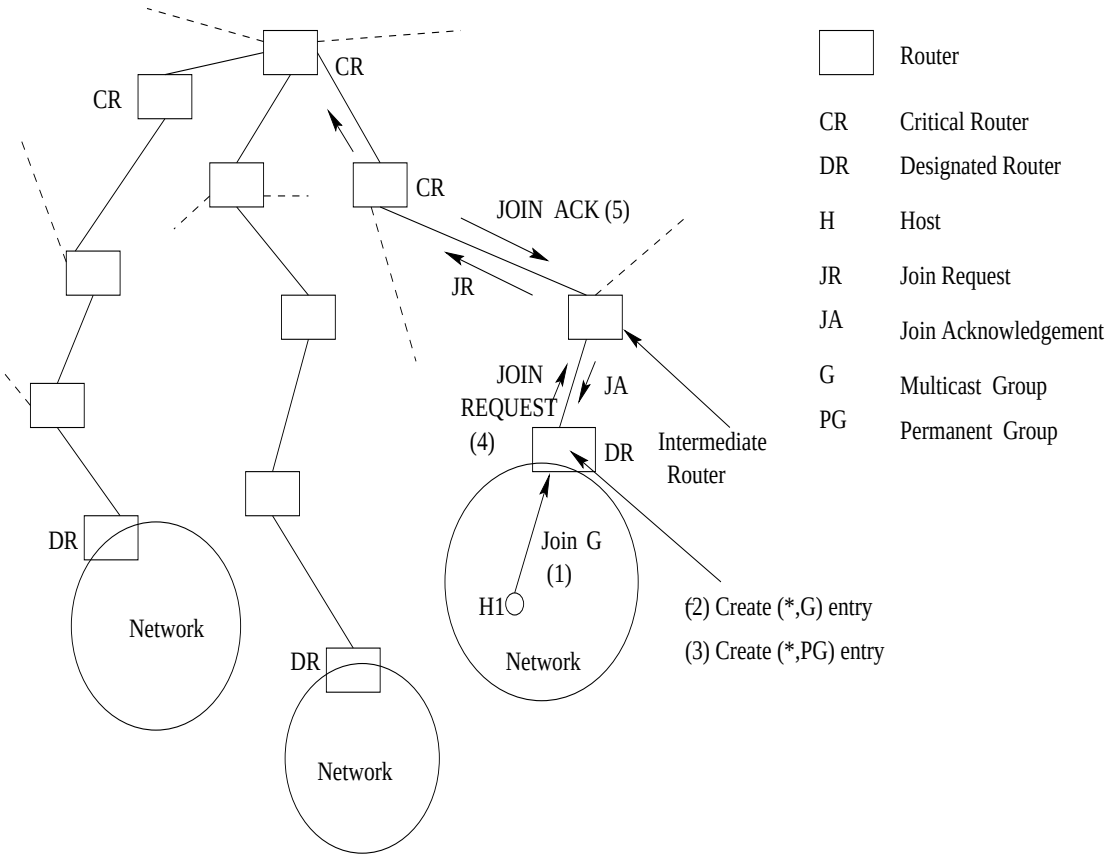


Figure 3.1: Joining the shared-signalling tree

3.3.4 Conveying Information about the Sources to all the DRs

When a host starts sending multicast data packets to a group i.e. it starts acting as a source, it delivers the packets to the Designated Router (DR) of its network. The DR on the receipt of data packets creates a source-specific multicast route entry referred to as (S, G) and includes the interface on which it receives the data packets in the incoming interface list. Then DR creates a SIGNAL_MESSAGE and sends it towards the critical router. This message contains the source and the group address in special fields with the destination as the critical router. The CR on receiving the SIGNAL_MESSAGE distributes it on the entire shared signalling tree. All the DRs which have joined the shared tree receive this information.

Each time a new source appears inside the network, the DR sends a new SIGNAL_MESSAGE. Also, each DR periodically sends an aggregate SIGNAL_MESSAGE on the signal tree which contains information of all the sources present inside the network i.e.

$$[(G_1, S_{11}), (G_2, S_{21}, S_{22}, S_{23}), (G_3, S_{31}, S_{32}, \ldots].$$

A new entry triggered signal message can be merged with this aggregate message provided there is no delay between the appearance of a new source and the sending of aggregate message.

3.3.5 Establishing Source-Based Trees

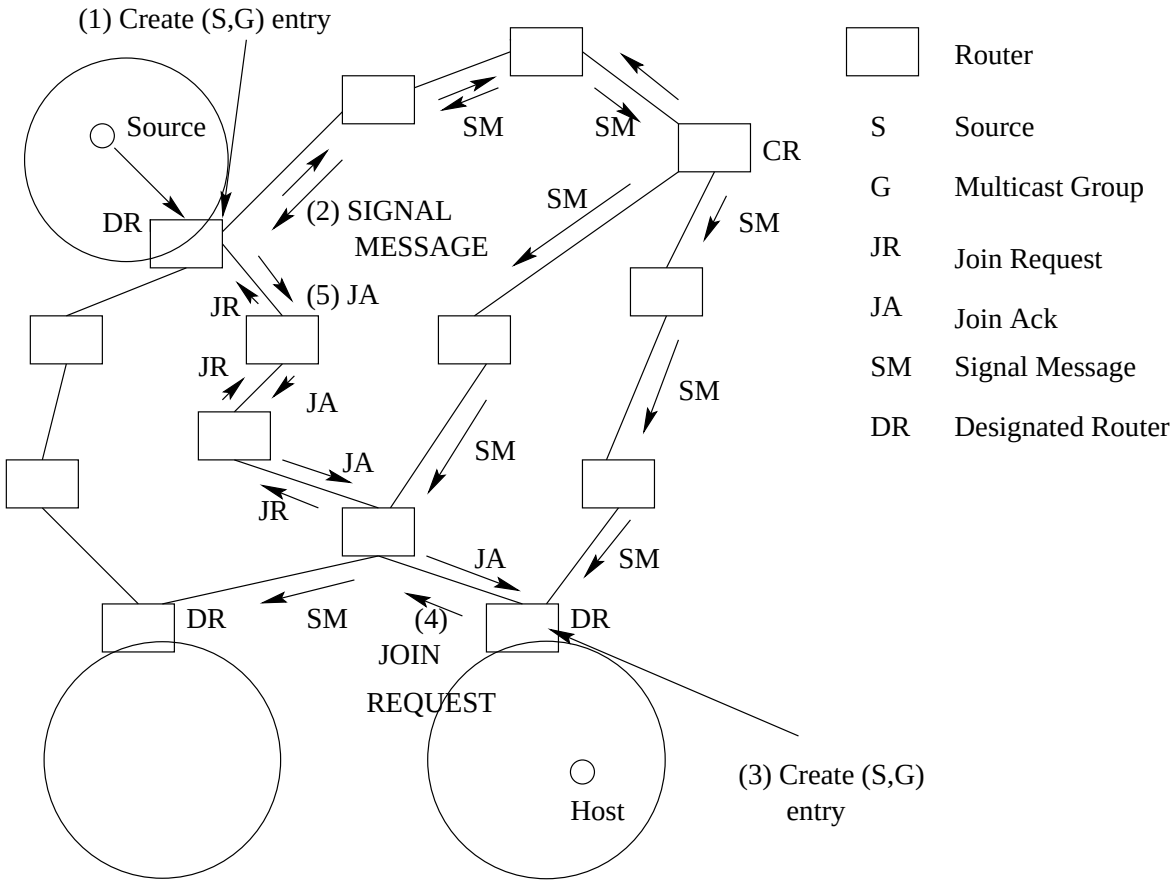


Figure 3.2: Joining the source-based tree

The DRs receive all the SIGNAL messages being transmitted on the shared-signalling tree. A DR on the receipt of a SIGNAL_MESSAGE compares the group part of all the entries in the message with the group part of the entries in its wildcard entry table. If the group parts match, it creates a (S, G) entry for that group provided such an entry does not already exists. Triggered by (S, G) entry created as a result of receipt of signal message on the shared-signalling group, the DR initiates a JOIN_REQUEST message and sends it towards the source i.e. the source network DR. This message contains the address of the source and the multicast group address in special fields. The interface on which this message is sent is included in the incoming interface list. The network interface i.e. the interface on which the router acts as a DR for that network is included in the outgoing interface list.

The JOIN_REQUEST message travels upstream towards the source. It is not necessary for this message to travel all the way to the source DR. If an intermediate router has already joined the source for this particular group, it terminates this message and sends an acknowledgement i.e. JOIN_ACK

message towards the source of the REQUEST message. This acknowledgement travels the same route back as the ACK message travels in the reverse direction.

The intermediate router on the receipt of the REQUEST message creates a (S, G) multicast route entry and includes the interface on which the message arrived in the outgoing interface list. The interface on which the message is forwarded is included in the incoming interface list. The entry being created is confirmed by the acknowledgement message.

This way the source-based trees are formed, as shown in Figure 3.2. The formation of these trees is receiver- initiated.

3.3.6 Hosts Leaving a Group

In IGMP, the DR periodically sends GROUP MEMBERSHIP queries inside the network. If any member of a group is present, it sends a MEMBERSHIP REPORT in response to this. When the DR does not receives membership reports for a group, it removes the corresponding entry from its table and sends a LEAVE_REQUEST message upstream. The upstream router on the receipt of this request acknowledges it by sending a LEAVE_ACK message downstream and removes this interface from the outgoing interface list of this entry. The acknowledgement eliminates any need for the maintenance of LEAVE state and the timers associated with it. Then the upstream router goes on to check whether it has received LEAVE_REQUEST on all the downstream interfaces on which the receivers for this (source, group) pair were present downstream i.e. whether the outgoing interface list for this (source, group) is empty. If so it removes the corresponding entry from its table and sends a LEAVE_REQUEST message further upstream. The upstream router acknowledges this. This way the leave information is propagated upstream.

DRs remove the (source, group) pair entry from their table when they decide that they have no members or receivers of this group who want to receive data from this source inside the network. If the DR does not receives any report specific to a particular group, it removes the (*, G) entry from its wildcard entry table.

If a member appears again after some time, it again joins the tree by sending a JOIN_REQUEST towards the source which is acknowledged by JOIN_ACK message.

3.3.7 Tree Maintenance

The initial phase consists of explicit formation of trees. Once the tree is established, data packets start flowing. The source-based tree maintenance is data-driven i.e. the (source, group) entries are maintained by the flow of data packets. These entries exist as long as the source for this tree is active i.e. as long as there is any source inside a network which is sending data to this group. As soon as the DR removes the (source, group) entry from its table it sends a CEASE_TREE message on the tree. This message is a request to all the routers on the tree that no more data is available for this group and that they should remove the corresponding entry from their tables. This way the tree is gracefully broken down.

The maintenance of shared tree is through the exchange of HELLO messages between the adjacent routers periodically. The child or the downstream router sends a HELLO message to its parent or the upstream router, in response to which the parent also sends a HELLO message to the child. These messages are exchanged at regular intervals.

3.3.8 Multicast Route Entries

There are two types of entries maintained by the routers. First is the (source, group) pair entry, used for the building of source-based trees. A typical entry is shown in Figure 3.3. Second is the wildcard or (*, G) entry, used for the building of the shared signalling tree. A typical entry is shown in Figure 3.4.

SOURCE	MCAST GROUP	IIF	OIF

IIF - Incoming Interface List

OIF - Outgoing Interface List

Figure 3.3: Entry for (source, group) pair

MCAST GROUP	IIF	OIF	TIMER

IIF - Incoming Interface List

OIF - Outgoing Interface List

Figure 3.4: Entry for (*, G) pair

The individual entries are described as :

- SOURCE : The unicast address of the source network.
- MCAST GROUP : A group identifier or the multicast address of the group.
- IIF : An identifier for the interface on which the packets belonging to the (source, group) pair which this entry represents, arrive.
- OIF : A list of the outgoing interfaces on which the packets are to be forwarded.
- TIMER : The duration for which the wildcard entry exists. This timer is updated whenever a signal message is received.

Chapter 4

Protocol Analysis

As far as the router state is concerned, we can classify the routers into three categories i.e. the Designated Routers, the intermediate routers which lie on the source-based tree and the intermediate routers which lie on the shared signalling tree. The amount of state maintained by these routers is different. The state maintained by the DRs consist of wildcard entries (*, G) for all the groups present inside the network and (S, G) entries for all the group specific sources for which the group members are present. Some routers lie only on the source-based trees being formed. These routers maintain (S, G) entries only for those sources on whose trees they lie. The state maintained by routers which lie on the shared signalling tree consist of only (*, PG) entry. Comparing with other protocols, we see that DRs maintain more state but the intermediate routers maintain less or equal state. In the source-based trees formed in this protocol, the routers are only concerned with the sources on whose trees they lie unlike in DVMRP, M-OSPF and PIM-DM in which routers have to maintain state for all the sources since these protocols depend on flooding and pruning. Signalling tree routers maintain a single entry for the existence of shared tree. This is nothing compared to the state maintained by other shared-tree protocols such as CBT and PIM-SM. They maintain state for all the groups lying on the tree. Like in other source-based tree protocols, there is no state maintained by routers which have no members downstream. The traffic is confined only to that part of the network which have the group members present.

In CBT, the problem was that it cannot support high data rate due to the concentration of traffic along that part of the network which covers only the primary and the secondary cores. This was overcome in PIM-SM by using a set of RPs which were hashed to groups and then switching over to source-based tree when the data rate exceeded a certain threshold. Our protocol further distributes the traffic due to the use of source-based trees. The traffic on the signalling tree is very less since only the Signal Messages are passed on this tree. Thus there is no concentration of traffic in any part of the network.

The switching overhead in PIM-SM is eliminated at the cost of initial join time in our protocol. Therefore, the router processing cost i.e. keeping track of the data rate and maintaining switch-over timers, is eliminated. In PIM-SM the source-based and shared trees for the same source exist simultaneously with some receivers on the source-based tree and some on the shared tree. This

increases the router processing and bandwidth usage cost. SCAMP uses only the source-based trees for the distribution of data packets to the group members.

The router processing and the bandwidth usage cost due to the join traffic is also highly distributed due to the use of receiver-initiated source-based trees. This traffic is not concentrated in any particular part of the network. The DRs with sources inside their network are widely separated from each other. Therefore, the join traffic concerned with the source inside the network is confined to the source-based tree rooted at that DR. On the other hand, in CBT and PIM-SM, since they depend on certain routers (say core routers) for the distribution of data, the join traffic for all the groups using a particular Core is concentrated on the shared-tree rooted at that Core. For example, suppose that there are 100 sources belonging to different groups. These sources are located in different networks. Further, they are widely distributed. In shared-tree protocols, suppose we have ten cores and each supports 10 groups. The join traffic due to these 10 groups will be concentrated on the tree with the core acting as the root of the tree formed for these 10 groups. On the other hand, in our protocol the join traffic for any (source, group) pair is concentrated only on the tree whose root is the DR of the network which contains this source.

Due to the use of source-based tree for the distribution of data packets, the end-to-end delay incurred is less than that incurred in CBT and PIM-SM (shared mode).

As far as the tree maintenance traffic overhead is concerned, it is difficult to predict whether the traffic is more or less as compared to other protocols.

The cost incurred by this protocol is that the initial join time is increased but in the type of applications that we are considering, the join time is not that critical.

4.1 Quantitative Analysis

We have discussed how SCAMP compares with the existing protocols with respect to the router state maintenance, join time, traffic concentration, router processing etc. We now give a quantitative comparison for the same.

Today the Internet has more than 100,000 networks. Out of this, say 20,000 of the networks have support for IP multicast. Our goal is to define a protocol such that if *videoconferencing* application has members in 5% (i.e. 1000) of multicast capable networks, the overall cost incurred is less. Each network has a router which acts as a *Designated Router* (DR) for this network. We do not consider the routers inside the network. Apart from these routers, there are other routers in the Internet, whom we call *intermediate routers* which provide the connectivity among different networks. (See Figure 4.1)

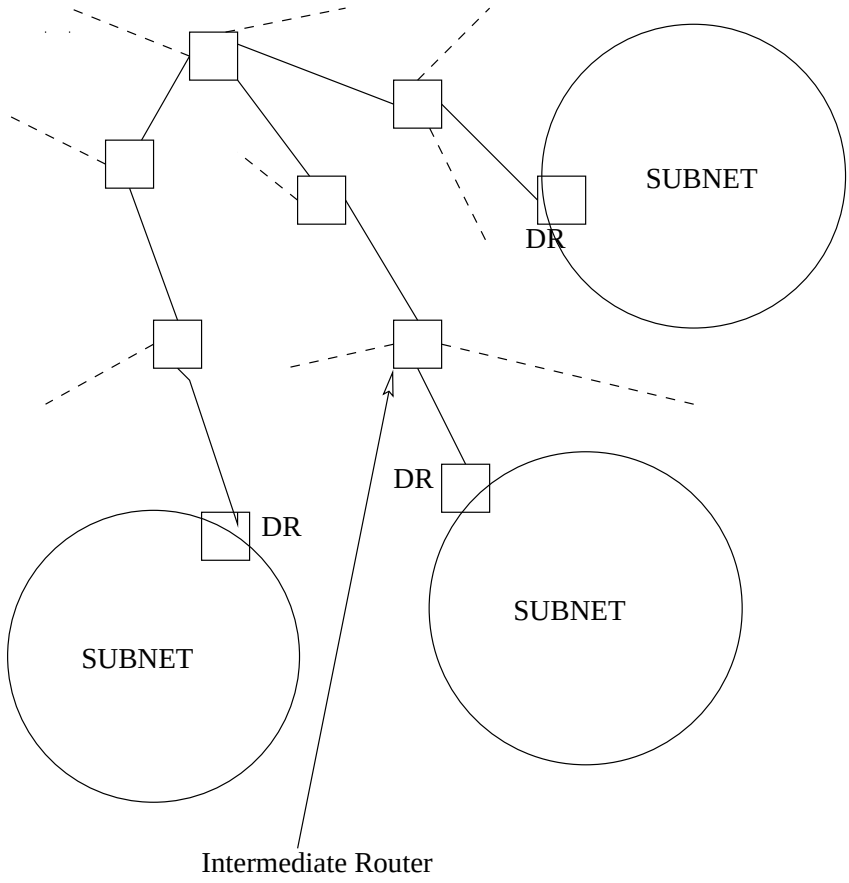


Figure 4.1: Intermediate routers

Total number of multicast capable networks in the Internet	=	S (say 20,000)
Number of active multicast groups in the network	=	N (=1)
i.e. we consider a scenario when a single		
multicast groups exists in the Internet		
Average number of senders to a group	=	n
i.e. those members of a group which act as sources		

There are two types of tree formation i.e. source-based tree (where the tree is routed at the network containing the source) or shared tree (where the tree is rooted at some specific router which acts as Core). It may be that a router lies either on the source-based tree or the shared tree or both. If a router lies on the shared tree for a particular group, we say it supports shared-tree for that group. Similarly, for the source-based tree. For simplification, we assume that there is only one multicast group. the number of sources in this group are finite (some small number). For the shared tree protocols, we assume that all the sources depend on a single core for the distribution of data. Also, there is one shared tree for all the sources. Similarly, in source-based trees we assume that all the sources lie on a single source-based tree.

Not all the routers in the Internet are involved in the formation of trees. If the Internet has 20,000 multicast capable routers in all, then about 25% of them are those which lie either on the shared tree or source-based tree.

Fraction of routers involved = b, where $b \leq 1$

(This includes all the routers which incur processing and state maintenance cost.)

For source-based trees or Dense mode protocols, $b = 1$, since it uses flooding, so all the routers in the Internet receive the packets being transmitted to any group.

For Sparse mode protocols, $b < 1$.
In sparse-mode protocols, the first phase is the building of trees and then once the trees are formed, the data starts flowing. So during the first phase a subset of routers are selected to lie on the tree.

4.2 State Maintained by Routers

4.2.1 DVMRP and PIM-DM

In dense-mode protocols since all the routers in the Internet receive packets initially due to the use of flooding, therefore
Number of routers involved = R

A router has to maintain state for each (source, group) pair. Therefore,
Number of entries = Number of groups * Number of sources or senders per group (1*n = n)

4.2.2 CBT

In sparse-mode protocols such as CBT and PIM-SM, since there is an explicit formation of trees, therefore the number of routers incurring the cost of processing and state maintenance is very less i.e. some fraction of the total number of routers in the Internet, therefore
Number of routers involved = Fraction of routers involved * Total number of routers (b*R)

We assume that there is only one group. All routers involved in tree formation support this group.
Number of entries = Number of groups (= 1)

4.2.3 PIM-SM

Number of routers involved = Fraction of routers involved * Total number of routers (b*R)
We assume that a router lies on all the source-based trees.
For a router which lies only on the source-based tree,
number of entries = Total number of (source, group) pairs (1*n)

For a router which lies only on shared tree, number of entries = Total number of groups (= 1)

A router which lies on both the trees = 1*n + 1

4.2.4 SCAMP

Number of routers involved = Fraction of routers involved * Total number of routers (b*R)
Again we assume that an intermediate router lies on all the source-based trees.
For an intermediate router which lies only on source-based tree, number of entries = Total number of (source, group) pairs (1*n)

For an intermediate router which lies only on shared tree, number of entries = an entry for the shared signalling tree = 1

For DRs, number of entries = Wildcard entries for all the groups present inside the network + (source, group) entries for all the sources belonging to groups for which it has members present + One entry for the shared signalling tree (1 + 1*n + 1)

For an intermediate router which lies on both the trees, number of entries = 1*n + 1

4.2.5 Example

We have studied several multicast routing protocols. The routers involved in these protocols have differnt roles to play. For instance, a single router can maintain the state for (source, group) pair or simply for each group or both. Depending on this, we can classify the routers in four categories. These categories are given below. Further, a router can either lie on the boundary of a network or spread accross the internet to provide connectivity. The former are called designated routers while the latter are called intermediate routers. The two routers are shown in Figure 4.1. The table below in Figure 4.2 shows the number of prunes being processed by routers for various combinations of the number of groups and the number of senders per group.

RST : Router lying on the source-based tree
RShT : Router lying on the shared tree
RB : Router lying on both the trees
DR : Designated Router
In. : Intermediate Router

Number of groups (N) Number of senders/group (n)	N = 1 n = 1	N = 1 n = 2	N = 1 n = 5	N = 1 n = 10
DVMRP, PIM-DM RST	1	2	5	10
CBT RShT	1	1	1	1
PIM-SM RST RShT RB	1 1 2	2 1 3	5 1 6	10 1 11
SCAMP In. RST In. RShT In. RB DR	1 1 2 3	2 1 3 4	5 1 6 7	10 1 11 12

Figure 4.2: Comparison of router state maintenance

Thus we see that the state maintained by DRs is more. The state maintained by intermediate routers is either less than or almost equal to the routers in both CBT and PIM-SM. Though the protocol supports the formation of source-based trees but the state maintained by routers is much less than the state maintained by routers in other Source-based protocols.

4.3 Traffic Concentration

We consider sparse groups i.e. only one member of a group (be it a sender or receiver) belongs to one network, though a network may have members of different groups.

4.3.1 CBT and PIM-SM

In these shared tree protocols, a set of routers are designated to act as Cores or Rendezvous Points (RPs). A member wishing to join a group, somehow calculates the address of an RP and then proceeds to join the tree rooted at that RP. Thus, an RP supports trees for a small set of groups. We assume that there is only one RP and a single shared tree.

Number of RPs (Cores)	=	1
Average number of groups supported by an RP	=	Total number of groups divided by the number of RPs (=1)

In a particular group, it may be that Y out of X sources are transmitting data simultaneously.

Fraction of senders per group sending simultaneously	=	y
Average data rate of each sender	=	K Kb/sec.
i.e. the rate at which the source is transmitting data.		
Average number of children of a router		
i.e. number of downstream interfaces	=	c

We assume that the group being supported by an RP is present on all the downstream interfaces of that RP.

Number of joins processed by an RP	=	Number of downstream interfaces (c)
Data processed by an RP	=	Fraction of senders per group sending simultaneously * number of senders/group * Number of groups supported by an RP * Data rate per sender (y * n * 1 * K) Kb/sec.

4.3.2 SCAMP

We treat the DRs of source networks analogous to the RPs. As in shared-trees, all the senders first send the data towards the RP which then distributes to the entire group, similarly in our protocol, the sender within the network sends data to the DR which then distributes it to the entire group. This analogy is just to show that how the Join and data traffic is distributed.

Source-based trees DR

Number of joins processed	=	Number of downstream interfaces (c)

Number of senders present inside one network = 1

Data processed by DR	=	Number of senders inside the network * Data rate of each sender (1 * K) Kb/sec.

Shared tree Critical Router

Number of joins processed	=	Number of downstream interfaces (c)
Signal messages processed	=	Total number of (source, group) pairs (1 * n)

4.3.3 Example

$Number of RPs = 1$; $Number of groups, N = 1$;
Number of senders per group, $n = 1, 2, 5, 10$;
Fraction of senders per group sending simultaneously,$y = 0.2$;
We assume that for n = 1, 2, all senders per group send simultaneously i.e. $y = 1$
Data rate of each source, $K = 10Kb/sec.$; $c = 4$;
Number of senders inside a network , $m = 1$
We can consider Size of Join Pkts = Size of Signal Messages = 30 bytes
and that these packets are distributed over a time interval of T = 4 sec.

	Number of Joins	Data (Kb/sec.)	Number of Signal msgs
CBT and PIM-SM			
n = 1	4 = 30 bytes/sec.	10	
n = 2	4 = 30 bytes/sec.	20	
n= 5	4 = 30 bytes/sec.	50	
n = 10	4 = 30 bytes/sec.	200	
SCAMP			
n = 1			
DR	4 = 30 bytes/sec.	10	
RP	4 = 30 bytes/sec.		1 = 7.5 bytes/sec.
n = 2			
DR	4 = 30 bytes/sec.	10	
RP	4 = 30 bytes/sec.		2 = 15 bytes/sec.
n = 5			
DR	4 = 30 bytes/sec.	10	
RP	4 = 30 bytes/sec.		5 = 37.5 bytes/sec.
n = 10			
DR	4 = 30 bytes/sec.	10	
RP	4 = 30 bytes/sec.		75 bytes/sec.

Figure 4.3: Comparison of traffic concentration

Thus we see from Figure 4.3 that in our protocol there is no concentration of traffic in any part of the network. The number of signal messages are further reduced. Initially, as the new sources appear inside the network, the DR of that network sends a signal message for that (source, group) pair. Thus initially the number of signal messages is equal to the total nunber of (source, group) pairs. Then the DRs send aggregate signal messages periodically which contain a list of all the (source, group) pairs for which it has group-specific sources inside the network. Therefore, the maximum number of signal messages is equal to the actual number of networks with sources inside them.

4.4 Join Time

In our protocol the receiver network DR have to wait for the signal messages through which it comes to know of the senders of the groups for which it has members inside the network. It then proceeds to join the sources. This increases the initial Join time. Therefore, the initial join time is more as compared to other protocols.

4.5 Router processing

A router has a number of downstream interfaces. We have assumed that there is a single multicast group. This group is present on all the interfaces of a router which lies on the group specific multicast tree. Then gradually the group can appear or disappear periodically on an interface. We consider two cases with regard to the number of groups and the number of senders to a particular group, i.e. N = 1, n = 2 and N = 1, n = 5 . Number of downstream interfaces, c = 4. Apart from these parameters, the value of any new parameter being introduced is given along with it. We give an analysis of the average number of packets being processed by a router, where [case 1, case 2] specify the number of packets being processed when n = 2 and 5 respectively.

4.5.1 DVMRP & PIM-DM

In these protocols only a subset of routers in the Internet are involved in supporting the formation of trees, while other routers just incur the cost.

Considering routers which support the tree formation. Initially, we assume that the group is present on all the interfaces of such a router.
Suppose that the group members disappear on say x (= 2) interfaces. We call these interfaces as inactive interfaces. Then the

Number of prunes processed by a router	=	Number of (source, group) pairs for all the groups not present per interface * Number of inactive downstream interfaces
	=	1 * n * x
		[4, 10]

Similarly, if the group members reappear on these interfaces,

Number of grafts processed	=	Number of (source, group) pairs for all the dynamic groups per interfce * Number of interfaces
	=	1 * n * x
		[4, 10]

Now, consider routers which do not lie on trees. In these routers the group is not present on any interface.

Number of prunes processed by a router	=	Number of (source, group) pairs * Number of downstream interfces
	=	1*n*c
		[8, 20]

4.5.2 CBT

There are two phases, the initial phase being the explicit tree formation phase and then intermediate phase during which ocassionally groups can appear downstream.

During the initial phase,

Number of joins and join-acks processed by a router	=	Fraction of groups present downstream per interface * Number of interfaces + Number of Acks
	=	1*c + 1
		[5, 6]

If the group disappears on x (= 2) interfaces,

Number of Quits processed by a router	=	1*x
		[2, 2]
Average number of Keepalive messages	=	Number of downstrean interfaces
	=	c per T interval
		[4, 4]

During the intermediate phase, suppose that the group reappears on x interfaces.

Number of joins processed by a router	=	1*x
		[2, 2]

PIM-SM During the initial phase,
Number of joins processed by a router = Number of groups present downstream per interface * Number of interfaces
= 1*c
[4, 4]

Number of aggregate JOIN/PRUNEs processed by a router = Number of downstream interfaces (c) per T interval
[4, 4]

During switch-over to source-based trees,
Number of joins processed by a router = Number of (source, group) pairs supported per interface * Number of interfaces
= 1*n*c
[8, 20]

Apart from this, there is cost incurred due to C-RP-Adv messages, BSR messages etc.

4.5.3 SCAMP

Consider an intermediate router on source-based tree.
During the initial phase,
Average number of joins and join-acks processed by a router = Number of (source, group) pairs present downstream per interface * Number of interfaces + Number of Acks
= 1*n*c + 1*n
[10, 24]

Consider an intermediate router on the shared tree.
Number of joins and join-acks processed by a router = Number of downstream interfaces + 1
= c + 1
[5, 5]

Number of signal messages processed = Number of (source, group) pairs initially and then the number of networks involved

Consider a Designated Router (DR).
Number of joins processed = Number of downstream interfaces + 1
= c + 1
[5, 5]

Number of signal messages processed by a DR = Number of (source, group) pairs initially and then number of networks involved

On source-based tree, tree maintenance is data-driven. It may be that the group members disappear on x (= 2) interfaces. Therefore the

Number of Leave messages processed by a router	=	Number of groups disappearing per interface * Number of (source, group) pairs supported * Number of interfaces
	=	1*n*x
		[4, 10]
On shared-tree, number of Keepalive messages processed	=	Number of downstream interfaces (c) every T interval
		[4, 4]

Initially the number of signal messages processed are very large but the routers which lie only on the shared-tree process only these messages and no data messages, thus the load on them is not very high.

Chapter 5

Conclusion And Future Work

We have given an overview of the existing multicast protocols discussing their scalability in the context of large groups. Having studied the existing protocols, we felt the need for a new protocol which can support high data rate and sparse as well as dense multicast groups. We have described the new protocol in detail and showed how our protocol gains over other protocols. We gave a quantitative analysis of the protocol. We compared the various multicast protocols with regard to the router state maintenance, concentration of traffic, join time and router processing cost. We found that our protocol does reduces the costs compared to other protocols.

5.1 Results

Having compared our protocol with the other multicast routing protocols, we make the following observations.

1. The state maintained by the routers in our protocol is either less or equal to the other protocols.
2. The end-to-end delay in source-based trees is almost half of that in shared trees. Since our protocol builds source-based trees for the delivery of data packets, it is well suited for multicast applications which require less end-to-end delay.
3. Due to the building of source-based trees, our protocol can support high data rate which was a problem in shared tree protocols.
4. Our protocol reduces the traffic concentration problem in shared tree protocols by further distributing the traffic. In shared tree protocols, a set of routers acted as cores. This resulted in the concentration of traffic in certain parts of the network i.e. those parts in which the core routers are present. We distribute the functionality of the core to a large number of routers or the DRs of the networks with sources inside them. This has the effect of distributing the traffic. Further the dependance of sources on the cores for the distribution of data to the group members is also eliminated.

5. The distribution of traffic over a larger area also reduces the processing cost of the control and data packets incurred by the routers.

6. Our protocol achieves the above mentioned advantages at the cost of the join time of the group members. Since the receiver cannot join the group till the information regarding the source is conveyed to it, there is an increase in the joining time.

5.2 Future Work

We have developed the protocol and studied it theoritically. There is a lot of work that can be done in this area.

- There is a need to implement a multicast routing daemon of our protocol SCAMP.
- One can build a topology of a few routers all running an instance of this daemon. Further, one can create a few sample groups of the type described in the text and test the protocol.
- Our protocol supports multicast applications with large groups which have very strict Quality 0f Service (QoS) requirements. To ensure the desired Qos to the application, one can study how this protocol can be integrated with protocols which provide resource reservation, such as Resource Reservation Protocol (RSVP) [11].

Bibliography

[1] Internet Multicast Addresses. Internet Assigned Numbers Authority. Available from ftp://ftp.isi.edu/in-notes/iana/assignments/multicast-addresses.

[2] Sudhir Aggrawal and Sanjay Paul. A Flexible Protocol Architecture for Multi-party Conferencing.

[3] Kevin C. Almeroth and Mostafa H. Ammer. Providing a scalable, interactive video-on-demand service using multicast communication. Technical Report GIT-CC-94/36, Georgia Institute of Technology, Mar 1994. Available from http://www.cc.gatech.edu:81/Dinest/Repository/2.0/Body/ncstrl.gatech_cc/.

[4] Mostafa H. Ammar. Probabilistic multicast:generalizing the multicast paradigm to improve scalability. Technical Report GIT-CC-93/28, Georgia Institute of Technology, Apr 1993. Available from http://www.cc.gatech.edu:81/Dinest/Repository/2.0/Body/ncstrl.gatech_cc/.

[5] P. Bagnall, R. Briscoe, and A. Poppit. *Taxonomy of Communication Requirements for Large-scale Multicast Applications.* Internet Draft, Internet Engineering Task Force, draft-ietf-lsma-requirements-01.txt, Nov 1997. Work in Progress.

[6] A. Ballardie, P. Francis, and J. Crowcroft. Core Based Trees (CBT) and Architecture for Scalable Inter-domain Multicast Routing. *ACM SIGCOMM*, Sep 1993.

[7] A. J. Ballardie. *Core Based Tree (CBT) Multicast Interoperability - stage 1.* Internet Draft, Internet Engineering Task Force, draft-ietf-idmr-cbt-interop1-00.txt, Apr 1996. Work in progress.

[8] A. J. Ballardie. Core Based Tree (CBT) Multicast Routing Architecture. RFC: 2201, Sep 1997.

[9] A. J. Ballardie, S. Reeve, and N. Jain. Core Based Tree (CBT version 2) Multicast Routing. RFC: 2189, Sep 1997.

[10] Anthony J. Ballardie. *A New Approach to Multicast Communication in a Datagram Internetwork.* PhD thesis, Deptt. of Comp. Sc., University College London, University of London, May 1995. Available from ftp://cs.ucl.ac.uk:darpa/IDMR/ballardie-thesis.ps.Z.

[11] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – version 1 Functional Specification. RFC: 2205, Sep 1997.

[12] Brad Cain, Steve Deering, and Ajit Thyagarajan. *Internet Group Management Protocol, Version 3.* Internet Draft, Internet Engineering Task Force, draft-ietf-idmr-igmp-v3-00.txt, Nov 1997. Work in progress.

[13] K. Carlberg and J. Crowcroft. Building shared trees using a one-to-many joining mechanism. *ACM SIGCOMM Computer communication review*, Jan 1997.

[14] S. Casner. *Frequently Asked Questions (FAQ) on the Multicast Backbone (MBONE).* USC Information Sciences Institute (ISI), Dec 1994. Available from ftp://venera.isi.edu/mbone/faq.txt.

[15] S. Deering. Host Extensions for IP Multicasting. RFC: 1112, May 1988.

[16] S. Deering, A. Thyagarajan, and S. Casner andand others. mrouted Release 3.8 implementation, Nov 1995. Available from ftp://parcftp.xerox.com/pub/net-research/ipmulti/mrouted3.8.tar.Z.

[17] Stephen E. Deering. Multicast routing in internetworks and extended lans. *ACM SIGCOMM*, 25(1), Jan 1995.

[18] Steven Deering, Deborah Estrin, Van Jacobson, Puneet Sharma and, et al. *Protocol Independent Multicast-Sparse Mode (PIM-SM) : Motivation and Architecture.* Internet Draft, Internet Engineering Task Force, draft-ietf-idmr-pim-arch-05.txt, Aug 1998. Work in progress.

[19] C. Diot, W. Dabbous, and J. Crowcroft. Multipoint communication:a survey of protocols, functions and mechanisms. *IEEE Journal on Selected Areas in Communications*, 15(3), Apr 1997.

[20] Hans Eriksson. Mbone:the multicast backbone. *Communication of the ACM*, 37(8), Aug 1994.

[21] Deborah Estrin, Steven Deering, Van Jacobson, Puneet Sharma and, et al. Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification. RFC: 2362, Jun 1998.

[22] Deborah Estrin, Dino Farinacci, Ahmed Helmy, Van Jacobson and, et al. *Protocol Independent Multicast (PIM), Dense Mode:Protocol Specification.* Internet Draft, Internet Engineering Task Force, draft-ietf-idmr-PIM-DM-spec-09.txt, Sep 1996. Work in progress.

[23] Deborah Estrin, Mark Handley, Ahmed Helmy, Polly Huang, and David Thaler. A dynamic bootstrap mechanism for rendezvous-based multicast routing. Technical report, University of Southern California. Available from ftp://catarina.usc.edu/pub/pim/pimsm/bootstrap-TechReport.ps.gz.

[24] W. Fenner. Internet Group Management Protocol, version 2. RFC: 2236, Nov 1997.

[25] J. Hawkinson. *Multicast Pruning a Necessity.* Internet Draft, Internet Engineering Task Force, draft-ietf-mboned-pruning-00.txt, Aug 1996. Work in progress.

[26] Ahmed Helmy. Pim multicast daemon. University of Southern California, Mar 1998.

[27] David Meyer. Administratively Scoped IP Multicast. RFC: 2365, Jul 1998.

[28] J. Moy. Mospf: Analysis and Experience. RFC: 1585, Mar 1994.

[29] J. Moy. OSPF version 2. RFC: 1583, Mar 1994.

[30] John Moy. Multicast Extensions to OSPF. RFC: 1584, Mar 1994.

[31] Sassan Pejhan, Alexandros Eleftheriadis, and Dimitris Anastassiou. Distributed multicast address management in the global internet. *IEEE Journal on Selected Areas in Communications*, 13(8), Oct 1995.

[32] J. M. Pullen, M. Myjak, and C. Bouwens. *Limitations of Internet Protocol Suite for Distributed Simulation in the Large Multicast Environment*. Internet Draft, Internet Engineering Task Force, draft-ietf-lsma-limitations-02.txt, Feb 1998. Work in Progress.

[33] T. Pusateri. *DVMRP Version 3*. Internet Draft, Internet Engineering Task Force. Work in Progress.

[34] T. Pusateri. Distance Vector Multicast Routing Protocol. RFC: 1075, 1997.

[35] Kevin Savetz, Neil Randall, and Yves Lepage. *MBONE: Multicasting Tomorrow's Internet*. IDG, 1996. Available from http://www.savetz.com/mbone/.

[36] Steve Seidensticker, W. Garth Smith, and Michael Myjak. *Scenarios and Appropriate Protocols for Distributed Interactive Simulation*. Internet Draft, Internet Engineering Task Force, draft-ietf-lsma-scenarios-01.txt, Mar 1997. Work in Progress.

[37] Ajit S. Thyagarajan and Stephen E. Deering. Hierarchical distance vector multicast routing for the mbone. *ACM SIGCOMM*, 25(4), Oct 1995.

[38] Thierry Turletti and Christian Huitema. Videoconferencing on the internet. *IEEE/ACM Transactions on Networking*, 4(3), Jun 1996.

[39] Mark H. Willebeek-LeMair and Zon-Yin Shae. Videoconferencing over packet based networks. *IEEE Journal on Selected Areas in Communications*, 15(6), Aug 1997.