A Unified Framework for Multicast Forwarding

Zheng Wang and Jon Crowcroft

Computer Science Department, University College London
Gower Street, London WC1E 6BT, United Kingdom

Abstract. This paper examines the issues of implementation and integration in multicast. A framework for multicast forwarding is presented, which separates multicast forwarding from any associated protocols that may affect the forwarding decisions, such as multicast routing, resource management and host membership protocols. This framework allows seamless integration of various associated protocols and provides a unified interface for new development.

1 Introduction

Multicast provides an important service that has been extensively used by multimedia applications such as video and audio conferencing. The experiments over the MBONE [1], an Internet-wide experimental multicast testbed, have demonstrated the enormous potential of multicast, and also provide us directions of further enhancing this service.

To support multimedia applications such as video and audio conferencing, the routers must have the following essential components:

- A multicast routing protocol (MRP) for constructing and maintaining the multicast trees. The multicast routing protocol running over MBONE now is a version of DVMRP [2], based on the truncated-broadcast algorithm [3]. A number of proposals have been put forward to improve the scaling of multicast routing. CBT [4] proposes a core-based tree to replace multiple sender-based trees. Two other proposals SRPM and PIM [5, 6] attempt to improve the scaling for some particular cases while preserving the basic sender-based tree structure. It is likely that we have to run multiple multicast routing algorithms, and let the applications choose the best multicast algorithm for their style.
- A resource management protocol (RMP) for resource setup and enforcement. Multimedia applications have much more stringent QoS requirements than traditional data applications. For a network to deliver QoS guarantees, the routers must make appropriate resource reservation and excise necessary resource control. The current Internet does not have any mechanisms for resource control yet. Nevertheless, a number of resource management schemes have been proposed over the past several years [7, 8, 9, 10, 11].
- A host membership protocol (HMP) for managing the host membership of a multicast group on a subnetwork. The Internet Group Message Protocol (IGMP) has been developed in DVMRP for this purpose.

Most of the protocols are still under research and development. It is not clear yet whether any of the protocols may eventually be accepted and standardized by the Internet community. Also, Multicast routing and resource management are active research areas hence new protocols will be proposed in the future. It is also possible that multiple multicast and resource management protocols will co-exist and the applications may choose the protocols for their style of multicast.

Therefore, from the perspective of implementation and integration, the issues in multicast are:

- How to integrate various components under a unified framework
- How to accommodate the need for rapid prototyping and new development
- How to copy with multiple multicast and resource management protocols

In this paper, we propose a unified framework for multicast forwarding. The basic idea is to separate multicast forwarding from multicast routing, resource management and host membership protocols. The framework has the following features:

- It seamlessly integrates multicast routing, resource management and host membership protocols with multicast forwarding.
- It provides a unified interface for new development.
- It allows the co-existence of multiple multicast routing and resource management protocols.
- It eliminates forwarding decision making on a per-packet basis, thus speeds up packet forwarding.

2 Basic Approach

In the implementation of DVMRP based on [3], the forwarding of multicast packets is oriented around each network interface's structure (vif). There is not a clear boundary between the multicast routing and forwarding; they are integrated in one piece. The decision as to whether a packet should be forwarded on an interface is based on the protocol state information stored in the interface's structure. Such a design makes it difficult to integrate other multicast routing protocols within the same structure.

In unicast routing, there is a clear separation between routing and forwarding, which is regarded by some people as one of the fundamental design decisions leading to the success of the Internet [12]. In the BSD UNIX design, the routing algorithm exchanges information among nodes and produces a *forwarding table*. The router simply forwards packets to the next hop specified in the forwarding table. This separation has several advantages. It allows the routing and forwarding to evolve separately, and also provides a general framework for implementation of any new unicast routing algorithms.

In our design, we attempt to extend this concept further. We separate multicast forwarding from any associated protocols and define a clear interface between them. We use the term ***associated protocols*** here to refer to any protocols which may affect forwarding decisions, such as multicast routing protocols, resource management protocols, and host membership protocols. This separation reflects the different functions that the associated protocols and the multicast forwarding engine perform; the associated protocols *make* forwarding decisions while the forwarding engine simply *executes* the decisions. A useful analogy to our approach is the way in which parliaments and governments work. In theory, the parliaments make legislation while the governments execute them.

The key to this separation is to decouple the protocol state information from the forwarding state information. In our design, each associated protocol maintains the protocol-specific state information in its own structure, and the forwarding engine maintains the forwarding-specific state information in its forwarding table. Each time when there is a change in the protocol state information, the associated protocol evaluates the change. If the change affects the forwarding decision, the associated protocol makes a call to modify the the forwarding state information. In such a design, the forwarding decisions pass from the associated protocols to the forwarding engine. The forwarding engine does not have to make any forwarding decisions; it simply does the forwarding according to the forwarding state information. The forwarding state information is protocol-independent and can serve as a unified interface to any associated protocol.

3 A Unified Multicast Forwarding Engine

The minimum information needed for packet forwarding is the next-hop to which the packet should be forwarded. In unicast, the forwarding path is a sink tree towards the destination (see Figure 1). Therefore, the forwarding information can be expressed as `(Tree-Id, Next-Hop)`. In single-path unicast routing, each destination has a unique tree thus the `Tree-Id` may be replaced by the `Destination-Id`. So the forwarding information can be expressed as `(Destination-Id, Next-Hop)`. In multicast, the forwarding path is a source tree to a group of destinations (see Figure 2). Therefore, the forwarding information can be expressed as `(Tree-Id, Next-Hops)`. Since we can have encapsulated tunnels, we usually use the out-going interfaces to describe the tree branches. Thus, the forwarding information can be expressed with a list of branches `(Tree-Id, Out-Branch-List)`.

Figure 3 shows the forwarding table. Each multicast tree is uniquely identified by a `Tree-Id`. the exact form of the `Tree-Id` depends on the multicast routing protocols. For example, a sender-based algorithm such as DVMRP can use sender's source address plus the multicast group address as the `Tree-Id`. A core-based algorithm such as CBT can use the core address plus the multicast group address as its `Tree-Id`.

The `In-Branch` is a pointer to the interface from which the multicast packet of that tree should be received. It is necessary because in multicast protocols based on reverse-path forwarding, such as DVMRP [3], a router has to check the in-coming packets to detect the last hop of the reverse path. Packets received

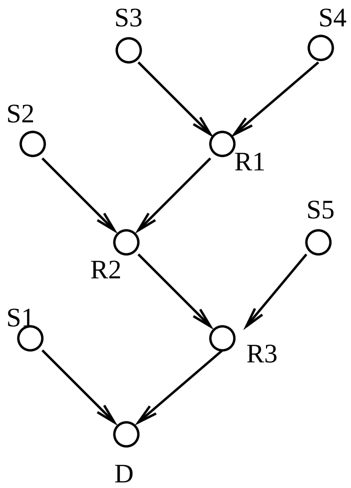

Fig. 1. A Sink Tree in Unicast Routing

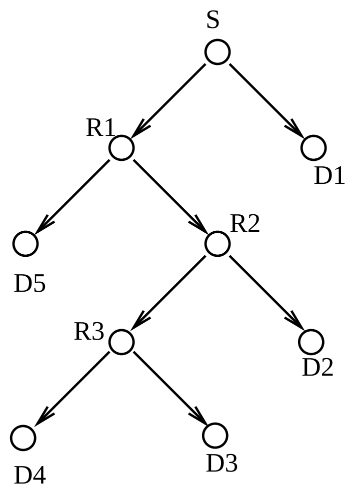

Fig. 2. A Source Tree in Multicast Routing

from the interfaces other than the one specified in the `In-Branch` should be discarded.

The `Out-Branch-List` is a list of `Out-Branches`. Each `Out-Branch` represents a branch of that multicast tree that the router should forward the packet to. An `Out-Branch` consists of a pointer to the interface to which the packet should be forwarded, and a pointer to the `Resource Context`. The `Resource Context` is a pointer to the structure where resource management state information for this `Tree-Id` on this `Out-Branch`.

When a multicast packet is received, the router looks up the entry in the

Tree-ID	In-Branch	Out-Branch-List

Fig. 3. Multicast Forwarding Table

forwarding table for the tree the packet belongs to. If the `In-Branch` does not match the interface from which the packet has arrived, the packet is discarded. Otherwise, the router forwards the packet to each of the interfaces specified in the `Out-Branch-List`. The resource management information that the `Resource Context` points to can be used to determine where the packet should be placed in the out-going queue. Note that for each packet forwarding, only one lookup in the routing table is required for routing and resource management, i.e. determining the out-going interface and the position in the out-going queue.

The multicast forwarding engine provides a unified interface to the associated protocols with a set of routines for manipulating the forwarding table. The multicast forwarding engine itself is independent of any associated protocols. Any associated protocol can be implemented upon the set of manipulating routing.

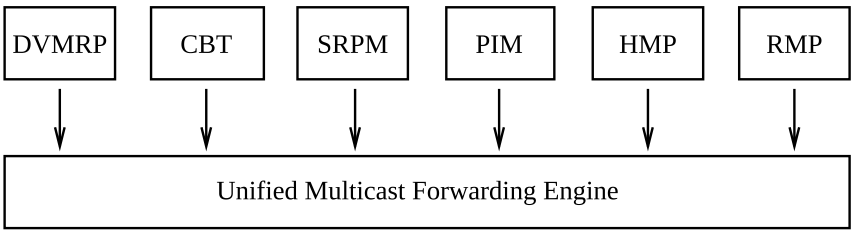

Fig. 4. A Unified Interface for Associated Protocols

Our design allows multiple multicast routing protocols and multiple resource management protocols to operate simultaneously. Each entry in the forwarding table is independent of each other, and may be set up by different protocols. For example, two entries in the forwarding table may be set up by CBT and DVMRP respectively.

4 Examples

Let us now illustrate the operation with a few examples. Suppose that router R has three interfaces A, B and C. When it receives a request for setting up a new DVMRP tree with source address Add1 and multicast group Grp1, it checks with the unicast routing table to find the next-hop to Add1 from R. Suppose that interface A is the shortest path to Add1 from R. DVMRP then adds the following entry in the forwarding table:

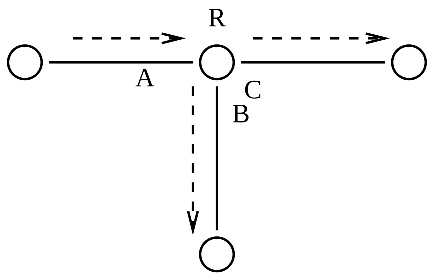

Fig. 5. An Simple Configuration

```
Tree-Id         = {Add1 , Grp1}
In-Branch       = {A}
Out-Branch-List = {(B, null) , (C, null)}
```

if a DVMRP pruning message is received for interface B, the DVMRP has to modify the Out-Branch-List as follows:

```
Out-Branch-List = {(C, null)}
```

Now suppose a user makes a resource reservation on the branch with interface C for source Add1 and multicast group Grp1. The resource management protocol sets up the resource state information. Suppose RC is the pointer to the resource structure, the resource management protocol modifies the Out-Branch-List again:

```
Out-Branch-List = {(C , RC)}
```

Suppose that the DVMRP pruning from B times out. The DVMRP simply manipulates the entry again:

```
Out-Branch-List = {(B, Null) , (C , RC)}
```

A different multicast group may choose CBT to manage its tree and it can operate at the same time along the DVMRP tree. Suppose that router R receives a request to set up a CBT tree with core address Add2 and group Grp2, and interface C is the branch to from the router to Add2. A new entry is then added:

```
Tree-Id         = {Add2 , Grp2}
In-Branch       = {A, C}
Out-Branch-List = (C, null)
```

5 Future Work

We are currently making detailed design and plan to implement DVMRP, CBT and resource setup schemes such as RSVP [10] with our unified framework, and experiment over the MBONE.

References

1. S. Casner, S. Deering, *First IETF Internet Audiocast*, ACM Computer Communication Review, Vol. 22, No. 3, July 1992.
2. D. Waitzman, C. Partridge, S. Deering, *Distance Vector Multicast Routing Protocol*, RFC 1075, Nov 1988.
3. S. Deering, *Multicast Routing in a Datagram Internetwork*, Stanford University, PhD Thesis, Nov 1991.
4. T. Ballardie, P. Francis, J. Crowcroft, *Core Based Tree (CBT)*, ACM Sigcomm'93, Sept 1993.
5. S. Brim, *Scalable Reverse Path Multicasting (SRPM)*, Preprint, June 1993.
6. S. Deering, D. Estrin, D. Farinacci, and V. Jacobson, *IGMP Router Extensions for Routing to Sparse Multicast-Groups*, Preprint, July 1993.
7. V. Jacobson, D. Clark, *Flexible and Efficient Resource Management for Datagram Networks*, Presentation Slides, Sept 1991.
8. D. Ferrari, A. Banerjea, H, Zhang, *Network Support for Multimedia, A Discussion of the Tenet Approach*, Technical Report TR-92-072, University of California at Berkeley, Nov 1992.
9. D. Clark, S. Shenker, and L. Zhang, *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism*, ACM Sigcomm'92, Sept 1992.
10. L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, *RSVP: A New Resource ReSerVation Protocol*, IEEE Networks, Sept 1993.
11. S. Deering, *MBONE Enhancements*, Internal Presentation, End-to-End Research Meeting, June 1993.
12. V. Jacobson, *Internal Discussion*, End-to-End Research Meeting, June 1993.

This article was processed using the LaTeX macro package with LLNCS style