Internet Draft
Document:
draft-nickless-mcast-ipv4-impl-survey-00.txt
Expires: July 2002

B. Nickless
Argonne National
Laboratory
January 2002

A Survey of IPv4 Multicast Service Implementations

1 Status of this Memo

2 Abstract

This document surveys various ways of implementing the IPv4 Multicast Service Model. Each mechanism is evaluated against a common set of criteria.

3 Table of Contents

4 Introduction

The traditional Multicast [MCAST][SVCMODEL] service model is conceptually simple: a source application sends datagrams to the network, which delivers those datagrams to interested receiver applications. Some applications could benefit from extensions to this conceptual model, such as providing the source application with knowledge of the existence of receivers. There remain networks where IP Multicast deployment is not complete, requiring applications to work around the problem. An application may wish to control which sources and receivers are permitted to participate.

Criteria for evaluating implementations to support the application framework are supplied, along with a brief survey of some existing possible implementations.

5 Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

6 Evaluation Framework

As shown in the rest of this Draft, there are several possible ways of providing the multicast service to applications. Criteria for evaluating each approach should include:

- How expensive is the deployment of the approach? Does it require the cooperation of the local network administrator to operate?
- Is it compatible with the Best Current Practices of routing IP multicast in the Internet core?
- Which of the Multicast Service Model [SVCMODEL] Core and Desired operations are supported?
- Does it ensure that only one copy of a multicast packet need traverse a network link?
- Does it allow for both multicast transmission and reception?

Each of the following approaches is evaluated by these criteria.

7.1 Native Multicast (IGMPv2)

IGMPv2 [IGMPv2] is the traditional way for hosts to interact with networks to provide the core Multicast Service Model Operations.

InterfaceDiscovery is supported by putting reachability information in the host reachability table, generally with a static route to the 224.0.0.0/4 logical internet subnet. The application looks up this

reachability information and uses the IP address associated with the appropriate host interface.

GroupTransmit is supported by the host executing the equivalent of a POSIX sendto() system call, with the source address set by the InterfaceDiscovery step above, and the destination address set to the intended Group address.

GroupJoinAny is supported through the operation of the IGMP protocol. The application instructs the host to create IGMP Group Membership reports, which the network uses to create appropriate forwarding trees to the local subnetwork.

The GroupJoinOnly operation is not supported in IGMPv2.

GroupReceive is supported by the host executing the equivalent of a POSIX recvfrom() call, with the from address set to the Group address.

GroupLeave is supported through the operation of the IGMPv2 protocol. The application instructs the host to stop issuing IGMPv2 Group Membership reports indicating interest in that Group.

An extension of IGMP might support the SecureJoinAny Operation.

Implementation of this approach requires that the host, and the host network, be made IGMP-aware.

If properly configured by the network administrator, it is compatible with IPv4 Multicast Routing Best Current Practices.

Only one copy of a multicast packet is sent or received from the local network by the local IGMPv2 Designated Router.

Multicast transmission and reception is supported.

7.2 Native Multicast (IGMPv3)

Single Source Multicast [SSM]is generally implemented by a combination of IGMPv3 between the host and the network, and source-specific-tree-only PIM operations. Group addresses are in the 232.0.0.0/8 range, and any transmission within that range is source-specific.

Within that context, InterfaceDiscovery, GroupTransmit, GroupJoinAny, GroupReceive, and GroupLeave operate the same way as described in the Native Multicast section. In addition, IGMPv3 supports the GroupJoinOnly operation.

Implementation of this approach requires that the host, and the host network, be made IGMPv3-aware.

If properly configured by the network administrator, it is compatible with IPv4 Multicast Routing Best Current Practice.

Only one copy of a multicast packet is sent or received from the local IGMPv3 Designated Router.

Multicast transmission and reception is supported.

7.3 PIM-speaking Host Implementation of Core Operations

The design and implementations of IGMPv2 predate the PIM routing protocol. However, one could imagine a host that ignores IGMP and speaks PIM directly with routers on the local subnetwork.

InterfaceDiscovery would have to be hard-coded or otherwise discovered by the implementation. The application or host would implement the PIM protocol instead of the IGMP protocol. This might mean the host would have to discover or be configured with the appropriate upstream PIM speaker (PIM Designated Router) for the local subnetwork.

GroupTransmit would be accomplished by encapsulating the application data into a PIM Register message and transmitting that encapsulated data via unicast to the PIM Rendezvous Point. This would happen relatively infrequently; most of the data would be discarded until a PIM Join was received. Given a PIM Join, the application would transmit directly onto the local subnetwork.

JoinOnly would be accomplished by sending an (S,G) PIM Join towards the local PIM Designated Router, refreshed appropriately. JoinAll would be accomplished by sending a (*,G) PIM Join towards the local PIM Designated Router, and then sending (S,G) PIM Joins when datagrams arrive with the PIM SPT bit set.

GroupReceive is done by the host executing the equivalent of a UNIX recvfrom() call, with the from address set to the Group address.

A variant of GroupSize is accomplished as a side effect. Until a PIM Join arrives, the application can assume that no receiver has expressed an interest in the data at full rate, so the application need not even generate the data for transmission, much less transmit it on the local subnetwork.

Implementation of this approach requires that the host and at least one router on the local network speak the PIM protocol.

It is consistent with IPv4 Multicast Routing Best Current Practice.

Only one copy of a multicast packet is sent or received between the host and its PIM Neighbor, except for the occasional Register packet transmitted to the PIM Rendezvous Point.

Multicast transmission and reception is supported.

7.4 TCP-based Tunnel Implementation of Core Operations

Consider a hypothetical tunneling protocol that bypasses the local subnetwork of a host with an interested application. Multicast packets are transmitted from a tunnel host participating in the native IP multicast core of the internetwork, over a TCP channel, to the application on a host.

InterfaceDiscovery requires configuration on both the tunnel host and the application host. (This configuration might be automated.) The application host needs to know what IP address to use to source packets, so that the internetwork core knows to get those packets from the tunnel host.

GroupTransmit would be accomplished by encapsulating the application data into the TCP channel and transmitting it to the tunnel host. If the TCP transmit buffer is full, the application data would be discarded.

GroupJoinAny, GroupJoinOnly, and GroupLeave would be accomplished by a special message through the TCP channel, where the application host notifies the tunnel host of its interest (or disinterest) in data from various groups (and possibly specific sources.)

GroupReceive would be accomplished by the tunnel host encapsulating data into the TCP channel towards the application host. If the TCP transmit buffer is full, the application data would be discarded. The application host would decapsulate the data from the TCP stream and deliver it to the application in datagram form.

TCP is preferred in this scenario because of its ability to sense congestion and back off appropriately.

Implementation of this approach requires that the application host implement the TCP-Based Tunnel protocol, and that the application host be associated with at least one tunnel host. It does not require that the application host subnetwork be configured for multicast in any way.

Given proper configuration of the tunnel host, it is consistent with IPv4 routing Best Current Practice.

If there are multiple application hosts on a given subnetwork, multiple copies of a multicast packet may be carried to that subnetwork from the tunnel host (one each per TCP tunnel).

Multicast transmission and reception is supported.

7.5 Emcast Multicast Library

Emcast [EMCAST] is a user-level implementation of several of the Core Operations. It implements the GroupTransmit, GroupJoinAny, GroupLeave, and GroupReceive operations with library calls, and has hooks for handlers of various types. Currently supported handlers

include the traditional Internet Multicast, Banana Tree Protocol, and Internet Relay Chat.

GroupJoinOnly and InterfaceDiscovery are not supported.

Use of Emcast requires that the application support the Emcast library and at least one handler. If using the Banana Tree Protocol or Internet Relay Chat handlers, it does not require that the local subnetwork be configured for multicast in any way.

If Emcast was extended to support the InterfaceDiscovery operation, specifically giving the application an internetwork-unique multicast-enabled IP source address, then the non-traditional user-level multicast forwarding schemes (e.g. Banana Tree Protocol and Internet Relay Chat) could be made consistent with IPv4 routing Best Current Practice. This would of course require some gateway function between a Banana Tree Protocol or Internet Relay Chat domain and the IPv4 multicast core.

Multicast transmission and reception is supported.

7.6 UDP Multicast Tunneling Protocol (UMTP) and CastGate

The UDP Multicast Tunneling Protocol [UMTP] supports IPv4 multicast applications across a unicast-only internetwork. A server bridges traffic between the multicast-enabled and unicast-only portions of an internetwork. Only UDP multicast datagrams are supported.

GroupTransmit, GroupReceive, GroupJoinAny, and GroupLeave are supported through the UMTP protocol operation.

The InterfaceDiscovery and GroupJoinOnly operations are not supported.

Implementation of this approach requires a tunnel server connected to a multicast-enabled portion of the internetwork.

Multicast transmission and reception is supported; however, the transmitted UDP datagrams from a tunnel client are rewritten to appear as though they were sourced from the tunnel server.

CastGate [CASTGATE] provides a general mechanism for UMTP tunnel clients to automatically discover and initiate connections to UMTP tunnel servers.

8 Security Considerations

Multicast technology is subject to various denial-of-service attacks.

Certain host-based implementation of IP multicast forwarding are subject to security vulnerabilities; especially when insecure

services are delivered IP multicast packets without first executing a GroupJoinAny or GroupJoinOnly operation.

9 Acknowledgements

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

David Helder provided useful comments and suggestions.

10 References

[MCAST] RFC 1112: Host extensions for IP multicasting. S.E. Deering. August 1989.

[SVCMODEL] draft-nickless-mcast-svc-model-00.txt. B. Nickless. February 2002.

[RFC2119] RFC 2119: Key Words for use in RFCs to Indicate

Requirement Levels. S. Bradner. March 1997.

[IGMPv2] Internet Group Management Protocol, Version 2. W. Fenner. RFC 2236.

[SSM] draft-ietf-ssm-overview-02.txt. S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockwell, J. Meylor, D. Meyer, G. Shepherd, B. Haberman. December 2001.

[EMCAST] http://www.junglemonkey.net/emcast by David A. Helder

[UMTP] http://www.live.com/umtp.txt by Ross Finlayson

[CASTGATE] draft-liefooghe-castgate-01.txt. P. Liefooghe. November 2001.

11 Author's Address

Bill Nickless
Argonne National Laboratory
9700 South Cass Avenue #221   Phone: +1 630 252 7390
Argonne, IL 60439   Email: nickless@mcs.anl.gov