Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

D. Farinacci
lispers.net
L. Giuliano
HPE
M. McBride
Futurewei
N. Warnke
Deutsche Telekom

2 March 2026

Multicast Lessons Learned from Decades of Deployment Experience
draft-ietf-pim-multicast-lessons-learned-08

Abstract

This document gives a historical perspective about the design and deployment of multicast routing protocols. The document describes the technical challenges discovered from building these protocols. Even though multicast has enjoyed success of deployment in special use-cases, this draft discusses what were, and are, the obstacles for mass deployment across the Internet. Individuals who are working on new multicast related protocols will benefit by knowing why certain older protocols are no longer in use today.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 September 2026.

Copyright Notice

Table of Contents

1. Introduction

In the 1980’s, Steve Deering developed a multicast service model where packets from a group will only be received if explicity requested by a receiver. Over the next several decades, many multicast protocols, related drafts and RFC’s were built around IPv4, IPv6, tunnel and label based solutions. These protocols include DVMRP [RFC1075], PIM-DM [RFC3973], PIM-SM [RFC7761], PIM-BIDIR [RFC5015], PIM-SSM [RFC4607], MSDP [RFC3618], MBGP [RFC2858], MVPN [RFC6513], P2MP RSVP-TE [RFC4875], MLDP [RFC6388], BIER [RFC8279], LISP [RFC6830], MOSPF [RFC1584] IGMP [RFC2236], MLD [RFC3810] and several others. Perhaps due to these many multicast protocols, and their perceived complexity over unicast, there has been much angst over deploying IP Multicast over the last 30 years. It is not uncommon, with technical topics on multicast routing, for the discussion to evolve into what makes up a multicast address, whether that address identifies the source content or the set of receivers, does multicast create too much state on the network, why hasn’t it captured the heart of the internet, why is it so complicated, what’s the best multicast protocol to use, amongst many other questions. Despite the existence of multicast related BCPs, the authors felt it important to have a draft which helps answer some of these questions through identifying the lessons learned from multicast development and deployment over the last 30 years. This draft attempts to explain the current, and future, state of multicast affairs by reviewing the distractions, hype and innovation over the years and what was learned from the evolution of IP Multicast.

1.1. Requirements Language

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].

2. Glossary

PIM: Protocol Independent Multicast

PIM-DM: PIM Dense Mode

PIM-SM: PIM Sparse Mode

PIM-BIDIR: PIM Bi-Directional

PIM-SSM: PIM Source Specific Multicast

DVMRP: Distance Vector Multicast Routing Protocol

MVPN: Multicast Virtual Private Network

MSDP: Multicast Source Discovery Protocol

MBGP: Multi-protocol Border Gateway Protocol

BIER: Bit Indexed Explicit Routing

IGMP: Internet Group Management Protocol

MLD: Multicast Listener Discovery

P2MP RSVP-TE: Point-to-Multipoint TE Label Switched Paths

MLDP: Multicast Label Distribution Protocol

MOSPF: Multicast OSPF

MBONE: Multicast Backbone

3. Lessons learned about IP Multicast over the last 30 years

Various topics are addressed, in this section, which are relevant enough to warrant a discussion around what was learned since their development. New designers may come up with multicast proposals and then hear from more experienced designers saying their proposals wont work and it’s already been attemped. It’s important to document history or past mistakes will be repeated. This draft will start with one of the original multicast routing protocols, that Steve Deering developed, called Distance Vector Multicast Routing Protocol (DVMRP).

3.1. DVMRP

DVMRP preserved Deering’s multicast service model by sending the multicast packets throughout the domain (having no router state) and then pruning where there was no interest. Pruning was the exception in most financial networks of the time because most people wanted the financial data. DVMRP computes its own routing table to determine the best path back to the source. DVMRP uses a distance-vector routing algorithm. This algorithm requires that each router periodically inform its neighbors of its routing table. DVMRP was a unicast routing algorithm but it had tree building messages which formed distribution trees which could be pruned. There are no join messages in DVMRP because the RPF-tree is the default distribution tree. The Mbone (Multicast backbone) was an experimental virtual network built on top of the Internet for carrying IP multicast traffic. The Mbone intended to minimize the amount of data required

for multipoint audio/video-conferencing. DVMRP formed the basis for the Mbone tunnels.

The flooding and pruning of DVMRP was a good initial solution but it was quickly realized that it wouldn’t scale when using increasingly higher bit rates for multicast content. Using the network to discover sources was also something originally thought to be a good idea but later discovered to be resource and state intensive. DVMRP is a flood and prune distance vector protocol, similar to RIP, that relied on a hop count and depended upon itself as a routing protocol to build the RPF table rather than using existing unicast routing tables to build the rpf table as, the later developed, PIM-SM does. DVMRP worked good for small scale deployments but began to suffer when deployed in larger multicast environments so a better interdomain solution, like PIM, was needed.

3.2. MOSPF

For customers running an ospf network, multicast extensions that went into the ospf unicast protocol were developed in early 90s. The IGMP reports coming from receivers provided Dykstra with the ability to find out the exact branches. When Dykstra is built, for a particular source sending to a group, you knew exactly what the tree was. The packets would only go where they needed to go using the link state database. Join messages, or any additional signalling, was not needed to build the branches of the tree. This is where S,G state was introduced by using a source tree only protocol similar to DVMRP. Everything was always rooted by the source. MOSPF would look at the forward metrics towards the receivers rather than the RPF. But MOSPF was limited to OSPF and lots of sources so could be a state problem. Multicast customers wanted multicast and had to get it working using the early routing protocols of RIP, IGRP and EIGRP. Redistribution was popular for unicast so developers needed to decide if it should also be created for multicast. But perhaps a multicast protocol should be created that is independent of unicast protocols. It could even work interdomain using BGP and have a distribution tree across domains. This seemed like the right thing to do because brokerage firms, that only had 5-10% pruning, wanted multicast to flood everywhere and be able to perform RPF on whatever unicast protocols were in use. PIM dense mode was incrementally developed from DVMRP.

3.3. Shared vs Source Trees

With PIM shared trees, all sources send to a root of a shared distribution tree called the Rendezvous Point (RP). When multicast group members join a group, they cause branches of the distribution tree to be appended to the existing shared tree. New sources that send to the multicast group, send their traffic to the RP so existing receivers can receive packets. The path multicast packets take, are from the source encapsulated to the RP and then natively sent on the shared-tree branches. When a better/shorter path is desired, the source tree can be built. A source-tree is a multicast distribution tree routed at the source. As receivers on the shared-tree discover new sources, they join those sources on the source tree. The path on the source tree is determined by routing table lookups for the source's unicast address and is also known as the 'RPF path'. These lookups can use the actual unicast routing table, or include unicast routing data specifically intended for determining the multicast RPF path. With source trees, on the other hand, multicast traffic bypasses the RP and instead flows from the multicast source down the tree towards the receivers using the multicast forwarding table and the shortest available path. There is machinery to allow the multicast data to switch from the shared tree to a source tree once the source is discovered. Shared trees were designed to reduce state at a time when memory was scarce and expensive, while shortest path trees were simpler, and more optimal, but consumed more state.

Utilizing the network to provide the discovery of sources and receivers, and the machinery necessary to provide it, was an important development at the time. But there was no way to discover sources when adhering to this Deering model, The Deering model was like an ethernet and sources could just send and receivers would just receive the packets. When Deering augmented multicast routing, the receivers then needed to be discovered, so he added IGMP. But then he decided to not have source discovery and as he continued developing the model, he added DVMRP where the sources still didn't need to be discovered because their packets would flow down a default distribution tree and then later pruned the per-group tree so packets wouldn't flow where there were no receivers. When PIM was built, the designers wanted to change the default behavior to where the multicast packets would go nowhere and hence explicit joins built a tree. The flood-and-prune problem that DVMRP had needed to be solved. That problem was fixed but didn't provide any explicit signaling from the source to discover them. So the multicast routing protocol discovered the sources (via the PIM shared-tree).

Having two types of trees was the hard part. Switching from one tree (shared) to the other (source) was a difficult routing distribution problem. Because as you joined the source-tree, you had to prune

that source from the shared-tree so duplicates wouldn't continue for a long time. As protocol designers and implementors, that was a challenge to get right. It was later realized that source trees were needed which were ableto discover the multicast source outside of the network thus removing the source discovery burden from the network. Source-discovery originally had to be performed in the network because the multicast service model did not have a signaling mechanism like with SSM and IGMPv3.

PIM Sparse Mode is the most commonly used multicast routing protocol. PIM Dense Mode is also used. Bidirectional PIM is less widely used. With PIM-BIDIR, there are no source-based trees and no (S,G) state. There is no option for routers to switch from a shared tree to a source-based tree. The forwarding rules are much more simple than in PIM-SM and there are no data-driven events in the control plane. The main advantage of PIM-BIDIR is scaling. It scales well when there are many sources for each group such as with videoconferencing and many to many financial applications. However, with the lack of source-based trees, the traffic is forced to remain on the inefficient shared tree. There has been a lack of vendor support for PIM-BIDIR features such as IPv6 and MVPN along with TCAM scalability limitations.

During this process it was learned that PIM-SM (or more generally ASM (Any Source Multicast)) is more susceptible to DoS attacks by unwanted sources than is PIM-SSM. And address allocation with ASM is much more restrictive than it is with PIM-SSM.

3.4. IGMP

IGMPv1 was the first protocol to allow end hosts to indicate their interest in receiving a multicast stream. There was no message to indicate the receiver has left receiving the multicast stream so the router had to eventually figure it out. This caused bandwidth problems especially when quickly changing channels. IGMPv2 provided a leave message to prevent wasted bandwidth. And IGMPv3 provided support for source specific multicast. IGMPv1 and IGMPv2 do not have the capability to specify a particular sender of multicast traffic. This capability is provided in IGMPv3.

Multicast Listener Discovery (MLD) provides the corresponding functionality for IPv6. MLDv1 is functionally similar to IGMPv2 and MLDv2 aligns with IGMPv3 including support for SSM.

In hindsight ASM could have been easily developed with IGMPv2 from the start. All an (S,G) is, is a longer group address. If IGMPv2 was changed to have a more general encoding, IPv6 groups, IPv6 (S,G), and IPv4 (S,G) encoding would have been all created at the same time.

And, if it was made a library, it would have likely been deployed faster. Additionally, because "Integrated IS-IS" and "IPv6" were being worked on at the same time, one protocol could have been developed - similar to how BGP works today. PIM was integrated but it was developed as "ships in the night" with other protocols.

3.4.1. IGMP/MLD Snooping

IGMP and MLD snooping are commonly implemented in Layer 2 switches to limit multicast flooding within a VLAN by observing membership reports exchanged between hosts and routers. When operating correctly, snooping can significantly reduce unnecessary multicast replication in bridged domains.

Operational experience shows that snooping introduces a dependency on the presence of a functioning querier. In the absence of an active IGMP or MLD querier, group state will age out and multicast traffic can be unintentionally blackholed. Ensuring that each multicast enabled VLAN has a stable querier has proven to be a fundamental deployment requirement.

Snooping behavior during topology changes has also been a source of transient outages. Events such as link failures, spanning tree reconvergence or switch reloads may invalidate learned state. Implementations vary in recovery behavior and delayed re-learning of memberships has resulted in service interruption.

Additional interoperability issues have been observed in some deployments, including incorrect handling of report suppression, version differences or link-local multicast groups. Operators are advised to review the considerations in [RFC4541] and validate snooping behavior under failure conditions.

3.5. Data Driven State Creation and RPF

When a router, with a directly connected source (First Hop Router), receives the first multicast packet of a stream, it selects an optimal route from the unicast routing table based on the source address of the packet. The outbound interface of the unicast route, towards the source, is the RPF interface, and the next hop of the route is the RPF neighbor. The router compares the inbound interface of the packet with the RPF interface of the selected RPF route. If the inbound interface is the same as the RPF interface, the router considers that the packet has arrived on the correct path from the source and forwards the packet downstream. If a router does a lookup in the unicast routing table to perform an RPF check on every multicast data packet received, system resources would be overwhelmed. To save system resources, a router first performs a

lookup for the matching (S, G) entry after receiving a data packet sent from a source to a group. If no matching (S, G) entry is found, the router performs an RPF check to find the RPF interface for the packet. The router then creates a multicast route with the RPF interface as the upstream interface towards the source and delivers the route to the multicast forwarding information base (MFIB). If the RPF check succeeds, the inbound interface of the packet is the RPF interface, and the router forwards the packet to all the downstream interfaces in the forwarding entry. If the RPF check fails, the packet has been forwarded along an incorrect path, so the router drops the packet. The RPF is a functional requirement of PIM but it has caused some problems. When there are RPF changes, inconsistencies in the MFIB are created which can cause forwarding failures. Problems may occur when hosts (not ip forwarders) are also configured with RPF check. It is important to note that SSM doesn’t have the data-driven state creation described above. It’s also important to note the subtle difference between a "state problem" and a "state problem on a particular platform from a particular vendor".

PIM runs on a control-plane processor where the multicast routing table is maintained, and (S,G) state is downloaded to data-plane hardware forwarders. Whenever there is an RPF change, all routes that had changed in the multicast routing table have to get updated to the hardware forwarders.

3.6. MSDP

In PIM-SM, there can be only one active RP for a given group. MSDP was created to enable interdomain support for ASM given this requirement of PIM-SM. MSDP [RFC3618] is a protocol that enabled RPs to exchange information about active sources with one another. Operators at the time did not want to rely on a 3rd party for RP service and it was important to them to be able to own and manage their own independent RPs.

In addition to connecting RPs between domains, MSDP also allowed operators to run multiple RPs within a domain. Anycast addressing was used for these RPs to circumvent the aforementioned PIM-SM requirement that there can be only one active RP for a given group. So by sharing the same IP address, Anycast RP using MSDP [RFC3446]allowed multiple RPs within a domain to be active for a given group range, which enabled redundancy, load-balancing and localization, as sources and receivers within a domain could use the topologically closest RP (and re-route to the next closest RP if it failed).

MSDP deployment revealed a number of operation challenges. First, MSDP peers rely on a mechanism called Peer-RPF to select the correct peer from which to accept an MSDP Source Active (SA) message. Peer-RFP uses a complex set of forwarding rules that compare the originating RP in the SA message to the peer from whom it was received. Troubleshooting Peer-RPF problems was often an exceedingly cumbersome process.

Next, MSDP was susceptible to SA Storms, which plagued Internet multicast deployments in the early 2000s. SA Storms were usually caused by Internet worms that would infect a host and propagate by using these infected hosts to discover and attack other vulnerable hosts. To discover other vulnerable hosts, they typically selected a large block of addresses at random and port scanned all hosts in that block. In just a few minutes, each infected host could scan hundreds of thousands of other hosts. Unfortunately, many of these worm coders were sloppy and didn’t confine this random selection to unicast ranges. That is, there was the potential that multicast address blocks might be selected as destination addresses for these port scans. For example, imagine a worm-infected host selected a /16 of valid multicast address space and sent a single packet to every address in that block in an attempt to discover other vulnerable hosts. If that host was on a multicast-enabled network, its FHR would send a PIM register for each group address to its local RP, which would then send an MSDP SA for each group to all its MSDP peers. In a matter of minutes, 65k SAs would be flooded across all the MSDP-speakers on the Internet. To make matters even worse, in order for MSDP to support bursty source applications, originating RPs would encapsulate and include the first data packet of the flow within the MSDP SA message. These encapsulated data packets would generate forwarding entries in the FIBs of the routers, in addition to control plane entries for SAs. This state explosion quickly caused MSDP-speaking routers to run out of memory and crash. Ironically, these attacks weren’t even intentionally targeting the multicast infrastructure.

Implementations would later add throttling and policing mechanisms to protect against SA Storms, but by that time many operators had lost confidence in Internet Multicast and found deployment to not be worth the effort and risk. When an interdomain ASM solution for IPv6 was sought, there was no appetite in the IETF for adding IPv6 support to MSDP. Instead, Embedded RP [RFC3956]was created to enable interdomain ASM IPv6 capabilities by embedding the RP address into the actual group address, thereby obviating the need for RPs in different domains to exchange information on active sources. Combined with Anycast RP using PIM [RFC4610], which used PIM registers instead of MSDP SAs to exchange active source information between a set of Anycast RPs, the two functions of MSDP (connecting RPs within and between domains) were completely replaced in IPv6.

The story of MSDP is replete with teachable moments and lessons to be learned. MSDP has often been blamed for the most egregious of multicast’s challenges. Indeed, Peer-RPF was exceedingly complex and data encapsulation of the first packet of a flow to support bursty source applications was likely not worth the cost of forwarding plane resources. But ultimately, MSDP may have been a scapegoat- any protocol that attempted to create a synchronized database of every active multicast source on the Internet in every RP would likely have faced the same problems. The ultimate root cause of the problem was the assumption that the network should be responsible for source discovery; MSDP was merely a symptom. Thus, the ultimate solution is SSM, which completely eliminates the need for MSDP since source discovery is not done by the network (typically, handled by the application layer instead).

Additionally, MSDP provides an interesting counterpoint to the once-heated arguments on "BGP Overloading." That is, over the years there have been concerns about overextending BGP with functionality and capabilities that create risk in a protocol so critical to the Internet’s infrastructure. MSDP is an example of a new protocol that was created instead of extending BGP (at the time; it would be added to BGP for MVPNs [RFC6514]. The key observation is that a separate protocol likely didn’t yield any better results; ultimately, it’s not the name of the protocol, nor in what protocol the functionality resides, but rather what the functionality ~does~ that will determine how well it performs.

Another interesting observation is that MSDP remains in Experimental status, proving the durability of what was once considered a temporary solution. Further it illustrates the point that specification status does not tend to have an impact on deployment, as MSDP remains in many networks today since it is the only way to support multiple IPv4 RPs in different domains (and is probably the most popular way of connecting multiple Anycast RPs within a domain).

3.7. MPLS MVPNs

Multicast was not originally supported with MPLS. That is a lesson learned in and of itself. The workaround was point-to-point GRE tunnels from CE to CE which was not scalable when having many CE routers. MVPN solutions were complicated at times in the ietf. The MVPN complexity was organic because PE based unicast VPNs were already deployed. So it didn't allow for simpler multicast designs. The architecture was already built, multicast functionality was an incremental add-on, which made it easier to deploy but the cost of running the service was the same, or worse, than running unicast VPNs. There were years of debate about PIM based draft-rosen mvpn vs bgp based mvpn using P2MP RSVP-TE. Cisco wound up progressing an independent submission with [RFC6037] because it defined procedures which predated the publication of IETF mvpn standards, and these procedures differ in some respects from a fully standards-compliant implementation. Eventually the pim and bgp based mvpn solutions were progressed together in Multicast in MPLS/BGP IP VPNs in [RFC6513]. Perhaps one lesson learned here is that there will often be a conflict between providing timely implementations for customer needs vs waiting for the untimeliness of standards to work themselves out. A combined draft from the beginning, providing multiple multicast vpn solutions, would have been helpful in preventing years of conflict and non standard compliant solutions. Another lesson is that it was good to decouple the control plane from the data plane so that the control plane could scale better and the dataplane could have more options. Tunnels may now be built by PIM (any flavor), Multicast LDP (p2mp or mp2mp), RSVP-TE p2mp and we can map multiple provider multicast service interface's (PMSI) onto one aggregated tunnel.

3.8. SD and SDR

SD and SDR were good initial applications but we didn' t go far enough with them to help source discovery since the app layer is indeed a better place to handle source discovery (than the network). SDR is a session directory tool designed to allow the advertisement and joining of multicast streams particularly targeted for the Mbone. The Mbone (multicast backbone) was an experimental backbone and virtual network built on top of the Internet for carrying IP multicast traffic. The Session Directory Revised tool (SDR) was developed to help discover the group and port used for a multicast multimedia session. The original Session Directory (SD) tool was written by Lawrence Berkley Labs and was replaced by SDR. SDR is a multicast application that listens for SAP packets on a well known multicast group. These SAP packets contain a session description, the time the session is active, its IP multicast group addresses, media format, contact person and other information about the advertised multimedia session. In hindsight we should have continued

developing SDR to more fully help with source discovery perhaps by utilizing http. That would have been better than focusing on the network to provide multicast source discovery.

3.9. All or Nothing Problem

For multicast to function, every layer 3 hop between the sourcing and receiving end hosts must support a multicast routing protocol. This may not be a difficult challenge for enterprises and walled-garden networks where the benefits of multicast are perceived to be much greater than the costs to deploy (eg, financial, video distribution, MVPN SPs, etc). However, on the global Internet, where the cost/benefits of multicast (or any service, for that matter) are not likely to ever be universally agreed upon, this "all or nothing" requirement tends to create an insurmountable barrier. It should be noted that IPv6 suffers the same challenge, which explains why IPv6 has not been ubiquitously deployed across the Internet to the same degree as IPv4, despite decades of trying. Simply put, any technology that requires new protocols to be enabled on every interface on every router and firewall on the Internet is not likely to succeed. One approach to address this challenge is to develop solutions that facilitate incremental deployment and minimize/eliminate the need for coordination of multiple parties. Overlay networking is one such approach and allows the service to work for end users without requiring every underlay hop to support multicast-only the layer 3 hops in the overlay topology require multicast support. For example, AMT [RFC7450] allows end users on unicast-only networks to receive multicast content by dynamically tunneling to devices (AMT Relays) on multicast-enabled networks. Another example is Locator/ID Separation Protocol (LISP) [RFC8378], where multicast sources and receivers can be on the overlay and work with a any combination of unicast and/or native multicast delivery from the underlay. Endpoint identifiers (EIDs) are assigned to end hosts. Routing locators (RLOCs) are assigned to devices (primarily routers) that make up the global routing system.The LISP overlay nodes can roam while keeping their same EID address, can be multi-homed to load-split packets across multiple interfaces, and can encrypt packets at the overlay layer (freeing applications from dealing with security).

3.10. AMT and TreeDN

Automatic Multicast Tunneling (AMT) [RFC7450] allows end users, on unicast-only networks, to receive multicast content by dynamically tunneling to devices (AMT Relays) on multicast-enabled networks. AMT empowers interested end users to enjoy the service while also enabling content providers and operators, who have deployed multicast, to realize the benefits of more efficient delivery while

tunneling over the parts of the network (last/middle/first mile) that haven't deployed multicast. Further, this incremental approach can provide the necessary incentive for operators who haven't deployed multicast natively to do so in order to avoid carrying duplicate tunneled traffic.

TreeDN [I-D.ietf-mops-treedn] is a tree-based CDN architecture that leverages AMT. TreeDN is essentially the synthesis of SSM plus overlay networking technologies like AMT. TreeDN is designed to address the scaling challenges of live streaming to mass audiences. TreeDN enables operators to offer Replication-as-a-Service (RaaS) at a fraction the cost of traditional, unicast-based CDNs- in some cases, at no additional cost to the infrastructure. In addition to efficiently utilizing network resources to deliver existing multi-destination traffic, this architecture also enables new types of content and use cases that previously were not possible or economically viable using traditional CDN approaches. TreeDN is a decentralized architecture and a democratizing technology for content distribution.

TreeDN has several advantages over traditional unicast-based CDN approaches. First, the TreeDN functionality can be delivered entirely by the existing network infrastructure. Specifically, for operators with routers that support AMT natively, multicast traffic can be delivered directly to end users without the need for specialized CDN devices, which typically are servers that need to be racked, powered and connected to revenue-generating ports on routers. In this way, SPs can offer new RaaS functionality to content providers at potentially zero additional cost in new equipment (modulo the additional bandwidth consumption).

3.11. Network Based Source Discovery

In ASM, the network is responsible for discovering all multicast sources. This responsibility leads to massive protocol complexity, which imposes a huge operational cost for designing, operating and troubleshooting multicast. In SSM, source discovery is moved out of network and is handled by some sort of out-of-band mechanism, typically in the application layer. By eliminating network-based source discovery in SSM, we eliminate the need for shared trees, PIM register message encap/decap, RPs, SPT-switchover, data-driven state creation and MSDP, and the resulting protocol, PIM-SSM, is dramatically simpler than previous ASM routing protocols. Indeed, PIM-SSM is merely a small subset of PIM-SM functionality. The key insight is that source discovery is not a function the network should provide. One would never expect ISIS/OSPF and BGP to discover and maintain a globally synchronized database of all active websites on the Internet, yet that is precisely what is required of PIM-SM and

MSDP for ASM. This insight can apply more generally to other functions, like accounting, access control, transport reliability, etc. One simple heuristic for whether a function should exist in the multicast routing protocol is to simply ask what would unicast do (WWUD)? If unicast routing protocols like OSPF, ISIS or BGP do not provide such a function, then multicast routing protocols like PIM should not be expected to provide that function either. Further, moving functionality to the application layer, rather than in the network layer, allows allows faster innovation and greater levels of creativity, as these two layers tend to have vastly different requirements, expectations (and, therefore upgrade cycles) for stability, scale, functionality and innovation.

3.12. Dynamic Multicast Group Address Allocation

Approaches to multicast group allocation have been proposed in the past including mDNS [RFC6762], MADCAP [RFC2730], MASC [RFC2909], along with the related IPv6 Allocation Guidelines [RFC3307]. The problems are that they require manual configuration, are used on a single subnet, are not decentralized and have problems associated with address collisions. MADCAP, for instance, is used to dynamically assign addresses but its reliance on a dedicated server results in a single point of failure which is not acceptable for many environments. It is also susceptible to link-layer address collisions. It was later determined that multicast addresses really should be dynamically assigned by a decentralized, and zero configuration, protocol for many of todays environments.

Two solutions have recently emerged to improve upon the limitations of these previous solutions. One is Zeroconf Multicast Address Allocation Protocol [I-D.ietf-pim-ipv6-zeroconf-assignment] and the other is Group Address Allocation Protocol (GAAP) [I-D.ietf-pim-gaap]. Zeroconf works on any ipv4/v6 network (although the initial effort has focused on IPv6 marine networks). It delivers zero configuration, is decentralized and provides active collision detection by using a random number and saving state between power cycles. Applications randomly assign multicast group IDs from a reserved range and prevent collisions by using mDNS to publish records in a new "eth-addr.arpa" special-use domain. GAAP also works on any IPv4/v6 network, is zero configuration, decentralized, and provides active collision detection using a hashing algorithm. Instead of extended an existing protocol, as with Zeroconf, GAAP is a new lightweight protocol having a well-known UDP port number for the GAAP protocol. It allocates one v4 and one v6 multicast address that GAAP uses for messaging and allocates a multicast address block for GAAP allocated group addresses.

3.13. Reliable Multicast

IP Multicast uses UDP [RFC768] which is connectionless, best effort delivery, ie, it does not guarantee the delivery of data packets. Packets may be dropped, delivered multiple times, or delivered out of order. The primary goal of multicast is to deliver data to a group of recipients as quickly as possible and the lack of reliability in UDP is typically not a significant concern. But reliable IP multicast is a concern in situations where data needs to be distributed to a large number of receivers simultaneously, especially when data loss cannot be tolerated. The IETF had a, now concluded, RMT (Reliable Multicast Transport) working group whose purpose was to standardize reliable multicast transport. They developed several RFC's which fall under either a NACK-based protocol or an asychronous layered coding protocol that uses forward error correction. The base protocols produced, in this WG, are the NACK-Oriented Reliable Multicast (NORM) transport protocol [RFC5740] and the reliable content delivery protocol called the Asynchronous Layered Coding (ALC) Protocol [RFC5775]. NORM essentially provides a reliable data stream, over unreliable IP Multicast, by utilizing negative acknowledgments (NACKs) to request retransmissions of missing packets. ALC enables reilable delivery of content over IP multicast by each receiver potentially joining a data stream at different times and receiving data at their own pace based on their available bandwidth. Protocols, such as File Delivery over Unidirectional Transport (FLUTE) [RFC6726] were built on these base protocols.

The PIM working group later produced an experimental extension to PIM called the PIM over reliable transport (PORT) protocol [RFC6559] for the reliable transmission of PIM Join/Prune messages. This eliminates the need for periodic Join/Prune message transmission and processing by using TCP [RFC793] or SCTP [RFC4960]. PORT makes some fundamental changes to how PIM works in that Join/Prune state does not require periodic updates, and it partly turns PIM into a hard-state protocol. When a router supports this specification, it need not use the reliable transport mechanism with every neighbor. It can be negotiated on a per-neighbor basis.

3.14. Premature Optimization

Premature optimization can saddle the protocols with complexity burdens long after the optimizations are no longer relevant or even before the optimizations can be used. Typically those optimizations are implemented for scale even though you don't need or see a need for them in early deployments. But they must be thought ahead of time and planned for (that means designed and implemented up front). Shared trees were born in the 1990s out of a (well-founded at the time) concern for state exhaustion when memory was a scarce resource.

As memory got cheaper and more abundant, these concerns were reduced, but the complexity remained. It was once ironically noted that we eliminated the state problem by making the protocols so complex that no one deployed them. Although, to be fair, other protocols also have had state problems and private enterprises have successfully used multicast in their wall-gardens without state problems.

3.15. Kernel vs User Space

In hindsight, what we should have done with multicast is the same thing QUIC did which is implemented as a library rather than in the kernel. If we had done that, then when the app is deployed that needs a network function, it comes at the same time (inside the app). This is similar to what we have done with AMT in VLC which was a practical decision to get apps access to a native multicast cloud.

By packaging the protocol stack in the application, it allows a developer to add features and fix bugs quickly. And get the updates deployed quickly by having users download and update the app. This rather modern way of distributing new code has proved successful in may mobile and cloud based environments. With respect to multicast, we could have made faster deployed changes to IGMP as well as any tunneling technology we felt useful.

3.16. 802.11

We’ve learned many things over the years about the problems (such as high packet error rates, no acknowledgements and low data rates) with deploying multicast in 802.11 (Wi-Fi) networks. We even created [RFC9119] specifically to address all the many ways multicast is problematic over Wi-Fi. Performance issues, for instance, have been observed over the years, when multicast packets transmit over IEEE 802 wireless media, so much so that that it is often disallowed over Wi-Fi networks. Various workarounds have been developed including converting multicast to unicast at layer 2 (aka, ingress replication) in order to more successfully transit the wireless medium. There are various optimizations that can be implemented to mitigate some of the many issues involving multicast over Wi-Fi. The lesson we’ve learned now is that we (vendors, IETF) should have worked closely with the IEEE many years ago on detailing the problems in order to improve the performance of multicast transmissions at Layer 2. The IEEE is now designing features to improve multicast performance over Wi-Fi but it’s expensive to do so and will take time.

3.17. RPT-to-SPT Switchover Thresholds

A common implementation feature in PIM-SM is the ability to configure a bandwidth or packet rate threshold that triggers the switch from the shared tree (RPT) to the source tree (SPT). This is typically implemented as a configurable parameter such as "switch to SPT when exceeding X kbit/s of traffic for this (S,G)".

While the default behavior in most implementations is to switch to SPT immediately upon receiving the first data packet, this configurable threshold feature was designed to give operators control over when to build source trees for high-bandwidth flows. However, it has proven to be operationally challenging. The challenges arises from:

* The need to accurately measure traffic rates per (S,G) entry
* The state machinery required to trigger and manage the transition
* Ensuring consistent behavior across different router implementations
* The potential for flapping between RPT and SPT if thresholds are set too close to typical traffic patterns

While this feature provides granular control, many operators find the default immediate SPT-switchover behavior sufficient for their needs. The operational experience has shown that the costs of implementing and managing the threshold feature often outweigh the benefits for many deployment scenarios. Operationally, most networks set the SPT threshold to either 0 (immediate switchover) or infinity (no switchover).

This experience reinforces the broader lesson that features requiring per-flow state and measurement can introduce significant complexity with limited operational benefit. The fact that many deployments operate successfully with the simpler immediate SPT-switchover behavior suggests that the additional complexity of configurable thresholds was perhaps unnecessary for most use cases.

4. Conclusions

5. IANA Considerations

N/A

6. Security Considerations

7. Acknowledgement

Beau Williamson’s publications helped with some of the history of the protocols discussed. Toerless Eckert, Hitoshi Asaeda and David Lamparter provided excellent draft review comments.

8. References

8.1. Normative References

[RFC1075] Waitzman, D., Partridge, C., and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, DOI 10.17487/RFC1075, November 1988, <https://www.rfc-editor.org/info/rfc1075>.

[RFC1584] Moy, J., "Multicast Extensions to OSPF", RFC 1584, DOI 10.17487/RFC1584, March 1994, <https://www.rfc-editor.org/info/rfc1584>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[RFC2730] Hanna, S., Patel, B., and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, DOI 10.17487/RFC2730, December 1999, <https://www.rfc-editor.org/info/rfc2730>.

[RFC2858] Bates, T., Rekhter, Y., Chandra, R., and D. Katz, "Multiprotocol Extensions for BGP-4", RFC 2858, DOI 10.17487/RFC2858, June 2000, <https://www.rfc-editor.org/info/rfc2858>.

[RFC2909] Radoslavov, P., Estrin, D., Govindan, R., Handley, M., Kumar, S., and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", RFC 2909, DOI 10.17487/RFC2909, September 2000, <https://www.rfc-editor.org/info/rfc2909>.

[RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, DOI 10.17487/RFC3307, August 2002, <https://www.rfc-editor.org/info/rfc3307>.

[RFC3446] Kim, D., Meyer, D., Kilmer, H., and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)", RFC 3446, DOI 10.17487/RFC3446, January 2003, <https://www.rfc-editor.org/info/rfc3446>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <https://www.rfc-editor.org/info/rfc3618>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, DOI 10.17487/RFC3956, November 2004, <https://www.rfc-editor.org/info/rfc3956>.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, DOI 10.17487/RFC3973, January 2005, <https://www.rfc-editor.org/info/rfc3973>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <https://www.rfc-editor.org/info/rfc4541>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, DOI 10.17487/RFC4610, August 2006, <https://www.rfc-editor.org/info/rfc4610>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <https://www.rfc-editor.org/info/rfc4875>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <https://www.rfc-editor.org/info/rfc5015>.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, DOI 10.17487/RFC5740, November 2009, <https://www.rfc-editor.org/info/rfc5740>.

[RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, DOI 10.17487/RFC5775, April 2010, <https://www.rfc-editor.org/info/rfc5775>.

[RFC6037] Rosen, E., Ed., Cai, Y., Ed., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, DOI 10.17487/RFC6037, October 2010, <https://www.rfc-editor.org/info/rfc6037>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC6559] Farinacci, D., Wijnands, IJ., Venaas, S., and M. Napierala, "A Reliable Transport Mechanism for PIM", RFC 6559, DOI 10.17487/RFC6559, March 2012, <https://www.rfc-editor.org/info/rfc6559>.

[RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, DOI 10.17487/RFC6726, November 2012, <https://www.rfc-editor.org/info/rfc6726>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7450] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450, DOI 10.17487/RFC7450, February 2015, <https://www.rfc-editor.org/info/rfc7450>.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC793] Postel, J., "Transmission Control Protocol", RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[RFC8378] Moreno, V. and D. Farinacci, "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC 8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

[RFC9119] Perkins, C., McBride, M., Stanley, D., Kumari, W., and JC. Ziga, "Multicast Considerations over IEEE 802 Wireless Media", RFC 9119, DOI 10.17487/RFC9119, October 2021, <https://www.rfc-editor.org/info/rfc9119>.

8.2. Informative References

[I-D.ietf-mops-treedn]
Giuliano, L., Lenart, C., and R. Adam, "TreeDN- Tree-based CDNs for Live Streaming to Mass Audiences", Work in Progress, Internet-Draft, draft-ietf-mops-treedn-07, 21 August 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-mops-treedn-07>.

[I-D.ietf-pim-gaap]
Farinacci, D. and M. McBride, "Group Address Allocation Protocol (GAAP)", Work in Progress, Internet-Draft, draft-ietf-pim-gaap-10, 25 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-gaap-10>.

[I-D.ietf-pim-ipv6-zeroconf-assignment]
Karstens, N., Farinacci, D., and M. McBride, "Zero-Configuration Assignment of IPv6 Multicast Addresses", Work in Progress, Internet-Draft, draft-ietf-pim-ipv6-zeroconf-assignment-07, 30 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-ipv6-zeroconf-assignment-07>.

Authors' Addresses

Dino Farinacci
lispers.net
Email: farinacci@gmail.com

Lenny Giuliano
HPE
Email: lenny@juniper.net

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Nils Warnke
Deutsche Telekom
Email: Nils.Warnke@telekom.de