Workgroup: Network Working Group
Internet-Draft:
draft-mcbride-mboned-lessons-learned-01
Published: 24 October 2022
Intended Status: Informational
Expires: 27 April 2023
Authors: D. Farinacci   L. Giuliano   M. McBride
lispers.net   Juniper   Futurewei

Multicast Lessons Learned from Decades of Deployment Experience

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, we discuss what were, and are, the obstacles for mass deployment across the Internet.

Status of This Memo

This Internet-Draft will expire on 27 April 2023.

Copyright Notice

Table of Contents

1. Introduction

There are many multicast related drafts and RFC's 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. We attempt to better understand the current, and future, state of multicast affairs by reviewing the distractions, hype and innovation over the years and what we've 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

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

We will address various topics, in this section, which are relevant enough to warrant a discussion around what we've learned since their development. We will start with one of the original multicast routing protocols called Distance Vector Multicast Routing Protocol (DVMRP).

3.1. DVMRP

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 flooding and pruning of DVMRP was a good initial solution but we 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 we needed better solutions.

3.2. 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 the unicast routing table and is also known as the "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, we wanted to change the default behavior to where the multicast packets would go nowhere and hence explicit joins built a tree. We had to fix the flood-and-prune problem that DVMRP had. We fixed that problem 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. What we then later realized was that we needed source trees which 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 we now have with SSM and IGMPv3.

During this process we also 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.

3.3. 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 security feature 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.

3.4. 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. We had 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.

3.5. 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.

3.6. 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 multicastonly 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. This 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. 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).

3.7. 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.

3.8. 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.

3.9. 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.

3.10. 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.

In hindsight we could have easily developed SSM with IGMPv2 from the start. All an (S,G) is, is a longer group address. So if we changed IGMPv2 to have a more general encoding, we would have created IPv6 groups, IPv6 (S,G), and IPv4 (S,G) encoding all at the same time. And, if we had made it a library, it would have likely been deployed faster. Additionally, because we were working on "Integrated IS-IS" and "IPv6" all at the same time, we could have developed one protocol - similar to how we do it for BGP today. PIM was integrated but it was developed as "ships in the night" with other protocols.

3.11. 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.

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.

8. 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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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. Zúñiga, "Multicast Considerations over IEEE 802

Wireless Media", RFC 9119, DOI 10.17487/RFC9119, October 2021, <https://www.rfc-editor.org/info/rfc9119>.

Authors' Addresses

Dino Farinacci
lispers.net

Email: farinacci@gmail.com

Lenny Giuliano
Juniper

Email: lenny@juniper.net

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com