MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 16 June 2022

K. Kompella
R. Balaji
Juniper Networks
R. Thomas
Cohesity

13 December 2021

Label Distribution Using ARP
draft-kompella-mpls-larp-11

Abstract

This document describes extensions to the Address Resolution Protocol to distribute MPLS labels for IPv4 and IPv6 host addresses. Distribution of labels via ARP enables simple plug-and-play operation of MPLS, which is key to deploying MPLS in data centers and enterprises.

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 16 June 2022.

Copyright Notice

Table of Contents

1. Introduction

This document describes extensions to the Address Resolution Protocol (ARP) [RFC0826] to advertise label bindings for IP host addresses. While there are well-established protocols, such as LDP [RFC5036], RSVP [RFC3209], BGP [RFC3107] and SPRING-MPLS [RFC8660], that provide robust mechanisms for label distribution, these protocols tend to be relatively complex, and often require detailed configuration for proper operation. There are situations where a simpler protocol may be more suitable from an operational standpoint. An example is the case where an MPLS Fabric is the underlay technology in a Data Center; here, MPLS tunnels originate from host machines. The host thus needs a mechanism to acquire label bindings to participate in the MPLS Fabric, but in a simple, plug-and-play manner. Existing signaling/routing protocols do not always meet this need. Labeled

ARP (L-ARP) is a proposal to fill that gap.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The term "server" will be used in this document to refer to an ARP/L-ARP server; the term "host" will be used to refer to a compute server or other device acting as an ARP/L-ARP client.

1.2. Approach

ARP is a nearly ubiquitous protocol; every device with an Ethernet interface, from hand-helds to hosts, have an implementation of ARP. ARP is plug-and-play; ARP clients do not need configuration to use ARP. That suggests that ARP may be a good fit for devices that want to source and sink MPLS tunnels, but do so in a zero-config, plug-and-play manner, with minimal impact to their code.

The approach taken here is to create a minor variant of the ARP protocol, labeled ARP (L-ARP), which is distinguished by a new hardware type, MPLS-over-Ethernet. Regular (Ethernet) ARP (E-ARP) and L-ARP can coexist; a device, as an ARP client, can choose to send out an E-ARP or an L-ARP request, depending on whether it needs Ethernet or MPLS connectivity. Another device may choose to function as an E-ARP server and/or an L-ARP server, depending on its ability to provide an IP-to-Ethernet and/or IP-to-MPLS mapping.

2. Overview of Ethernet ARP

In the most straightforward mode of operation [RFC0826], ARP queries are sent to resolve "directly connected" IP addresses. The ARP request is broadcast, with the Target Protocol Address field (see Section 5 for a description of the fields in an ARP message) carrying the IP address of another node in the same subnet. All the nodes in the LAN receive this ARP request. All the nodes, except the node that owns the IP address, ignore the ARP request. The IP address owner learns the MAC address of the sender from the Source Hardware Address field in the ARP request, and unicasts an ARP reply to the sender. The ARP reply carries the replying node’s MAC address in the Source Hardware Address field, thus enabling two-way communication between the two nodes.

A variation of this scheme, known as "proxy ARP" [RFC2002], allows a node to respond to an ARP request with its own MAC address, even when the responding node does not own the requested IP address. Generally, the proxy ARP response is generated by routers to attract traffic for prefixes they can forward packets to. This scheme requires the host to send ARP queries for the IP address the host is trying to reach, rather than the IP address of the router. When there is more than one router connected to a network, proxy ARP enables a host to automatically select an exit router without running any routing protocol to determine IP reachability. Unlike regular ARP, a proxy ARP request can elicit multiple responses, e.g., when more than one router has connectivity to the address being resolved. The sender must be prepared to select one of the responding routers.

Yet another variation of the ARP protocol, called 'Gratuitous ARP' [RFC2002], allows a node to update the ARP cache of other nodes in an unsolicited fashion. Gratuitous ARP is sent as either an ARP request or an ARP reply. In either case, the Source Protocol Address and Target Protocol Address contain the sender's address, and the Source Hardware Address is set to the sender's hardware address. In case of a gratuitous ARP reply, the Target Hardware Address is also set to the sender's address.

3. L-ARP Protocol Operation

The L-ARP protocol builds on the proxy ARP model, and also leverages gratuitous ARP model for asynchronous updates.

In this memo, we will refer to L-ARP clients (that make L-ARP requests) and L-ARP servers (that send L-ARP responses). In Figure 1, H1, H2 and H3 are L-ARP clients, and T1, T2 and T3 are L-ARP servers. T4 is a member of the MPLS Fabric that may not be an L-ARP server. Within the MPLS Fabric, the usual MPLS protocols (IGP (i.e., SPRING-MPLS), LDP, RSVP-TE) are run. Say H1, H2 and H3 want to establish MPLS tunnels to each other (for example, they are using BGP MPLS VPNs as the overlay virtual network technology). H1 might also want to talk to a member of the MPLS Fabric, say T. Also, the "protocol" addresses in L-ARP requests are either IPv4 or IPv6 addresses; note that while it is common to use Neighbor Discovery (ND) [RFC4861] for "regular" ARP requests when dealing with IPv6 (i.e., to obtain Ethernet MAC addresses corresponding to an IPv6 address), ND is not used when the ARP request is for an MPLS label.

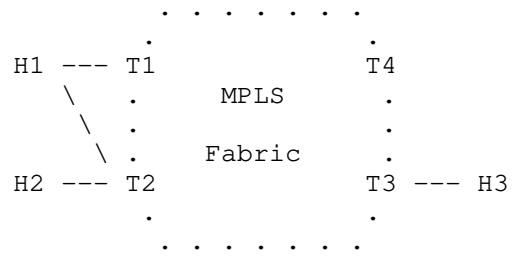

Figure 1: MPLS Fabric

3.1. Setup

In Figure 1, the nodes T1-T4, and those in between making up the "MPLS Fabric" are assumed to be running some protocol whereby they can signal MPLS reachability to themselves and to other nodes (like hosts H1-H3). T1-T3 are L-ARP servers; T4 need not be, since it doesn't have an attached L-ARP client. H1-H3 are L-ARP clients.

3.2. Egress Operation

A node (say T3) that wants an attached node (say H3) to have MPLS reachability allocates a label L3 to reach H3 and advertises this label into the MPLS Fabric. This can be triggered by configuration on T3, or when T3 first receives an L-ARP request from H3 (indicating that H3 wants MPLS connectivity), or via some other protocol. T3 then advertises (H3, L3) to its peers in the MPLS Fabric so that all members of the Fabric have connectivity to H3. This advertisement can be one of the following:

* a "proxy" LDP message (sent on behalf of H3) with prefix H3 and label L3; or

* a node Segment ID (SID) advertised on behalf of H3; or

* a labeled BGP advertisement, with prefix H3, label L3 and next hop self.

  On receiving a packet with label L3, T3 pops the label and send the packet to H3. (In the case of labeled BGP, there would be a two-label stack, with outer label to reach T3 and inner label of L3.) This is the usual operation of an MPLS Fabric, with the addition of advertising labels for nodes outside the fabric.

3.3. Ingress Operation

A node (say H1, an L-ARP client) that needs an MPLS tunnel to another node (say H3) identified by a host address (either IPv4 or IPv6) broadcasts over all its interfaces an L-ARP request with the Target Protocol Address set to H3 and Hardware Type set to "MPLS-over-Ethernet". A node receiving the L-ARP request (say T1, an L-ARP server) does the following:

1. checks if it has MPLS reachability to H3. If not, it ignores the L-ARP request.

2. if it does, T1 allocates a label TL3 to reach H3 (if it doesn't already have such a label) and installs an L-FIB entry to swap L1 with the label (stack) to reach H3.

3. sends a (proxy) L-ARP reply to H1 with the Source Hardware Address (SHA) set to (L, M), where M is T1's metric to H3. T1 may also set some attribute bits in the SHA.

3.4. Data Plane

To send a packet to H3 over an MPLS tunnel, H1 pushes L1 onto the packet, sets the destination MAC address to M1 and sends it to T1. On receiving this packet, T1 swaps the top label with the label(s) for its MPLS tunnel to H3. If T1's reachability to H3 is via a SPRING label stack, the label L1 acts as an implicit binding SID.

If H1 and H3 have an overlay connection (say an IPVPN [RFC4364] VPN-foo) whereby VM1 on H1 wishes to talk to VM3 on H3 over VPN-foo, H1 does the following:

1. H1 learns information about VPN-foo via BGP (or an SDN controller), including the VPN label VL3 to use to talk to VM3;

2. H1 installs a VRF for VPN-foo, with prefix VM3, label VL3 and next hop H3;

3. H1 binds the local "veth" interface to VM1 to this VRF.

4. When VM1 sends a packet to dest IP address VM3 over its veth interface, H1 looks up VM3 in the corresponding VRF, gets label VL3. It then sends an L-ARP request for next hop H3, and gets TL3.

5. Finally, H1 pushes the label pair (TL3, VL3) onto the packet from VM1 and sends this to T1. This packet will then end up at VM3 on H3.

Note that H1 broadcasts its L-ARP request over its attached interfaces. H1 may receive several L-ARP replies; in that case, H1 can select any subset of these to send MPLS packets destined to H3. As described later, the L-ARP response may contain certain parameters that enable the client to make an informed choice. If the target H3 belongs to one of the subnets that H1 participates in, and H3 is capable of sending L-ARP replies, H1 can use H3's response to send MPLS packets to H3.

4. Attributes

In addition to carrying a label stack to be used in the data plane, an L-ARP reply carries some attributes that are typically used in the control plane. One of these is a metric. The metric is the distance from the L-ARP server to the destination. This allows an L-ARP client that receives multiple responses to decide which ones to use, and whether to load-balance across some of them. The metric typically will be the IGP shortest path distance from server to the destination; this makes comparing metrics from different servers meaningful.

Another attribute is Entropy Label (EL) Capability. This attribute says whether the destination is EL capable (ELC). In Figure 1, if T3 advertises a label to reach H3 and T3 is ELC, T3 can include in its signaling to T1 that it is ELC. In that case, T1's L-ARP reply to H1 can have ELC bit set. This tells H1 that it may include (below the outermost label) an Entropy Label Indicator followed by an Entropy Label. This will help improve load balancing across the MPLS Fabric, and possibly on the last hop to H3.

4.1. Secondary Attributes

Beyond the basic attributes that are carried with every L-ARP request, there more optional attributes, for example, to ask for certain characteristics of the path traffic takes to the destination. These attributes are carried in TLVs that are carried in L-ARP requests and replies.

One such TLV is the Classful Transport (CT: see [I-D.kaliraj-idr-bgp-classful-transport-planes]) TLV. This TLV allows the L-ARP client to request a label to a destination over a tunnel of the given Transport Class. To satisfy this request, the L-ARP server creates (or finds) a tunnel to the destination that is routed over the CT Transport Plane, allocates a label L, inserts an entry in the LFIB to swap L to the tunnel, and sends L to the L-ARP client in its reply.

5. L-ARP Message Format

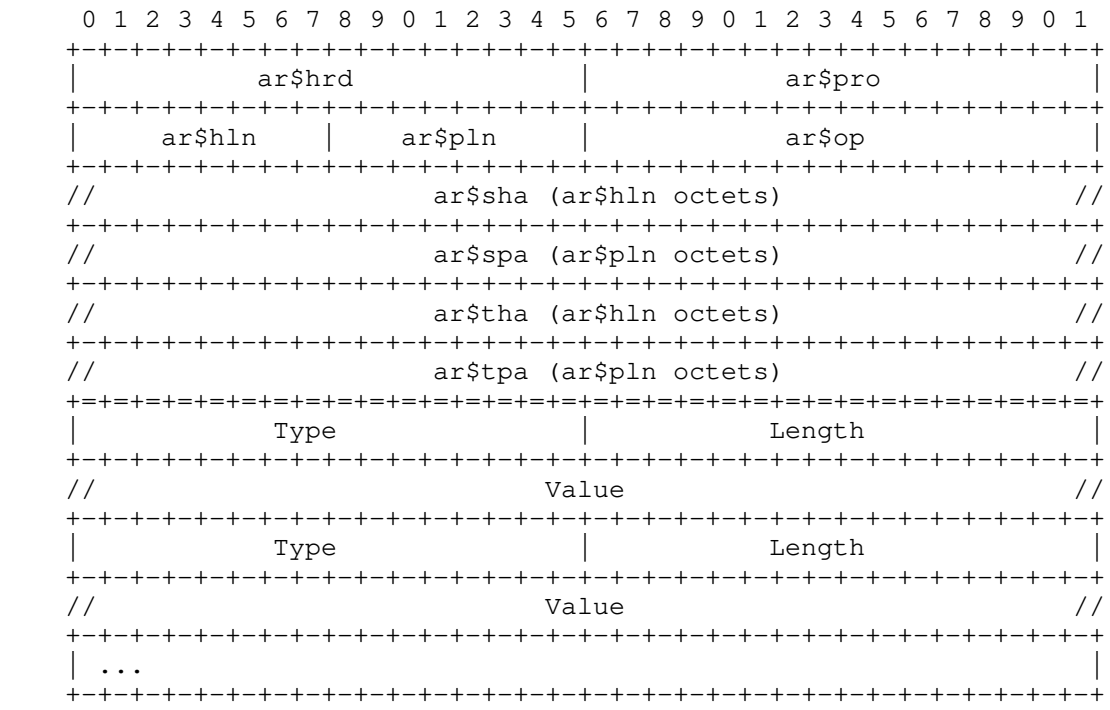

Figure 2: L-ARP Packet Format

ar$hrd: Hardware Type: MPLS-over-Ethernet. The value of the field used here is HTYPE-MPLS. To start with, we will use the experimental value HW_EXP2 (256).

ar$pro: Protocol Type: IPv4/IPv6. The value of the field used here is 0x0800 to resolve an IPv4 address and 0x86DD to resolve an IPv6 address.

ar$hln: Hardware Address Length: 6

ar$pln: Protocol Address Length: for an IPv4 address, the length is 4 octets; for an IPv6 address, it is 16.

ar$op: Operation Code: set to 1 for request, 2 for reply, and 10 for ARP-NAK. Other op codes may be used as needed.

ar$sha: Source Hardware Address: In an L-ARP request, this is usually all zeros. In an L-ARP reply, Source Hardware Address is the label to reach ar$spa, as specified in Figure 3 below.

ar$spa: Source Protocol Address: In an L-ARP request, this field carries the sender's IP address. In an L-ARP reply, this field carries the requested IP address (which may not be the sender's IP address).

ar$tha: Target Hardware Address: In an L-ARP message, this is all zeros.

ar$tpa: Target Protocol Address: In an L-ARP request, this field carries the IP address for which the client is seeking an MPLS label.

Type: a 2-octet field defining the Type of the TVL

Length: a 2-octet field defining the Length L of the TVL

Value: an L-octet field with the Value of the TLV

5.1. Hardware Address Format

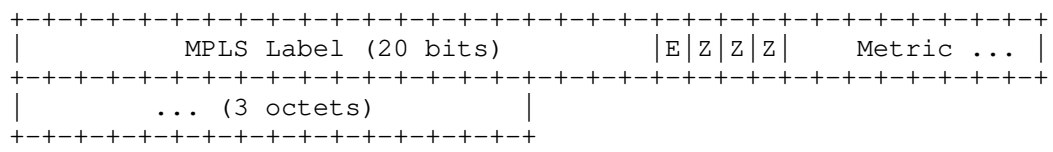

Figure 3: Label Format in L-ARP

MPLS Label: The 20-bit label

E-bit: Entropy Label Capable: this flag indicates whether the corresponding label in the label stack can be followd by an Entropy Label. If this flag is set, the client has the option of inserting ELI and EL as specified in [RFC6790]. The client can choose not to insert ELI/EL pair. If this flag is clear, the client must not insert ELI/EL after the corresponding label.

Z: These bits are not used, and SHOULD be set to zero on sending and ignored on receipt.

Metric: The 3-octet IGP metric to ar$tha from the point of view of the L-ARP replier.

5.2. CT TLV

The CT TLV has Type (TBD) and Length 4 octets; the Value field consists of the CT attribute.

6. Security Considerations

There are many possible attacks on ARP: ARP spoofing, ARP cache poisoning and ARP poison routing, to name a few. These attacks use gratuitous ARP as the underlying mechanism, a mechanism used by L-ARP. Thus, these types of attacks are applicable to L-ARP. Furthermore, ARP does not have built-in security mechanisms; defenses rely on means external to the protocol.

It is well outside the scope of this document to present a general solution to the ARP security problem. One simple answer is to add a TLV that contains a digital signature of the contents of the ARP message. This TLV would be defined for use only in L-ARP messages, although in principle, other ARP messages could use it as well. Such an approach would, of course, need a review and approval by the Security Directorate. If approved, the type of this TLV and its procedures would be defined in this document. If some other technique is suggested, the authors would be happy to include the relevant text in this document, and refer to some other document for the full solution.

7. IANA Considerations

IANA is requested to allocate a new ARP hardware type (from registry hrd) for HTYPE-MPLS [IANA].

IANA is further requested to create a registry for Types of L-ARP Secondary Attributes. This registry should contain an entry for the CT Type Section 5.2.

8. Acknowledgments

Many thanks to Shane Amante for his detailed comments and suggestions. Many thanks to the team in Juniper prototyping this work for their suggestions on making this variant workable in the context of existing ARP implementations. Thanks too to Luyuan Fang, Alex Semenyaka and Dmitry Afanasiev for their comments and encouragement.

9. References

9.1. Normative References

[IANA] IANA, "Address Resolution Protocol (ARP) Parameters/ Hardware Types", n.d., <https://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml>.

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC2002] Perkins, C., Ed., "IP Mobility Support", RFC 2002, DOI 10.17487/RFC2002, October 1996, <https://www.rfc-editor.org/info/rfc2002>.

[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>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[I-D.kaliraj-idr-bgp-classful-transport-planes] Vairavakkalai, K., Venkataraman, N., Rajagopalan, B., Mishra, G., Khaddam, M., Xu, X., Szarecki, R. J., and D. J. Gowda, "BGP Classful Transport Planes", Work in Progress, Internet-Draft, draft-kaliraj-idr-bgp-classful-transport-planes-12, 25 August 2021, <https://www.ietf.org/archive/id/draft-kaliraj-idr-bgp-classful-transport-planes-12.txt>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <https://www.rfc-editor.org/info/rfc3107>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, 94089
United States of America

Phone: +1-408-745-2000
Email: kireeti.ietf@gmail.com

Balaji Rajagopalan
Juniper Networks
Survey No.111/1 to 115/4, Wing A & B
Bangalore 560103
India

Email: balajir@juniper.net

Reji Thomas
Cohesity

Email: rejithomas.d@gmail.com