MPLS WG
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2016

K. Kompella
Juniper Networks
R. Balaji
Juniper Networks, Inc.
G. Swallow
Cisco Systems

September 8, 2015

Label Distribution Using ARP
draft-kompella-mpls-larp-04

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 a key goal of the MPLS Fabric architecture.

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 [RFC2119].

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.

Status of This Memo

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, RSVP and BGP, 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.

[TODO-MPLS-FABRIC] describes the motivation for using MPLS as the fabric technology.

1.1. 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 query is broadcast, with the Target Protocol Address field (see Section 9 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 query. All the nodes, except the node that owns the IP address, ignore the ARP query. 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, 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.

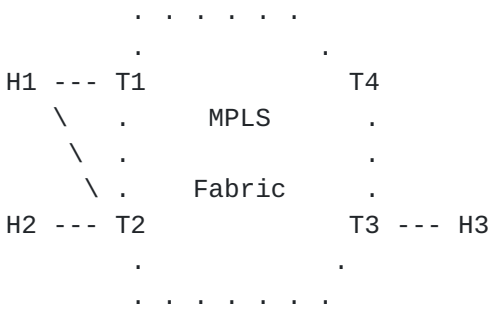


Figure 1

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 H1-H3). T1-T3 are L-ARP servers; T4 need not be. 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 via some other protocol. On receiving a packet with label L3, T3 pops the label and send the packet to H3. 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, the L-ARP client) that needs an MPLS tunnel to a node (say H3) identified by a host address (either IPv4 or IPv6) broadcasts over all its interfaces an L-ARP query with the Target Protocol Address set to H3. A node (say T1, an L-ARP server) that has MPLS reachability to H3 sends an L-ARP reply with the Source Hardware Address set to its Ethernet MAC address M1, with a new TLV containing a label L1. 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.

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, carried in the LST TLV, 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, if T1's L-ARP reply to H1 consists of a single label, T1 can set the ELC bit in the label field of the LST TLV. 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.

5. Client-Server Synchronization

In an L-ARP reply, the server communicates several pieces of information to the client: its hardware address, the MPLS label, Entropy Label capability and metric. Since ARP is a stateless protocol, it is possible that one of these changes without the client knowing, which leads to a loss of synchronization between the client and the server. This loss of synchronization can have several undesirable effects.

If the server's hardware address changes or the MPLS label is repurposed by the server for a different purpose, then packets may be sent to the wrong destination. The consequences can range from suboptimally routed packets to dropped packets to packets being delivered to the wrong customer, which may be a security breach. This last may be the most troublesome consequence of loss of synchronization.

If a destination transitions from entropy label capable to entropy label incapable (an unlikely event) without the client knowing, then packets encapsulated with entropy labels will be dropped. A transition in the other direction is benign.

If the metric changes without the client knowing, packets may be suboptimally routed. This may be the most benign consequence of loss of synchronization.

Standard ARP has similar issues. These are dealt with in two ways: a) ARP bindings are time-bound; and b) an ARP server, recognizing that a change has occurred, can send unsolicited ARP messages ([RFC2002]). Both these techniques are used in L-ARP: the validity of the MPLS label obtained using L-ARP is time-bound; an L-ARP client should periodically resend L-ARP requests to obtain the latest information, and time out entries in its ARP cache if such an update

is not forthcoming. Furthermore, an L-ARP server may update an advertised label binding by sending an unsolicited L-ARP message if any of the parameters mentioned above change.

6. Applicability

L-ARP can be used between a host and its Top-of-Rack switch in a Data Center. L-ARP can also be used between a DSLAM and its aggregation switch going to the B-RAS. More generally, L-ARP can be used between an "Access Node" (AN) (e.g., the DSLAM) and its first hop MPLS-enabled device in the context of Seamless MPLS [I-D.ietf-mpls-seamless-mpls]. The first-hop device is part of the MPLS Fabric, as is the Service Node (SN) (e.g., the B-RAS). L-ARP helps create an MPLS tunnel from the AN to the SN, without requiring that the AN be part of the MPLS Fabric. In all these cases, L-ARP can handle the presence of multiple connections between the access device and its first hop devices.

ARP is not a routing protocol. The use of L-ARP should be limited to cases where an L-ARP client has Ethernet connectivity to its L-ARP servers.

7. Backward Compatibility

Since L-ARP uses a new hardware type, it is backward compatible with "regular" ARP. ARP servers and clients MUST be able to send out, receive and process ARP messages based on hardware type. They MAY choose to ignore requests and replies of some hardware types; they MAY choose to log errors if they encounter hardware types they do not recognize; however, they MUST handle all hardware types gracefully. For hardware types that they do understand, ARP servers and clients MUST handle operation codes gracefully, processing those they understand, and ignoring (and possibly logging) others.

8. For Future Study

The L-ARP specification is quite simple, and the goal is to keep it that way. However, inevitably, there will be questions and features that will be requested. Some of these are:

1. Keeping L-ARP clients and servers in sync. In particular, dealing with:

   A. client and/or server control plane restart

   B. lost packets

   C. timeouts

2. Withdrawing a response.

3. Dealing with scale.

4. If there are many servers, which one to pick?

5. How can a client make best use of underlying ECMP paths?

6. and probably many more.

In all of these, it is important to realize that, whenever possible, a solution that places most of the burden on the server rather than on the client is preferable.

These questions (and others that come up during discussions) will be dealt with in future versions of this draft.

9. L-ARP Message Format

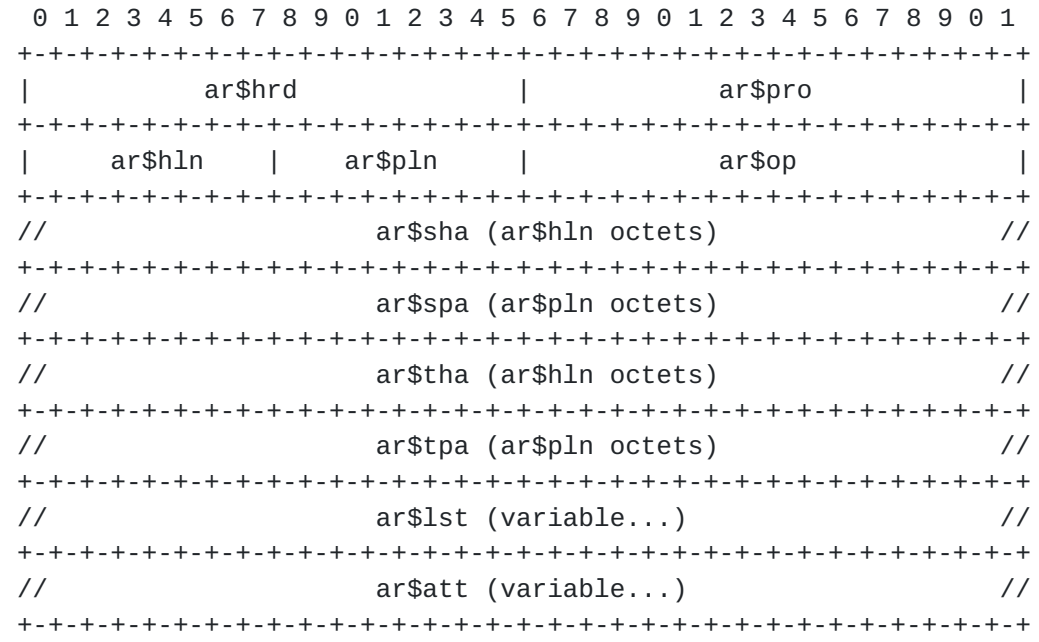


Figure 2: L-ARP Packet Format

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 Length: 6.

ar$pln Protocol Address Length: for an IPv4 address, the value is 4; 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 message, Source Hardware Address is the 6 octet sender's MAC address.

ar$spa Source Protocol Address: In an L-ARP message, this field carries the sender's IP address.

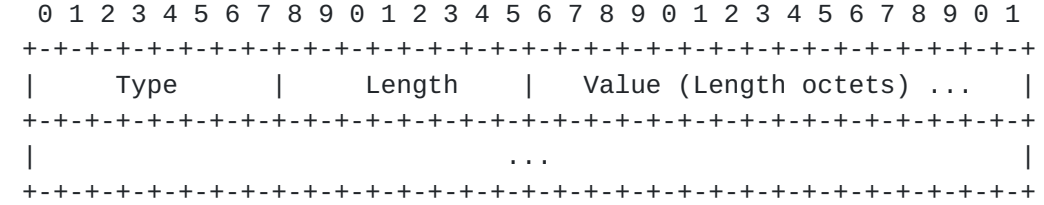
ar$tha Target Hardware Address: In an L-ARP query message, Target Hardware Address is the all-ones Broadcast MAC address; in an L-ARP reply message, it is the client's MAC address.

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

ar$lst Label Stack: In an L-ARP request, this field is empty. In an L-ARP reply, this field carries the MPLS label stack as an ARP TLV in the format below.

ar$att Attributes: In an L-ARP request, this field is empty. In an L-ARP reply, this field carries attributes for the MPLS label stack as an ARP TLV in the format below.

This document introduces the notion of ARP TLVs. These take the form as in Figure 3. Figure 4 describes the format of Label Stack TLV carried in L-ARP. Figure 5 describes the format of Attributes TLV carried in L-ARP.



Type is the type of the TLV; Length is the length of the value field in octets; Value is the value field.

Figure 3: ARP TLVs

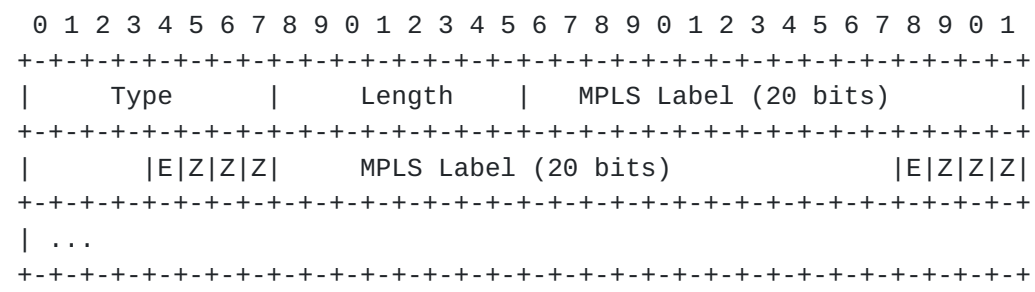


Figure 4: MPLS Label Stack Format

Label Stack: Type = TLV-LST; Length = n*3 octets, where n is the number of labels. The Value field contains the MPLS label stack for the client to use to get to the target. Each label is 3 octets. This field is valid only in an L-ARP reply message.

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.

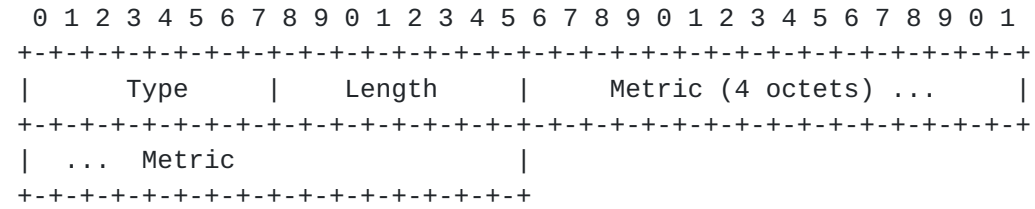


Figure 5: Attribute TLV

Attributes TLV: Type = TLV-ATT; Length = 4 octets. The Value field contains the metric (typically, IGP distance) from the responder to the destination (device with the requested IP address). This field is valid only in an L-ARP reply message.

If other parameters are deemed useful in the ATT TLV, they will be added as needed.

10. 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.

11. IANA Considerations

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

IANA is also requested to create a new registry ARP-TLV ("tlv"). This is a registry of one octet numbers. Allocation policies: 0 is not to be allocated; the range 1-127 is Standards Action; the values 128-251 are FCFS; and the values 252-255 are Experimental.

Finally, IANA is requested to allocate two values in the ARP-TLV registry, one for TLV-LST and another for TLV-ATT.

12. 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.

13. References

13.1. Normative References

[RFC0826] Plummer, D., "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, <http://www.rfc-editor.org/info/rfc826>.

[RFC2002] Perkins, C., Ed., "IP Mobility Support", RFC 2002, DOI 10.17487/RFC2002, October 1996, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6790>.

13.2. Informative References

[I-D.ietf-mpls-seamless-mpls] Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: kireeti.kompella@gmail.com

Balaji Rajagopalan
Juniper Networks, Inc.
Prestige Electra, Exora Business Park
Marathahalli - Sarjapur Outer Ring Road
Bangalore 560103
India

Email: balajir@juniper.net

George Swallow
Cisco Systems
1414 Massachusetts Ave
Boxborough, MA 01719
US

Email: swallow@cisco.com