# REVISION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rev.** | **Date** | **Agile Revision** | **Update By** | **Revision Description** | **Status** |
| 0.1 | 11/11/2016 |  | Sindhu Mohandas | Initial Draft |  |

# Overview

This purpose of this document is to present the different functions the HAP has to implement for mDNS solution in R3.0 for the different use cases described in this document. For details of mDNS and what is being achieved with this solution the original SRD must be referred. This document is mainly focused on the requirement for the HAN HAPs which will be operating in different modes based on the use case it is being deployed in.

# mDNS Use Cases

The assumption is that MLE deployment has a OV.

1. MLE Deployment with Aruba wireless controller(running mDNSResponder) and ALE switches and HAPs
   1. Here all the ALE switches and HAPs should configured with mDNS enable, mDNS mode as ‘tunnel’ and with tunnel-type as ‘L2GRE-aruba’,
   2. All the edge switches/HAPs needs to configured with the L2GRE tunnel to the Aruba wireless controller.
   3. Alll mDNS traffic entering the switch/HAP should be encapsulated with the L2GRE header and sent out on the interface through which the remote endpoint of the tunnel is reachable. The mDNS traffic sent through the tunnel must be 802.1Q tagged with the vlan associated with the mDNS packet.
   4. When the mDNS traffic comes back from the controller on the L2GRE tunnel
      1. if the packet is unicast it is forwarded based on the destination.
      2. If the packet is multicast then it is forwarded in the vlan-id tag in the packet.
   5. Statistics of Tx mDNS and Rx mDNS on the L2GRE has to be maintained.
2. MLE Deployment with Aruba wireless controller and only HAPs
   1. The requirements for the HAP is the same as in use case 1
3. MLE Deployment with ALE router (running mDNSResponder) and edge switches and HAPs and OV
   1. An ALE Switch/Router Standalone or Virtual Chassis is configured by OV as the mDNS responder
   2. All the ALE edge switches and HAPs should be configured with mDNS enable, mDNS mode as ‘tunnel’ and with tunnel-type as ‘L2GRE-aruba’,
   3. All the edge switches/HAPs needs to configured with the L2GRE tunnel remote tunnel endpoint IP address of the centralized ALE switch running the mDNSResponder.
   4. All mDNS traffic entering the switch/HAP should be encapsulated with the L2GRE header and sent out on the interface through which the remote end-point is reachable. The mDNS traffic sent through the tunnel must be 802.1Q tagged with the vlan associated with the mDNS packet.
   5. Since the centralized mDNS Responder is not aware of the client information it is required for this information to be sent to the mDNSResponder as part of the L2GRE packet sent in step ‘e’
      1. The suggestion is to append the mDNS Packet with a ‘TXT’ record with Access Role Profile name (limit to 128 characters) , Location (limit to 128 characters), Vlan and mac-address. This has to be proto-typed to make sure avahi is not broken when the packet is modified by the switch or HAP.
   6. When the mDNS traffic comes back from the controller on the L2GRE tunnel ,
      1. if the packet is unicast it is forwarded based on the destination without any modification except removing the tag on the switch port if the port is untagged for the vlan.
      2. If the packet is multicast then it is forwarded in the vlan-id tag in the packet. Again the packet is not modified except removing the tag on the switch port if the port is untagged for the vlan.
      3. Statistics of Tx mDNS and Rx mDNS on the L2GRE has to be maintained.
4. MLE Deployment with only HAPs
   1. The mDNSResponder should run on one or two of the highend HAPs. This is configured by OV.
   2. All the HAPs must be configured with L2GRE with the remote tunnel end-point of one or both the mDNSResponders , with one identified as the Primary and the other the Secondary.
      1. For these HAPs, running in edge mode, which includes the HAPs running the mDNSResponder, all the points from point 3d to 3f are applicable.
   3. The HAPs running the mDNSResponder will also have clients/servers which have mDNS. These should also use the L2GRE tunnel to send and receive mDNS packets same as the other HAPs in the network.
   4. The HAPs running the mDNSResponder will have these additional functions
      1. Learn , maintain, query and age out the service cache
      2. Enforce the service rules check with the information obtained about the client/server from the mDNS packet
      3. Maintain statistics of services discovered, services requested, service requests accepted and service requests rejected
      4. Primary and Secondary mDNSResponders running on the HAPs do not sync the service cache. When the Primary mDNSResponders is unreachable the secondary mDNSResponder shall receive all the mDNS packets and a new cache is learned from scratch.

# L2GRE

What is an L2GRE tunnel?

L2 GRE is a standards based technology used to provide seamless L2 connectivity through a virtual L2 overlay network that is superimposed on the physical L3 network. The virtual L2 network is agnostic of the physical topology of the underlying L3 network. L2 GRE eliminates classic vlan limitations.

The key difference between L2 GRE and L3 GRE:

L2 GRE and L3 GRE are identified through the value of the protocol type field in the GRE header.



The key difference is that the GRE Header protocol type = 0x800/0x8600 for L3 GRE and 0x6558 for standard L2GRE. For Aruba implementation, the requirement is to set the protocol type = 0x0 in the GRE Header. The GRE header is followed by IP header + IP payload for L3GRE whereas it is followed by L2 payload ( L2 Ethernet header + IP header + IP payload).

The detailed L2 GRE packet is as shown below and the contents of the some of the fields is as shown below:

The L2GRE for the Aruba mDNS solution will use GRE Header protocol type = 0x0 , Virtual Subnet ID(VSID) = 0x0.

The L2GRE for the ALE mDNS solution will use GRE Header protocol type = 0x6558, Virtual Subnet ID (VSID) = vlan id.

The L2GRE for the Guest Tunneling solution will use GRE Header protocol type = 0x6558, Virtual Subnet ID (VSID) = reserved service-id

**Outer Ethernet Header:**

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Outer) Destination MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|(Outer)Destination MAC Address | (Outer)Source MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Outer) Source MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|Optional Ethertype=C-Tag 802.1Q| Outer VLAN Tag Information |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Ethertype 0x0800 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Outer IPv4 Header:**

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|Version| IHL |Type of Service| Total Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Identification |Flags| Fragment Offset |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Time to Live | Protocol 0x2F | Header Checksum |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Outer) Source Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Outer) Destination Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**GRE Header:**

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|0| |1|0| Reserved0 | Ver | Protocol Type 0x6558 or 0 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Virtual Subnet ID (VSID) | FlowID |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Inner Ethernet Header**

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Inner) Destination MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|(Inner)Destination MAC Address | (Inner)Source MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| (Inner) Source MAC Address |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Ethertype 0x0800 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

**Inner IPv4 Header:**

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|Version| IHL |Type of Service| Total Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Identification |Flags| Fragment Offset |

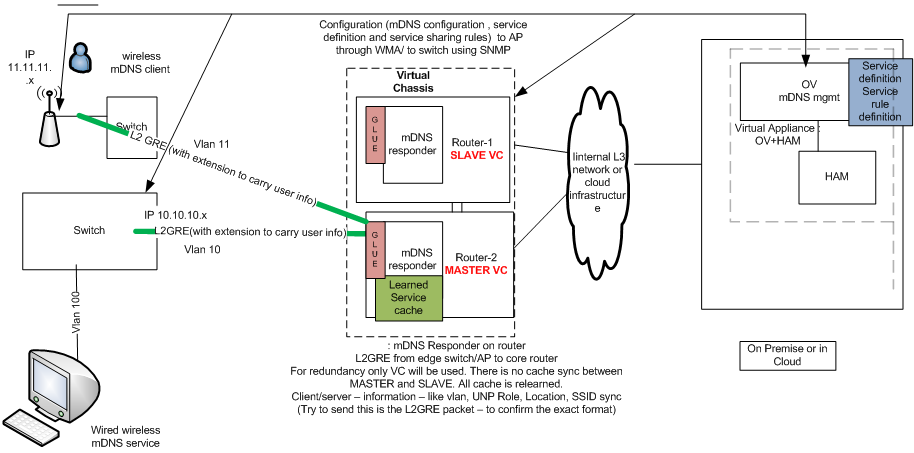
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Time to Live | Protocol | Header Checksum |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

# mDNS Solution

The centralized model running on one designated ALE switch /router in the network. This model is for a network that has both ALE wired switches and HAPs. The following diagram illustrates this model.



The centralized model running on a designated high-end HAP and a backup HAP in a pure HAN wireless network.



# HAP Requirements

The HAP has to be support the following modes of operation for mDNS solution

1. MDNS mode is ‘tunnel’ and Tunnel-type is ‘L2GRE-aruba’ for mDNS solution with mDNSResponder function on Aruba controller.
2. MDNS mode is ‘tunnel’ and Tunnel-type is ‘L2GRE-ale’ for mDNS solution with mDNSResponder function on ALE switch.
3. MDNS mode is ‘tunnel’ and Tunnel-type is ‘L2GRE-ale’ for mDNS solution with mDNSResponder function on a high-end HAP.
4. MDNS mode is ‘hybrid i.e tunnel + proxy’ on the HAP acting as an mDNSResponder and the source of the mDNS server/client packets from server/client coming through the HAP .

**Requirements of Tunnel-type “L2GRE-aruba” :**

1. L2 GRE Tunnel is setup to the far-end ip address of the L2GRE tunnel that is configured
2. The GRE header of the L2GRE packet is set with protocol type=0x0
3. All mDNS packets entering the SSID is encapsulated into a L2GRE tunnel
4. All mDNS packets coming in on the L2GRE tunnel is de-capsulated and has to be flooded or unicast based on the inner packet type is multicast or unicast
5. HAP in this mode should maintains statistics of Tx mDNSPackets and Rx mDNSPAckets

**Requirements of Tunnel-type ‘L2GRE-ale’:**

1. An L2GRE tunnel should be set up to the mDNSResponders.
   1. If the mDNSResponder is hosted on ALE switch the redundancy is achieved by hosting it on a Virtual chassis behind just one IP address. HAPs will be configured with one remote tunnel end-point.
   2. In a pure wireless use case, mDNSResponder is hosted on a high-end HAP. For redundancy the mDNSResponder must run on two HAPs with each having their own IP address. Hence the HAPs have to be configured with a Primary remote tunnel-endpoint and Secondary remote tunnel end-point.
2. The HAP should always send all the mDNS traffic to the Primary through the L2GRE tunnel
3. While sending the mDNS packet, the packet has to be appended with “access role profile, mac address, vlan and location” associated with the mac-address of the mDNS packet, then redo the checksum in the GRE header.
4. The mDNS packets coming back in the L2GRE tunnel is decapsulated and flooded or unicast based on the packet type.
5. HAP in this mode should maintain statistics of Tx mDNSPackets and Rx mDNSPAckets

**Requirements of mDNSResponder function on HAPs:**

1. The mDNSResponder running on a HAP can be enabled by setting the mode to hybrid (proxy+tunnel)
2. The mDNSResponder running on the HAP does not know if it is a Primary or secondary. It is the HAPs sending traffic to the mDNSResponder that is configured with Primary and Secondary tunnel end-point IP address. In the case of MDNSResponder on switch (there is no separate Primary/Secondary – it is on the same Virtual Chassis behind one IP address). In the case of pure wireless use case the mDNSResponder runs on two independent HAPs)
3. The mDNSResponder doesn’t sync the learned cache with the Secondary. They operate independently. When the HAP connecting to the mDNSResponder notices connectivity issues then it will send the traffic through the Secondary L2GRE. The HAP with the Secondary mDNSResponder will learn the cache when it starts receiving traffic. (Should there be heart beat implemented between Primary/Secondary, Should there is be automatic failover, Should there be cache synced???? The initial version will not support this)
4. When a Primary and Secondary is configured, then IP reachability has to be periodically tested from the HAPs in order to switchover after timeout ( The heart beat is ping every 3 sec and after 3 timeouts consider the Primary down. Send a syslog notification when the reachability to Primary or Secondary mDNSResponder is lost – TBC)
5. HAP running the mDNSResponder should maintain service cache, stats of services requested and stats of service requests accepted /filtered
6. HAP running the mDNSResponder should maintain statistics of Tx mDNSPackets, Rx mDNSPackets per tunnel terminating at the HAP running mDNSResponder

# OV Requirements for Configuring HAPs

1. mDNS Application should allow configuration of mDNSRelay {enable|disable}
2. mDNS Application should allow configuration of the MDNS mode – {tunnel, proxy, gateway, hybrid (proxy+tunnel)}.
3. mDNS Application should allow configuration of the the tunnel type when MDNS mode - ‘tunnel or ‘hbyrid’ is enabled in the previous step
4. mDNS Application should allow configuration of the L2GRE tunnel
   1. in tunnel-type L2GRE-aruba – source and destination ip address of L2GRE tunnel should be configured
   2. in tunnel-type L2GRE-ale (mDNSResponder running on a ALE switch in the network) – source and destination ip address of L2GRE tunnel should be configured
   3. in tunnel-type L2GRE-ale (pure wireless) –
      1. primary L2GRE - source and destination ip address of the L2GRE tunnel
      2. secondary L2GRE – source and destination ip address of the L2GRE tunnel
5. mDNS Application must allow the following configuration on the HAP Running mDNSResponder
   1. configure all the remote L2GRE endpoints or have a mode to accept any
   2. configure the service id
   3. configure the service sharing rule
   4. as an end-user (not all the other configuration is as OV admin) – register a service id and service sharing rule
6. mDNSApplication Show status
   1. On normal HAP
      1. Show the configuration
      2. Show the L2GRE tunnel being used
      3. Should the TX/RX mDNS statistics on the L2GRE
      4. Show the status of the L2GRE tunnel ( number of time connection was lost)
   2. on the HAP running mDNS Responder
      1. Show the configuration
      2. show the mDNS Tx/Rx packets from each tunnel end-point
      3. show the learned cache
      4. show the stats of responses to services
      5. show the stats of filtered service requests