# mDNS/SSDP Relay

The mDNS/SSDP Relay feature can be broken into the management plane and data/control plane.

In MLB deployments with OV, mDNS/Relay feature requires an OV and optionally HAM to get full feature.

Management for the mDNS/SSDP relay feature for MLB with service sharing rules configuration and device/service registration will be done through OV.

Configuration through the element management interface like CLI and WebView will only allow enabling of mDNS/SSDP feature and defining the vlan based service sharing for gateway model. OV is a requirement for MLB use case for supporting advanced service sharing rules.

The data/control plane could be distributed or centralized. The following sections captures the differences between the two models and also discusses the advantages of the centralized model over the distributed model.

# mDNS/SSDP Relay Management /Data Plane differences between the Centralized and Distributed model

The following table captures the differences between the two models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Configuration** | **Distributed** | | **Centralized** | |
|  | mDNS enable|disable | OV admininstrator | Configuration through OV  Configuration to be applied on edge switches through SNMP  Configuration to be applied on APs through the WMA | OV admininstrator | Configuration through OV  Configuration to be applied on   * edge switches through SNMP * APs through the WMA * mDNS component on HAM through API interface |
|  | Tunnel between edge switches and APs to the mDNSResponder running in HAM  Local and far-end ip address of the tunnel needs to be configured | OV administrator | NA | OV admininstrator | Configuration through OV  Configuration to be applied on   * edge switches through SNMP * APs through the WMA * mDNS component on HAM through API interface |
|  | Configuration of the mode of operation of mDNSResponder   * discover or enforcement + discover | OV Administrator | To be configured on the edge switches through SNMP  To be configured on the APs through the WMA | OV admininstrator | Configuration through OV  Configuration to be applied on   * mDNS component on HAM through API interface |
|  | Service definition i.e. the list of services and their service-ids that will be allowed to be shared on the network i.e airplay, airprint, dlna-media,etc | OV Administrator | Configuration through OV  Configuration to be applied on   * edge switches through SNMP * APs through the WMA | OV Administrator | Configuration through OV  Configuration to be applied on   * edge switches through SNMP * APs through the WMA * mDNS component on HAM through API interface |
|  | Network service sharing rules | OV Administrator | Configuration through OV  Configuration to be applied on   * edge switches through SNMP * APs through the WMA | OV Administrator | Configuration through OV  Configuration to be applied on   * mDNS component on HAM through API interface |
|  | End-user service sharing rules | End-user | Configuration through HAM  Configuration to be applied on   * edge switches through SNMP * APs through the WMA | End-user | Configuration through HAM  Configuration to be applied on   * mDNS component on HAM through API interface |
|  | **Data / Control Plane** | **Distributed** | | **Centralized** | |
|  | Components of the mDNS Module: mDNSResponder module + mDNSGlue module. | Both the components run on the edge switch and the AP | | Both the components run in the HAM | |
|  | When mDNS is disabled | At the edge switches/APs – mDNS is flooded normally in the L2 domain | | At the edge switches/APs – mDNS is flooded normally in the L2 domain | |
|  | When mDNS is enabled | The default mode is discover only.  The mDNSResponder is not activated on the edge switch/AP in this mode since there is no response to service requests.  mDNS traffic is flooded normally only in L2 mode  The switch and AP just logs the discovered services on the network in a circular buffer. OV can pull this information on demand for display and to provide as a list to be selected for service definition for the network | | The default mode is discover only.  On edge switch and AP:   * mDNS module will create a tunnel to the far-end mDNSResponder running in HAM * shall forward all mDNS traffic coming on their ports to the tunnel set up to the mDNS module on HAM   On the HAM:   * mDNS module will be in discover mode or enforcement (discover+enforcement) mode * tunnels shall be configured to all edge switches/AP | |
|  | What are the functions of the mDNSGlue module/ | * receive the mDNS packets from the ports onto CPU * keep a log of the discovered services when mDNS discovery is enabled * keep a log of the discovered services when mDNS enforcement is enabled and forwarding the packets to the mDNSResponder interface (vlan) * Intercept the packet that the mDNSResponder sends back on an interface (vlan), apply the service rules checks and forward only those packets that are allowed by the rules * Maintain stats of the service requests seen, service requests accepted to be forwarded and service requests rejected * Notify OV of the cached service   + Edge switch will use SNMP TRAP   + AP will notify WMA which will notify OV * Cache sync from OV   + Edge switches will receive SNMP set   + AP will receive WMA set messages | | * receive the mDNS packets from the L2GRE tunnel * to keep a log of the discovered services when mDNS discover is enabled * keep a log of the discovered services when mDNS enforcement is enabled and forwarding the packets to the mDNSResponder interfaces * Intercept the packet that the mDNSResponder sends back on an interface , apply the service rules checks and forward only those packets that are allowed by the rules * Maintain stats of the service requests seen, service requests accepted to be forwarded and service requests rejected | |
|  | What are the functions of the mDNSResponder? | * cache the mDNS service published and respond to service requests. Maintain the service cache | | * cache the mDNS service seen and respond to clients requesting the service. Maintain the service cache | |
|  | How is the mDNS packets reaching mDNS module? | It is copied to the CPU and then to the mDNS module (Glue module + Responder) within each edge switch/AP | | It is forwarded to the mDNS module running on HAM through a L2 GRE tunnel from the edge switch /AP | |
|  | Impact of OV/HAM goes down | mDNS and service configuration is impacted  The mDNS service cache sync is affected  The mDNS client request for services that have expired from remote cache will be affected | | mDNS and service configuration is impacted  mDNS data plane is redundant because HAM is redundant hence data forwarding should continue | |

# Distributed vs Centralized

Following drawing illustrates the distributed data plane model:



Following drawing illustrates the centralized data plane model:



|  |  |  |
| --- | --- | --- |
|  | **Cons of Distributed model** | **Pros of Centralized model** |
|  | Memory footprint required for the mDNSResponder software like Avahi ( atleast 8MB for mDNS and more for SSDP Proxy), the service cache (few hundred on the network) and service rules (few thousand clients) on edge switches/APs  Since the switches/APs are embedded systems with memory/flash and CPU constraints, running the mDNSResponder and storing the entire cache and client service request information will have significant system impact | This will be centralized on the VM of the HAM. Since this is not an embedded system with limited cpu/memory and flash, the resource requirements will be more scalable based on the network requirements |
|  | The open source mDNSResponder implementations are designed to work in the centralized model hence changes have to be made to work in a distributed model | The open source mDNSResponder implementations are designed to work in the centralized model. It is best fit for the centralized model |
|  | A new protocol needs to be developed for service cache sync across the network if SNMP TRAP/SET is not acceptable between switch and OV.  HAP will be using WMA interface to notify the service cache discovery and service cache sync | Since the mDNSResponder sees all the traffic from all the edge switches/APs through the L2GRE tunnel, there is no need to sync the service cache at the edge network elements.  This is a big |
|  | Redundancy is an issue because OV is not redundant. The service cache sync service will not be available if OV is down | Redundancy will be handled by the HAM Redundancy. mDNS module will run as part of HAM. A redundancy protocol should be created to sync the master and slave HAM to keep service cache and configurations in sync. |
|  |  | Solution works for use cases:   1. Solution with switches/HAPs and OV/HAM 2. Solution with only HAPs and OV/HAM |
|  | **Pros of Distributed model** | **Cons of Centralized model** |
|  | Solution works for use cases:   1. Solution with switches/HAPs and OV and with or without HAM 2. Solution with only HAPs and OV and without or without HAM | Solution depends on HAM because mDNSResponder/Glue modules run in HAM |
|  | Solution is more conducive for moving to cloud because data plane is still localized within the network | Solution may have issues since the mDNS data plane is in the cloud |

Based on the above comparison we are now considering the Centralized model as a solution for mDNSRelay.

# OV Requirements

1. Configuration to edge switches and HAPs. Switches are configured using SNMP and HAPs are configured through WMA
   1. mdns enable| disable mode {tunnel-aruba | tunnel-ale | gateway}
   2. Configure the tunnel endpoints in tunnel mode
2. Configuration to the HAM as OV Administrator
   1. mdns {enable| disable} {discover | enforce}
   2. Configure the tunnel endpoints of all the tunnels to all the edge switches and APs. This should be of all the edge switches /APs where mdns was enabled in the points 1 and 2 above
   3. Configure the services and their IDS that are allowed on the network that will be handled by the mDNSResponder
   4. Configuration of the devices and the services on them
   5. Configuration of the services sharing rules of the services configured
3. Configuration to HAM as a end-user administrator
   1. Configuration of the end user devices and the services on them
   2. Configuration of the end user services sharing rules of the services configured

# Switch Requirements

1. When mdns enable
   1. Just enable the feature globally
   2. Continue normal mDNS forwarding at L2. No special action
2. When tunnel configured
   1. Both tunnel-ale or tunnel-aruba should both create a L2 GRE tunnel
   2. Create a tunnel to far-end ip address
   3. If mdns is enabled
      1. then copy all the mDNS packets to CPU and send them through tunnel to central mDNS controller
      2. disable normal L2 forwarding of mDNS packets coming from the ports
   4. if mdns is disabled
      1. stop forwarding all the mDNS packets through the tunnel.
      2. enable the normal L2 forwarding only

# HAP Requirements

1. When mdns enable
   1. Just enable the feature globally
   2. Continue normal mDNS forwarding at L2. No special action
2. When tunnel configured
   1. Both tunnel-ale or tunnel-aruba should both create a L2 GRE tunnel to the respective tunnel end-point
   2. Create a tunnel to far-end ip address
   3. If mdns is enabled
      1. then copy all the mDNS packets to CPU and send them through tunnel to central mDNS controller
      2. No more forwarding at L2 of mDNS packets coming from the wireless
   4. if mdns is disabled
      1. stop forwarding all the mDNS packets through the tunnel and enable the normal L2 forwarding only

# HAM Requirements

1. The mDNS module will run as part of the HAM
2. It has two components
   1. mDNS Glue module
   2. mDNS Responder
3. mDNS Glue modules receives the configuration and provides configuration to the mDNSResponder
4. mDNS Glue will be responsible for getting traffic from the L2Gre tunnel
   1. and logging if only discover mode is enabled
   2. logging and forwarding to mDNSResponder if discover and enforcement is enabled
   3. performing service rules check before forwarding the mDNS packets out
   4. Maintain stats of the services discovered, services requested, services granted access and services denied access.

# WMA Requirements

1. WMA will continue to be the interface between OV and HAP for configuration push to HAPs or status poll from the HAP

# Appendix

The Service Cache can contain the following information:

* Service Type ( 1 byte – a number)
* Service ID ( recommended size of TXT record by Bonjour is 512 bytes)
* IP address of end device publishing the service ( 6 bytes)
* Port number used by the publishing service ( 1byte)
* End Device mac address ( 6 bytes)
* UNP Profile/Role associated with the end device mac address (??)
* Vlan ( 2 bytes)
* Location information ( ??)
* MAC address of the Switch/AP that saw this device (6 bytes)
* Mgmt vlan IP address of the Switch/AP that saw the device (4 bytes)
* TTL ( time to expire the entries)