Software Design Specification

Z/IP DNS-SD Service Discovery support

Document No.:	SDS11756
Version:	
Description:	Z/IP Router support for discovery of Z-Wave and Z/IP resources via mDNS and DNS-SD.
Written By:	ABR;JRM;AES;JFR;BBR
Date:	
Reviewed By:	ABR;JRM;JFR;AES;NTJ;JBU
Restrictions:	Public

Approved by:

Date	CET	Initials	Name	Justification
2018-03-06	09:25:19	NTJ	Niels Thybo Johansen	

REVISION RECORD

Doc. Ver.	Date	By	Pages affected	Brief description of changes
3	20160823	ABR	All	First revision for public release
4	20180306	BBR	All	Added Silicon Labs template

Table of Contents

1 ABBREVIATIONS

Abbreviation	Explanation

2 INTRODUCTION

This document presents a scalable and flexible platform for standards-based discovery of Z-Wave resources in an IP based infrastructure. A Z-Wave resource may be a self-contained Z-Wave node or a Z-Wave multichannel End Point.

The Z/IP Resource Directory (RD) described in [1] is based on the concepts defined for the CoRE Resource Directory (RD) [7] and the mapping from CoRE RD to mDNS [3] described in [6]. SDS11633 [1] specifies the information that is needed to provide the service discovery support described in this document.

mDNS is part of a complete vision for zero-configuration networking [5]. mDNS discovery allows IP based management systems with no knowledge of the network topology to locate a Z-Wave resource in a given network infrastructure; be that the home or an entire hotel.
While often mentioned as a discovery technology, mDNS [3] is really just a multicast-enabling transmission extension to DNS-SD [4].

mDNS is not limited to IPv6. Actually most mDNS deployments today run over IPv4. The Z/IP Gateway architecture allows for connectivity to, as well as discovery of, IPv4 and IPv6 resources.

2.1 Terms used in this document

The guidelines outlined in RFC 2119 [2] apply.
Essentially, the key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

The reader is strongly advised to read “Zero Configuration Networking: The Definitive Guide” [5] to get a deeper understanding of DNS-SD and mDNS.

3 CONCEPTUAL REQUIREMENTS FOR Z/IP DISCOVERY

This section defines the conceptual requirements that form the basis for the specific implementation requirements and guidelines provided in chapter 4.

3.1 Complete inventory discovery

Connecting a GUI management tool to the network for the first time, the management tool may need to determine all available resources in order to assist an installer in placing all resources in a floor plan of the home.

It MUST be possible to request ALL available Z-Wave resources in a subnet via a gateway connected to a backbone network.

3.1.1 Saving battery lifetime and network bandwidth

Most discovery information is static and there is no need to wake up sleeping devices or using network bandwidth on re-querying information that is still valid in the database maintained by the Z/IP RD [1]. Likewise, waiting for 5 minutes for response from a wakeup device would not be efficient.

The mDNS responder MUST return information for any Z-Wave resource. The Z/IP RD is REQUIRED to make this information available to the mDNS responder.

3.1.2 Providing scalability support

In most homes there is a single Z/IP Gateway with an associated Z/IP RD. In larger deployments like hotels or companies, there may be hundreds of Z/IP Gateways.

It MUST be possible to issue a specific query for Z/IP Gateways on the backbone IP network. Such a request MAY be implemented as a request for support of the Network Management command class. A requester SHOULD then use unicast transmission to one Z/IP Gateway at a time to determine resources at this particular Z/IP RD.

In order to further lower the network load, a requester SHOULD employ mDNS Known-Answer Suppression [3] in repeated queries for Z/IP resources so that mDNS announcements which were lost at the first event may have an improved chance of being delivered the next time.

3.2 Service specific inventory (mDNS sub-service support)

When deploying a surveillance application one may need to determine all available sensor and meter resources, e.g. power meters, temperature sensors, etc.

It MUST be possible to request ALL available Z-Wave resources providing a given Z-Wave Command Class in a subnet. Certain Z-Wave Command Classes require that additional information is conveyed for determining the actual properties of a Z-Wave resource.

3.3 Failing devices

For a service technician to do a quick check on the status of an installed system, the person needs to retrieve a list of failing devices from the Z/IP RD.

The Z/IP RD MUST return a record for every failing resource, not only the physical entity (Z-Wave End Point ID = 0). This allows a management tool to list all affected configurations in the installation.

3.4 Z-Wave Command Class awareness

mDNS queries for Z-Wave services MUST use the Z-Wave command class identifiers defined in the Z-Wave Command Class specifications.

An mDNS responder MUST return a TXT record that contains all relevant Command Classes supported by each Z-Wave resource; including supplemental Command Class information. Refer to 3.4.1.
The mDNS responder SHOULD omit command classes that have no operational value from an application control point of view. All Z-Wave Protocol Command Classes MUST be omitted.
An mDNS responder SHOULD return a TXT record that contains the Icon Type of each resource. Refer to 3.4.2.

3.4.1 Extended NodeInformation format for advanced Z-Wave services

The classic Z-Wave Node Information format provides generic and specific Device Types followed by a list of supported Command Classes.

In order to support queries for specific sensor types, e.g. a temperature sensor, the Z/IP mDNS responder and a Z/IP mDNS client MUST support the extended Node Information Format.
This maps into a string in the “TXT info=” DNS record and sub-service identifiers spanning more than one byte.

Compared to the classic Z-Wave Node Information format, the extended Z/IP mDNS Node Information format extends the concept of extended Command Class codes to also cover the following Command Classes

- Meter Command Class
- Multilevel Sensor Command Class
- Alarm Sensor Command Class
- Notification Command Class

The Multilevel Sensor Command Class code MUST be followed by a sensor type and scale.
The Meter Command Class code MUST be followed by a meter type.
The Notification Command Class code MUST be followed by a notification type.
The Alarm Sensor Command Class code MUST be followed by an alarm sensor type.

The extension allows the Z/IP mDNS solution to meet the following requirements:

It MUST be possible to issue a single mDNS query for a specific multilevel sensor service.
It MUST be possible to issue a single mDNS query for a specific meter service.
It MUST be possible to issue a single mDNS query for a specific notification service.
It MUST be possible to issue a single mDNS query for a specific alarm sensor service.

3.4.2 Supporting graphical representation

mDNS service discovery is about discovering services provided by resources. The operation of a service may be handled by very different devices. Thus it is not relevant to look for a particular device type. Light dimming may be provided by plug-in modules as well as ceiling outlets.

Visualization is however important for a GUI management tool. The Icon Type allows a management tool to assign meaningful icons to resources presented in a graphical floor plan in an installer tool or in the list of a user app.

Multi-resource devices such as power strip may present default icon types for each individual End Point as well as for the Root Device.

How a management tool visualizes multi-resource devices is out of scope of this document.

4 IMPLEMENTATION REQUIREMENTS AND GUIDELINES

The Z/IP mDNS discovery mechanism described in this document is compliant with all requirements and recommendations found in [3] and [4] while it deviates from the CoRE principles described in [7] and [6] where not applicable to Z-Wave service mapping.

The Z/IP mDNS discovery mechanism only works within the same IP subnet.

4.1 Managing queries for Z-Wave resources

The first step of the mDNS query process is to identify the requested service. The most general query for Z-Wave resources is

```
_z-wave._udp.local.  PTR ?
```

The query calls for all mDNS-aware responders which implements one or more Z-Wave resources.

In many cases the requester is looking for a particular Z-Wave service; e.g. light dimming. In mDNS terms, this is a sub-service. The following query specifically calls for Z-Wave resources supporting COMMAND_CLASS_SWITCH_MULTILEVEL (0x26).

```
_26._sub._z-wave._udp.local.  PTR ?
```

The query calls for all mDNS-aware responders which implements Z-Wave resources that support COMMAND_CLASS_SWITCH_MULTILEVEL. The identified devices MAY also be able to control other devices via this command class.

If instead the query is intended to find only resources which can control other resources via COMMAND_CLASS_SWITCH_MULTILEVEL the request is structured as follows:

```
_ef26._sub._z-wave._udp.local.      PTR ?
```

By prepending the COMMAND_CLASS_MARK (0xEF) to COMMAND_CLASS_SWITCH_MULTILEVEL (0x26) the query calls for all mDNS-aware responders which implements Z-Wave resources that can control COMMAND_CLASS_SWITCH_MULTILEVEL resources.

As a curiosity, a query may be constructed to locate resources which can be controlled via COMMAND_CLASS_SWITCH_MULTILEVEL but cannot control other resources:

```
_26ef._sub._z-wave._udp.local.      PTR ?
```

Appending the COMMAND_CLASS_MARK (0xEF) to COMMAND_CLASS_SWITCH_MULTILEVEL (0x26) indicates (since no more codes follow) that there must be no control capabilities for the actual command class. In most cases, however, when looking for a resource that can be dimmed, the requester really does not care if the resource that can be dimmed is also able to control dimming in other resources. One such example is a wall-mounted dimmer with local load controls.

If searching for more than one Z-Wave Command Class, several mDNS queries must be constructed.

4.2 mDNS PTR Query format

The mDNS query for <u>all</u> Z-Wave resources MUST be formed as follows:

```
_z-wave._udp.local PTR ?
```

The mDNS query for a sub-set of Z-Wave resources MUST be formed as follows:

```
_<Z-Wave Command Class*)>._sub._z-wave._udp.local PTR ?
```

*) covers any of

- Z-Wave Command Class (1 byte)
- Extended Z-Wave Command Class (2 bytes)
- Multilevel Sensor Command Class with appended Sensor Type and scale (3 bytes)
- Meter Command Class with appended Meter Type (2 bytes)
- Notification Command Class with appended Notification Type (2 bytes)
- Alarm Sensor Command Class with appended Alarm Sensor Type (2 bytes)

The Command Class field MUST be expressed in lower case 2-digit ASCII hex strings.

A COMMAND_CLASS_MARK identifier MAY be appended or prepended to any of the abovementioned constructs. Refer to 4.1.

4.3 mDNS PTR Query Response format

The Z/IP mDNS responder reacting to a PTR query MUST return a response with the following format:

```
_z-wave._udp.local PTR <Z-Wave Resource Name>._z-wave._udp.local
```

Where <Z-Wave Resource Name> is a unique name in the entire network topology. An example is shown below:

```
_z-wave._udp.local PTR  Acme Dimmer Dx7 [c001babe1201]._z-wave._udp.local
```

Two cases apply for obtaining unique resource names in the Z/IP RD: User-assigned names and autoassigned names. Refer to section 4.3.1.

In both cases, the name MUST be human readable as it is intended to show up in browser lists that the user can choose from. All characters are allowed in the <Z-Wave Resource Name> portion as UTF-8 format MUST be used. Note that the '<' and '>' characters are not part of the actual name. They are only used to indicate that <Z-Wave Resource Name> is a label.

4.3.1 UTF-8 in Z-Wave resource names

A requesting Z/IP mDNS client MUST interpret multiple dots in the Z-Wave resource name as follows:

Text from the left until the first dot MUST be interpreted as the Name. The Name MUST NOT be blank.

All text from the left after the first dot until "._z-wave" MUST be interpreted as the Location.
The Location field MUST be considered blank if the "._z-wave" label follows immediately after the Name.

There is no limitation to the use of international characters in the naming defined manually by users:

```
_26._sub._z-wave._udp.local.  PTR
Bæverlampe.Hjørnebord.Hjemmebiograf.Førstesal.Sydfløj._z-wave._udp.local.
```

In the above example,
“Bæverlampe” is the name (text until the first dot).
”Hjørnebord.Hjemmebiograf.Førstesal.Sydfløj” is the location.

The mDNS documentation [3] specifies that an mDNS responder MUST ensure that a resource name is unique before announcing that resource.

Z/IP RD MUST request that the Z/IP mDNS responder probes a new resource name before it requests that the Z/IP mDNS responder announces the resource name and the mode of that resource.

The mDNS responder SHOULD NOT implement any parsing logic with regards to the Name and Location information.

4.3.2 Unique resource name – autogenerated

Resource names discovered through mDNS Query Responses may be presented to end users.
Therefore the Z/IP RD MUST auto-generate meaningful resource names in case the user did not specify name and location information during inclusion.
The resource name MUST be unique all over the network infrastructure. Future homes may have more than one IP subnet. Hotels and commercial buildings certainly already do.
The Z/IP RD MAY provide the following information for each resource:

1. HomeID
2. NodeID
3. End PointID
4. Manufacturer (text)
5. Product Name (text)
6. Device Type

The Z/IP RD MUST combine the information to a unique human readable string of the following form:

```
“<ManufacturerString><ProductString> [<HomeID><NodeID><End PointID>]”
```

OR alternatively the Z/IP RD MUST provide the following if manufacturer information is not available:

```
“<DeviceType> [<HomeID><NodeID><End PointID>]”
```

The following resource name was composed after this recipe:

```
“Acme Dimmer Dx7 [c001babe1201]”
```

4.3.3 Client handling of auto-generated names

The absence of dots in a newly discovered resource name SHOULD cause a GUI based Z/IP client to group resources in an “unsorted” group from which the user can easily identify resources that have not yet received meaningful name and location information.

4.3.4 Optimized PTR Query Response transmission

If possible, the Z/IP mDNS responder SHOULD accumulate all response entries in one response message; including SRV, TXT and AAAA records.

4.4 mDNS SRV & TXT Query format

Where the PTR record is used to map service queries to responses, SRV and TXT records are used to resolve the resource names conveyed in PTR responses.

A typical SRV & TXT Query looks like this:

```
Acme Dimmer Dx7 [c001babe1201]._z-wave._udp.local.    SRV    ?
Acme Dimmer Dx7 [c001babe1201]._z-wave._udp.local.    TXT    ?
```

The string provided in the SRV and TXT query is the string received in the PTR Query Response.

Responses to SRV and TXT queries SHOULD be returned in one DNS response message. For clarity reasons, however, this document describes the SRV and TXT responses in separate sections 4.5 and 4.6.

An AAAA record is also returned to bind the host name to the IPv6 address.
An A record MAY also be returned if a gateway provides IPv4 access to resources.

4.5 mDNS SRV Query Response format

A typical SRV Query Response looks like this:

```
Acme Dimmer Dx7 [c001babe1201]._z-wave._udp.local.    SRV  0 0 4123
zwc001babe12.local.
                               zwc001babe12.local.        AAAA   fd06:1e14:c501::1::12
```

The host name in the SRV response MUST be composed from one of the following formulas:

```
"zw<HomeID><NodeID>.local."

"zw<MAC_ADDRESS[5..0]>.local."
```

For Z-Wave nodes which have no built-in MAC address, the HomeID and NodeID values MUST be used to construct a unique host name. IP hosts having a native MAC address MUST use the complete MAC address as host name. In both cases, the text “zw” MUST be prepended.

The host name AAAA record MUST resolve to a ULA IPv6 address.
The host name AAAA record MAY resolve additional routable IPv4 or IPv6 addresses.

4.6 mDNS TXT Query Response format

A typical TXT Query Response looks like this:

```
Acme Dimmer Dx7 [c001babe1201]._z-wave._udp.site       TXT    ?
```

The TXT record allows for many different pieces of information per resource. Each key-value pair has a length limitation of 255 octets. Each key-value pair is individually described in the following sections.

Some information is gathered directly from actual nodes. This includes the supported command classes, number of End Points and the communication mode; e.g. “alwayslistening”

The Z/IP RD MUST construct a string by concatenating the following sub-strings. Each sub-string starts with one binary length-byte that indicates the length of the actual sub-string; not including the length byte. Bytes in TXT records MAY be any UTF-8 character.

4.6.1 txtvers=1

txtvers=1 is by convention a field added by most mDNS responders to ensure that another TXT record format may be introduced at some time in the future.
The first version MUST be 1.

```
<0x09>txtvers=1
```

4.6.2 info=

The info field carries an extended Node Information (extNIF) format.

```
<length>info=<GenDevType><SpecDeviceType><CommandClass><CommandClass>
```

Compared to the NIF, the extNIF format extends the Multilevel Sensor, Meter, Notification and Alarm Sensor Command Classes to explicitly list the datatype served by the actual resource; refer to section 3.4.1.

Thus the extNIF adds more extended command classes in addition to the classic extended command class range.

The Marker Command Class MUST be supported just as for classic NIF formats; refer to section 3.4.1.

An example of a full-featured extNIF structure:

```
<length>info=<GenDevType><SpecDeviceType>
             <ZW+ Marker CC>
             <CommandClass 1>
             <CommandClass 2>
             <MARK>
             <CommandClass 1>
             <CommandClass 2>
             <CommandClass 3>
```

4.6.3 epid=

The epid field identifies the End Point that implements the actual resource.

```
<0x06>epid=<End PointID>
```

End Point ID = 0 represents the physical entity. If there are no other End Point IDs announced for the actual host, this is the only resource.

The resource names assigned to individual resources may be completely different.
A management tool that wants to group all resources hosted by the same physical instance MUST identify the resources which refer to the same SRV hostname.

Having done this, the management system may draw a graphical border around the individual resources.

4.6.4 icon=

The Z/IP RD MUST provide the Icon Type [9], which allows a GUI to show meaningful icons for a Root Device as well as for each individual End Point.

4.6.5 mode=

The mode field indicates the mode of the actual resource. The mode is composed of the CommunicationMode and OperationalMode fields.

```
<0x07>mode=<CommunicationMode><OperationalMode>
```

The CommunicationMode and OperationalMode values are received from the Z/IP RD and not processed in any way by the Z/IP mDNS responder.

Refer to [1] for details on mode signaling.

4.6.6 productID=

The Z/IP RD MUST combine the information to a 3 x 16 bit binary key:

```
<length>productid=<ManufacturerID><ProductType><ProductID>
```

Advanced management systems may use this information to retrieve actual product photos and other documentation from available data sources.

4.6.7 product=

The Z/IP RD SHOULD combine the information to a unique human readable string of the following form:

```
<length>product=<ManufacturerString> <ProductString>
```

The product name may always inform the user of the actual product, even when the resource name has been changed from the auto-generated

```
"Acme Dimmer Dx7 [c001babe1201]"
```

Into something more user-centric like this:

```
"table lamp.bedroom.level 2.south wing"
```

4.6.8 productURL= [optional]

A manufacturer SHOULD provide a link to a stable product support page for the actual product. The Z/IP RD may use information from available databases to include a link by looking up the <Manufacturer><ProdType><ProdID> key.

The URL may also lead to dedicated configuration tools, e.g. for advanced configurable remote controls.

Since there MAY be cases where no link is available from a database source, this TXT record is optional. The entire “productURL=” record MUST be omitted completely in this case

4.6.9 securityClass= [optional]

Bit mask of supported security classes. This field MAY be used by a Z/IP client to which Z-Wave nodes who will be able to talk directly to each other.

```
<length> securityClass =<bitmask>
```

The table below shows a list of currently supported bits.

Security Level (1 highest - 4 lowest)	Bit	Key	Description
3	0	Security 2 Class 0	Indicates support for Security 2 Class 0, Unauthenticated devices
2	1	Security 2 Class 1	Indicates support for Security 2 Class 1, Authenticated devices
1	2	Security 2 Class 2	Indicates support for Security 2 Class 2, Access Control devices
4	7	Security 0	Indicates support for Security 0, Secure legacy devices

Table 1

4.7 mDNS Z-Wave Resource Status Notification

For client applications to provide the “connected” experience to users, the Z/IP mDNS responder MUST announce significant changes in the status and availability of resources. As an example, if the Z/IP RD detects that a mailbox node repeatedly does not update its cache entry at the expected time for Wakeup Notifications, the Z/IP RD MUST issue a notification that allows mDNS listeners to update their local caches and eventually icons in a GUI. Examples include a red asterisk next to failing nodes and a read almost-empty battery next to nodes reported to be in a “Low Battery” state via the Operational Mode part of the ‘mode=’ field.

The Z/IP mDNS responder SHOULD NOT maintain any state with regards to these modes. Status management MUST be handled by the Z/IP RD.

4.7.1 Probing new resource names

Before announcing a new resource name, the Z/IP RD MUST request that the Z/IP mDNS responder performs an mDNS probing for the new resource name. This is accomplished by the Z/IP RD indicating the Operational Mode “probing” for the resource name.
mDNS probing involves sending out three queries [3] for a resource name before announcing the resource name.

4.7.2 Announcing resource names

The status “ok” is an indication that the Z/IP RD wants to announce the resource name with the actual resource mode.

Probing of resource names must be performed (see 4.7.1) before announcing new resource names.

4.7.3 Handling resource name changes

<u>This section is just a guideline for Z/IP RD design:</u>

In an mDNS perspective a new resource name is identical to a new resource. As a consequence, the Z/IP RD MUST first indicate that the resource is deleted, probe the new resource name and finally announce announce the new resource name with the proper mode.

The steps are handled as follows:

1. Delete the resource name by indicating Operational Mode “deleting”
   Refer to 4.7.5

2. Probe the new resource name by indicating Operational Mode “probing”.
   Refer to 4.7.1

3. Announce the new resource name by indicating Operational Mode “ok”
   Refer to 4.7.2

4.7.4 Handling failing nodes

A node MUST be considered failing if its Z/IP RD cache entry times out.
If a node implements several resources, all resources of that node MUST be marked as “failing” by the Z/IP RD.

The “failing” Operational Mode MUST be announced for every resource.
The Z/IP RD indicates that resources are failing by updating the Operational Mode in the TXT “mode=” record.

Note: The RD cache time out causing the Operational Mode “Failing” should not be confused with an mDNS listener timing out its cache entry. If an mDNS cache times out, the mDNS API will signal a “Did Remove Service”. The mDNS Goodbye Packet is just an expedited mDNS cache timeout. Refer to section 4.7.5 on how to handle the mDNS API “Did Remove Service” indication.

4.7.5 Handling deleted nodes

In case multiple resources (Z-Wave End Points) are hosted by the same physical entity (IP address) the Z/IP RD MUST delete each of the resources.

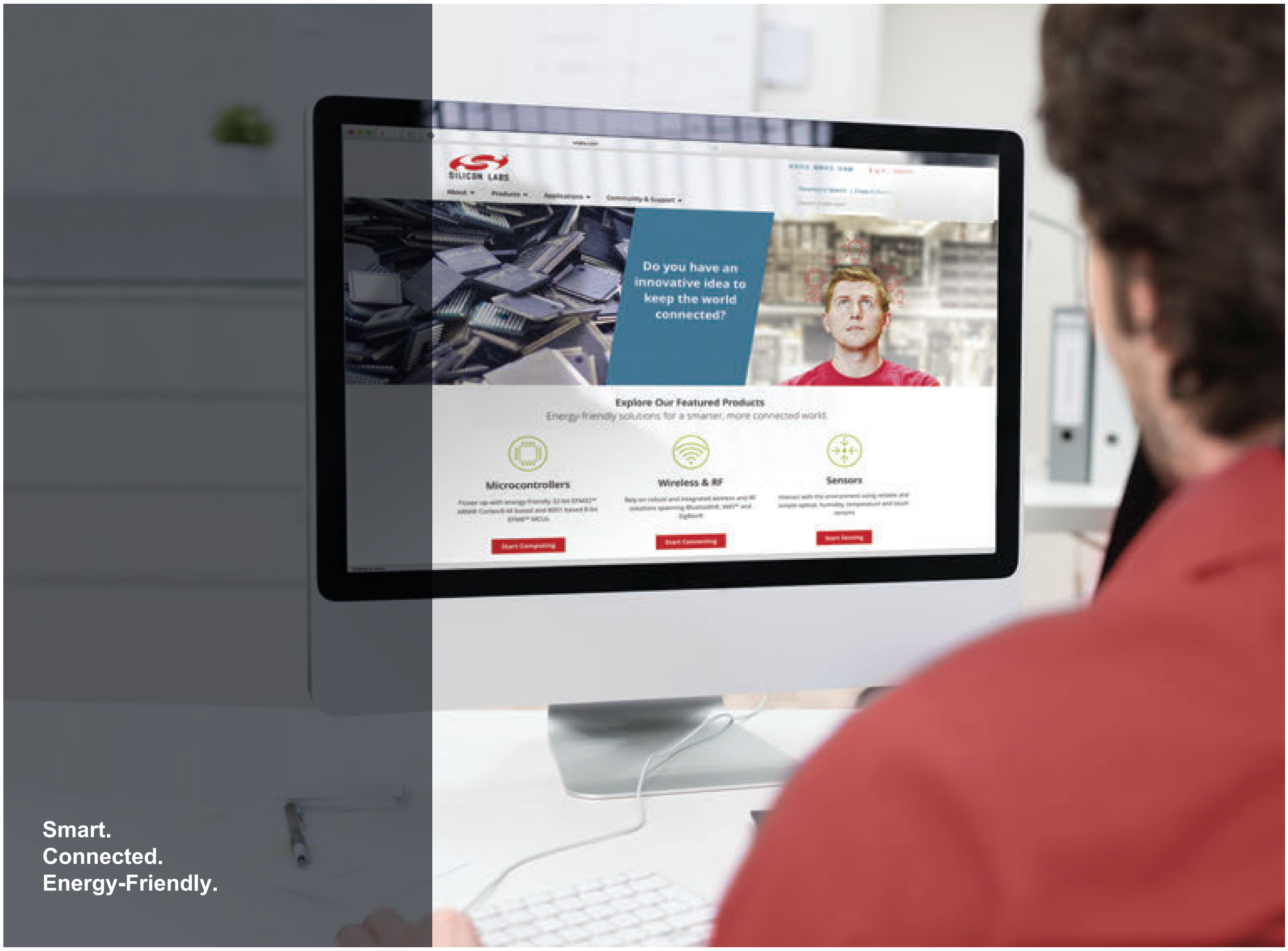
When notified by the Z/IP RD, the Z/IP mDNS responder MUST generate an mDNS Goodbye packet as per the guidelines of [3]. The mDNS Goodbye Packet is just an expedited mDNS cache timeout.

If an mDNS cache entry times out, the mDNS API will signal a “Did Remove Service”. It does not mean that the resource providing that service is gone forever. It may be that the gateway just went offline or that the Z/IP client was moved to another physical location out of reach of the gateway. In this case, group and scene relationships should be maintained by the Z/IP Client.
Thus, a Z/IP Client SHOULD keep information forever regarding resources that were once announced via mDNS. If the Z/IP Client receives a “Did Remove Service” indication from – or it does not receive a

“Did Find Service” during application initialization – the actual resource SHOULD be hidden from user interfaces.
It MUST however be possible to enable the display of all known resources so that the user can demonstrate the interface without having actual connectivity. This also allows the user to manually remove resources that the user knows will never return to the network.
When the display of offline devices is enabled, the Z/IP Client SHOULD clearly indicate for each resource that it is offline.

REFERENCES

[1] Silicon Labs, SDS11633, Z/IP Resource Directory
[2] IETF, RFC2119, Key words for use in RFCs to Indicate Requirement Levels
[3] IETF, RFC6762, Multicast DNS
[4] IETF, RFC6763, DNS-Based Service Discovery
[5] Stuart Cheshire & Daniel Steinberg, Zero Configuration Networking: The Definitive Guide, ISBN:0596101007
[6] IETF, Lynn & Shelby, draft-lynn-core-discovery-mapping, CoRE Link-Format to DNS-Based Service Discovery Mapping
[7] IETF, Shelby & Krco, draft-shelby-core-resource-directory, CoRE Resource Directory
[8] IETF RFC4193, Unique Local Address (ULA)
[9] Silicon Labs, SDS12738, Z-Wave Plus Assigned Icon Types.


Smart.
Connected.
Energy-Friendly.
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Microcontrollers
Wireless & RF
Sensors

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com