Bose SoundTouch Speaker Discovery

Bose Corporation

Version 1.1.0

Contents

1 Document Version History

Version	*Release Date*	*Description of Changes*
1.0.0	December 5, 2014	• Initial Release
1.0.1	December 17, 2014	• Section 3 updated with a link to the License Agreement.
1.1.0	February 5, 2016	• Section 4 updated to include new product • Section 5 removed legacy controller reference

2 Acronyms and Definitions

Acronyms	*Expanded Term*	*Definition*
API	Application Programming Interface	A definition for how to interact with and use a software component
REST	Respresentational State Transfer	A common type of web service API that is modeled around resources
WS API	Web Service API	An API made available by a web server
SSDP	Simple Service Discovery Protocol	A discovery protocol that uses unicast and multicast over UDP
MDNS	Multicast Domain Name System	A type of discovery protocol that requires zero configuration
	Bonjour	Apple's implementation of Zero-configuration

3 Contact Info/Legal

For any questions, comments, or suggestions for improvements please email us at SoundTouchAPI@bose.com

Use of this API material is subject to the API License Agreement, which can be found at developers.bose.com/SoundTouch-API-License

4 Overview

This document intends to describe the methods by which a SoundTouch speaker can be discovered on a given network. The SoundTouch ecosystem will be broken down in to individual components and the roles and relationships of these components will help illustrate working examples of discovery in a number of different scenarios. This document will not cover the Web Services API provided by the SoundTouch speakers.

5 SoundTouch Ecosystem

The SoundTouch ecosystem consists of a variety of components that work in harmony to deliver a unified experience to a customer. The following components exist:

- SoundTouch Speaker
- SoundTouch App
- SoundTouch Cloud Server
- SoundTouch Controller

5.1 SoundTouch Experience

The SoundTouch unified experience starts with a customer's home network. The SoundTouch speaker and the SoundTouch App work together during the out of box setup process to configure a SoundTouch speaker to a customer's home network. This setup can happen via wired ethernet or via wifi configuration in order to connect the SoundTouch speaker on to the customer's home wifi access point. The App then performs an update to a newer SoundTouch speaker firmware version if available and proceeds to the SoundTouch controller pairing process if it is available. After all this, the SoundTouch App performs the final step of associating the speaker to the customer's acount stored in the SoundTouch cloud.

5.2 SoundTouch Speaker

This is a speaker that may be in the form of one of the following available products:

- SoundTouch 10 music system - small, mono speaker with LEDs and built in buttons on the top of the speaker
- SoundTouch Portable music system - a battery powered, small, portable mono speaker with a display and built in buttons on the top of the speaker
- SoundTouch 20 music system - a medium sized one-piece speaker with display and built in buttons on the top of the speaker
- SoundTouch 30 music system - a large sized one-piece speaker with display and built in buttons on the top of the speaker
- Wave SoundTouch music system
- SoundTouch Stereo JC music system
- SoundTouch SA-4 amplifier package
- SoundTouch SA-5 amplifier package
- SoundTouch (1xx, 2xx) outdoor speaker systems
- SoundTouch (1xx, 5xx) home theater systems
- Lifestyle SoundTouch home theater systems

5.3 SoundTouch App

This is the rich user interface available as an app that runs on Apple Macs, Microsoft PCs. It also runs as a native app on Apple iOS devices and Google Android devices.

The role of the App in general is to find SoundTouch speakers on the network using one of the discovery protocols (SSDP, Bonjour) described in this document.

- SoundTouch App - Apple Mac OSX
- SoundTouch App - Microsoft Windows/PC
- SoundTouch App - Apple iOS
- SoundTouch App - Google Android
- SoundTouch App - Amazon Kindle

5.4 SoundTouch Cloud Server

The SoundTouch cloud creates, stores, and updates accounts in the cloud. A cloud server provides a central storage location for all things related to a particular SoundTouch account. This includes things like the list of SoundTouch speakers available to the account. Examples of some other information stored on the cloud include some of the following:

- A list of associated SoundTouch speaker products and related info such as the speaker name
- A list of recents for each product
- A list of presets for each product
- A list of logged-in music services/sources in the account

5.5 Putting It Together

Each of the components working together form a SoundTouch ecosystem. Any given ecosystem may combine one or more SoundTouch speakers, one or more SoundTouch apps, along with a single home network and a single SoundTouch cloud account. Within the SoundTouch app, the device tray will present a list of SoundTouch speakers provided by the SoundTouch cloud account. Each speaker will show up in one of two possible states depending on the outcome of device discovery on the home network between the SoundTouch app and the SoundTouch speakers. If a speaker is not discovered using one of the discovery protocols, then the speaker will be presented as a dotted outline in the device tray. If the speaker is discovered and is available, then the speaker will be presented as a solid-colored image within the device tray, indicating that it is ready to be interacted with.

In this example of how the app searches for SoundTouch speakers, it demonstrates the two roles played where the app plays the role of the CLIENT (searcher, seeker) and the speaker plays the role of the SERVICE PROVIDER (announcer, advertiser).

Third party developers integrating SoundTouch control functionality into apps or systems will need to implement these discovery protocols in order to find and communicate with SoundTouch devices on the network. Throughout this document, developers should use the SoundTouch App as the example to follow for their own implementations.

6 Discovery Protocols

There are two discovery protocols that are used within the SoundTouch ecosystem. The first is Simple Service Discovery Protocol, and the second is Zero-configuration which has implementations known as Bonjour, MDNS, and Avahi. The protocols provide redundancy and variation which work in favor of providing more reliable discovery in a number of different environments where port availability might vary.

6.1 SSDP (Simple Service Discovery Protocol)

6.1.1 SSDP Client

When a client wants to find devices that provide a specific service, it will do an M-SEARCH over UDP multicast port 1900, with a service type header containing the type of service it is looking for.

The two services used are

- urn:schemas-upnp-org:device:MediaRenderer:1 for devices that can play audio (SoundTouch speakers)
- urn:schemas-upnp-org:device:MediaServer:1 for devices that contain media (SoundTouch app and music server)

A client that wants to use services must continuously listen to UDP multicast and unicast port 1900 to receive replies to searches and also announcements from service providers.

A search should only be performed once per service type while the client is continuously listening. If the client stops listening for some period of time then wants to start using services again it must start listening again and perform another M-SEARCH.

When a service provider replies to a search or announces itself, it will provide a Location header and an expiration time in the Cache-control:max-age header. The UPnP specification specifies that the max-age must be a minimum of 1800 seconds, or 30 minutes.

All further communication to the service provider should be done via the address contained in the Location header.

The client must keep track of the device expiry. If it has not heard a new announcement containing a new expiration time from a service provider by the expiry time, it must consider the device no longer available. This is to handle devices that go offline unexpectedly.

6.1.2 SSDP Service Provider

A device that wants to provide services must continuously listen to UDP multicast port 1900 to receive search requests from clients.

When a device becomes available and wants to provide services it must send a NOTIFY over UDP multicast port 1900 for each service type it provides along with an ssdp:alive, its uuid (via the USN header) and an expiration time (in the Cache-control: max-age header). The uPnP specification specifies that max-age must be a minimum of 1800s (30m); it also recommends that initial notifications be sent more than once, but not more than three times, at random intervals between 0 and 100ms.

When a device receives an M-SEARCH request for a service it provides it must send a reply back via UDP unicast port 1900 to the originating requester containing its uuid (in the USN header) and an expiration time (in the Cache-control: max-age header). The uPnP specification specifies that max-age must be a minimum of 1800s (30m).

The device must keep track of when its notification/reply expiration times are about to occur and re-send another NOTIFY with a new expiration time before the expiration happens. The uPnP spec recommends that new NOTIFY messages be sent at randomly distributed intervals of less than 1/2 the expiration time.

When a device's services become unavailable it must send a NOTIFY over UDP multicast port 1900 for each service type that will be unavailable along with an ssdp:byebye.

If the device has multiple network interfaces, it must listen and send notifications on all interfaces that it wishes to provide services on. I.e. it's up to the device to decide which interfaces it wants to provide services on and then listen, announce, and reply on those interfaces.

6.2 MDNS

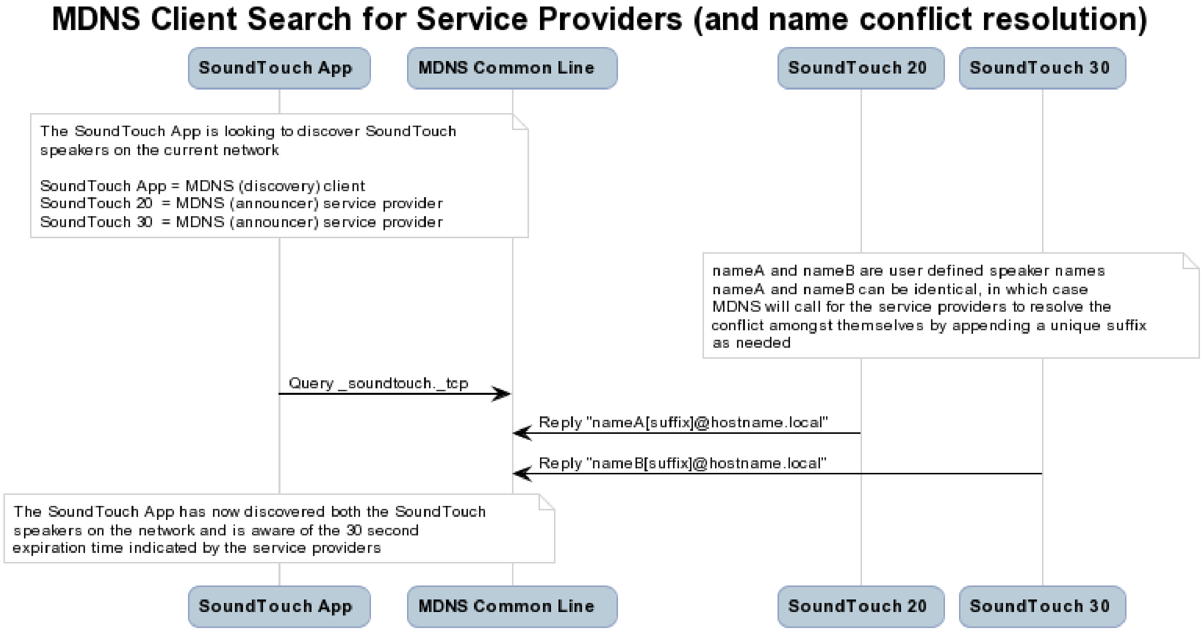
MDNS is the protocol that has implementations that go by the name of Zero-configuration, Bonjour, and Avahi. Each are different implementations of the same discovery protocol that allow systems on the same network to automatically discover each other and their capabilities via advertised services. For service providers running Zero-configuration, the service types used are:

- _soundtouch._tcp.local for general SoundTouch capabilities
- _raop._tcp.local for specific Airplay capabilities if available on a particular SoundTouch speaker

7 Examples

7.1 SSDP

7.1.1 Client search and service providers reply:



MDNS Client Search for Service Providers (and name conflict resolution)

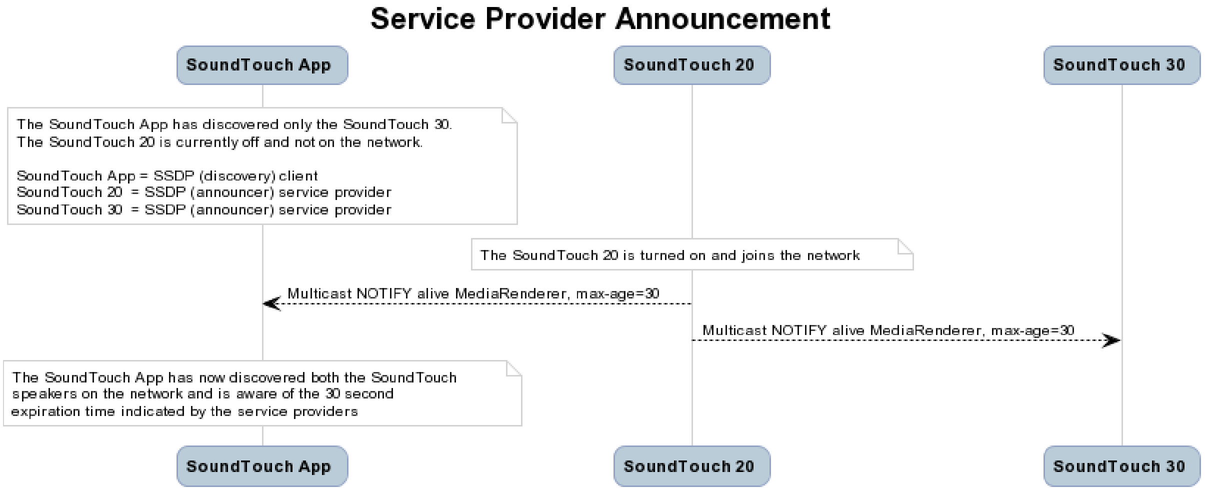
The SoundTouch App is looking to discover SoundTouch speakers on the current network

SoundTouch App = MDNS (discovery) client
SoundTouch 20 = MDNS (announcer) service provider
SoundTouch 30 = MDNS (announcer) service provider

nameA and nameB are user defined speaker names
nameA and nameB can be identical, in which case MDNS will call for the service providers to resolve the conflict amongst themselves by appending a unique suffix as needed

The SoundTouch App has now discovered both the SoundTouch speakers on the network and is aware of the 30 second expiration time indicated by the service providers

7.1.2 Service provider comes online and announce itself:



The SoundTouch App has discovered only the SoundTouch 30.
The SoundTouch 20 is currently off and not on the network.

SoundTouch App = SSDP (discovery) client
SoundTouch 20 = SSDP (announcer) service provider
SoundTouch 30 = SSDP (announcer) service provider

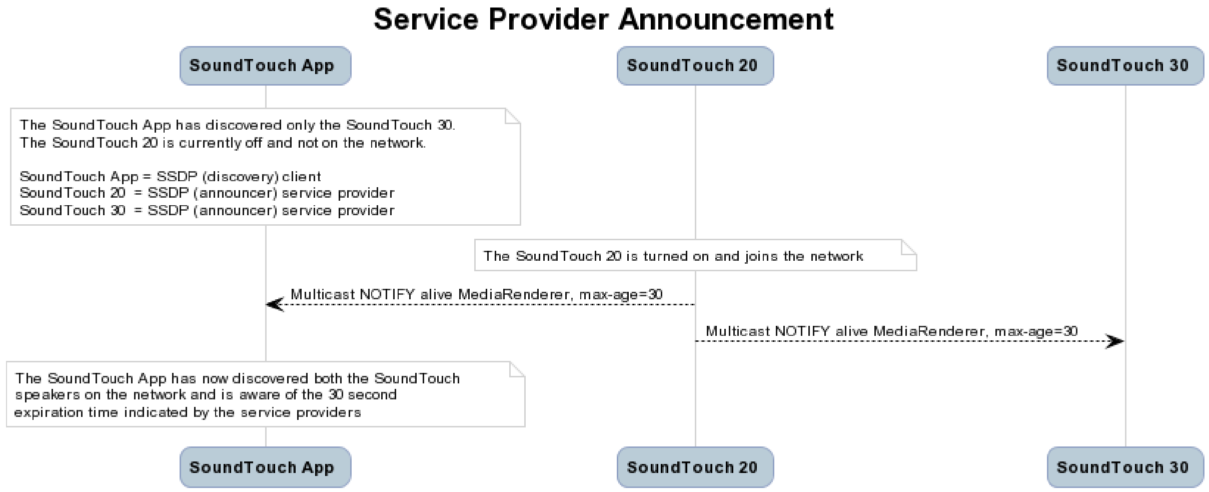
The SoundTouch 20 is turned on and joins the network

Multicast NOTIFY alive MediaRenderer, max-age=30

Multicast NOTIFY alive MediaRenderer, max-age=30

The SoundTouch App has now discovered both the SoundTouch speakers on the network and is aware of the 30 second expiration time indicated by the service providers

7.1.3 Service provider's previous announcement is about to expire so it announces itself:


Service Provider Announcement

SoundTouch App
SoundTouch 20
SoundTouch 30
Multicast NOTIFY alive MediaRenderer, max-age=30
Multicast NOTIFY alive MediaRenderer, max-age=30
SoundTouch App
SoundTouch 20
SoundTouch 30

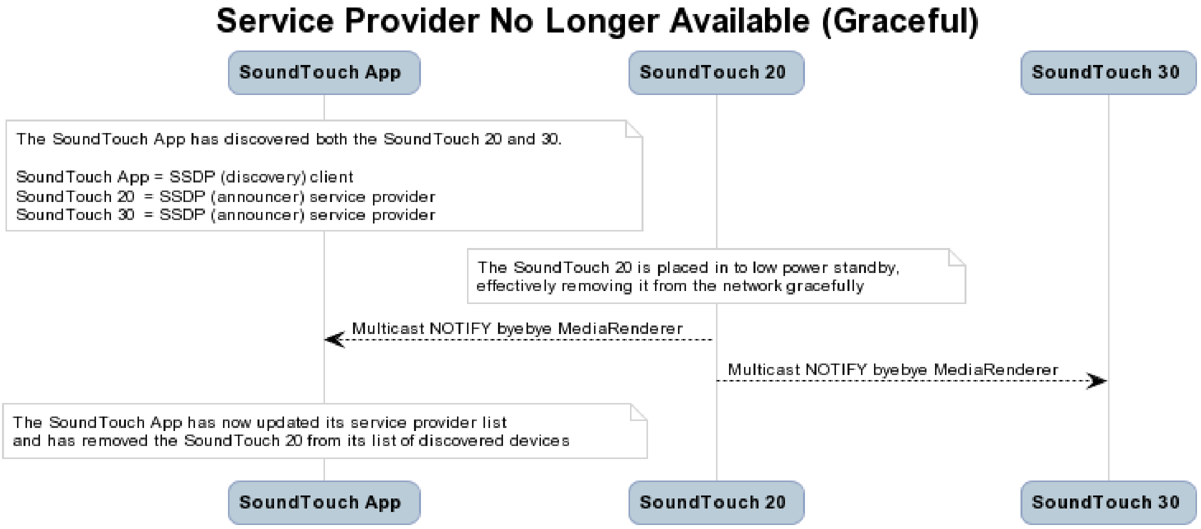
The SoundTouch App has discovered only the SoundTouch 30.
The SoundTouch 20 is currently off and not on the network.

SoundTouch App = SSDP (discovery) client
SoundTouch 20 = SSDP (announcer) service provider
SoundTouch 30 = SSDP (announcer) service provider

The SoundTouch 20 is turned on and joins the network

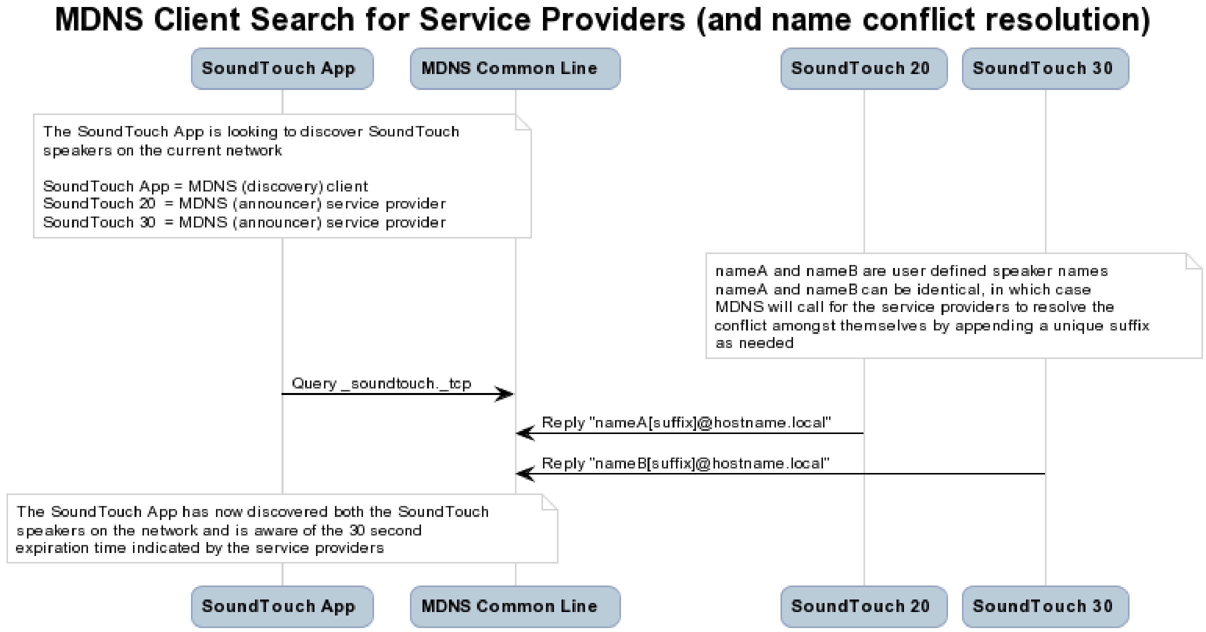
The SoundTouch App has now discovered both the SoundTouch speakers on the network and is aware of the 30 second expiration time indicated by the service providers

7.1.4 Service provider becomes unavailable due to a graceful shutdown:


Service Provider No Longer Available (Graceful)
SoundTouch App
SoundTouch 20
SoundTouch 30
The SoundTouch App has discovered both the SoundTouch 20 and 30.
SoundTouch App = SSDP (discovery) client
SoundTouch 20 = SSDP (announcer) service provider
SoundTouch 30 = SSDP (announcer) service provider
The SoundTouch 20 is placed in to low power standby, effectively removing it from the network gracefully
Multicast NOTIFY byebye MediaRenderer
Multicast NOTIFY byebye MediaRenderer
The SoundTouch App has now updated its service provider list and has removed the SoundTouch 20 from its list of discovered devices
SoundTouch App
SoundTouch 20
SoundTouch 30

7.2 Zero-configuration

7.2.1 Client searches for service providers


MDNS Client Search for Service Providers (and name conflict resolution)

The SoundTouch App is looking to discover SoundTouch speakers on the current network

SoundTouch App = MDNS (discovery) client
SoundTouch 20 = MDNS (announcer) service provider
SoundTouch 30 = MDNS (announcer) service provider

nameA and nameB are user defined speaker names
nameA and nameB can be identical, in which case MDNS will call for the service providers to resolve the conflict amongst themselves by appending a unique suffix as needed

The SoundTouch App has now discovered both the SoundTouch speakers on the network and is aware of the 30 second expiration time indicated by the service providers

8 Additional Resources

8.1 Software Tools For Development and Debugging

Some of the following tools might be of use during development and debugging:

- Wireshark
- Avahi
- Bonjour Browser
- A DLNA controller that uses SSDP to find a media renderer device

8.2 Web Resources For Further Information

- http://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol
- http://en.wikipedia.org/wiki/Multicast_DNS

9 Beyond Discovery

Discovery is the first phase in allowing the SoundTouch speakers to be found prior to engaging in communication with the speaker via the available WS API. This document does not cover the details of what is provided by the WS API, please refer to the Bose SoundTouch WS API document for more information about the SoundTouch WS API.