mDNS and DNS-Service Discovery Background Information

DNS Service Discovery is a way of using standard DNS programming interfaces, servers and packets to browse the network for services. DNS-Based Service Discovery is only peripherally related to multicast DNS, in a way that the standard unicast DNS queries used by DNS-SD may also be performed using multicast when appropriate, which is particularly beneficial in Zeroconf environments .

Multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source. Copies are automatically created in other network elements, such as routers, but only when the topology of the network requires it.

Zero Configuration Networking (zeroconf), or Bonjour as Apple calls it, is a networking technology that allows devices and services to be automatically discovered in networks that do not have central DNS server. In other words Bonjour uses Multicast DNS Service Discovery (mDNS-SD) which is a combination of mDNS and DNS-SD standards.

DNS service discovery (DNS-SD) helps in registering a service, browsing for services and resolving service names to host names. The mDNS-SD is query driven and retrieves the type of service, the name of the service, IP and port addresses and other optional information.

Bonjour comes pre-installed on iOS, unix operating systems have equivalent implementation called Avahi which is is installed and configured by default in the desktop versions of Ubuntu and many other unix operatyng systems. However on Windows it needs to be installed by the user unless the user has already installed some software that includes Bonjour such as iTunes. Bonjour for Windows includes a System Service that helps applications discover shared services on the local network.

Secfone Service Broadcast

The idea is to use Android Network Service Discovery (which is supported since API level 16 i.e. Jelly Bean) and provide the user with the ability to register a service from the Secfone app. This will allow the user, when it shares the connection via hotspot or usb tethering, to use the Secfone services on the device with which it shares the connection. Of course for the Secfone service to be discovered on other devices, those other devices need to have Bonjour, Avahi or some other service discovery software. Bonjour and Avahi always run in background and perform network service discovery which consists of resolving service names to host names and updating the host files on that device with the IP addresses and host names found.

In the particular case for using the email service proxy tasks on the Secfone, the user only needs to set the incoming and outgoing server name to android.local (instead of mail-jb.navayo). Also the port numbers should be set to the appropriate ones that are set in the proxy tasks on the Secfone.

The only drawback to this simple setup is the hostname i.e. the android part from android.local which cannot be changed since it is forced by the Android system. In the scenario where multiple Android devices broadcast a service only the first device registered on the network receives a hostname android.local .Each of the following devices gets a hostname in the form android-n.local where n is a number that corresponds to the order in which the devices register their services and is forced by the Android system.

The solution to this drawback is to write Thunderbird plugin that will discover the IP address according to the service name and not the hostname of the device, where the service is being hosted. The Poco C++ libraries that we already use in other projects provide DNS-SD and service name resolution and offer cross-platform support for Windows, Unix and iOS.