



# mDNS

## Multicast DNS (MDNS) on Home Networks

Multicast DNS is part of Zero-configuration networking (zeroconf) set of technologies designed to enable devices to work on networks without manual setup.

Multicast DNS is used to locate a device or service by name on a small local network without using a preconfigured name server i.e DNS.

Originally developed by Apple and named "Bonjour", it is now an Internet standard named 'Multicast DNS RFC 6762'.

Multicast DNS uses the same packet structure and commands as DNS, but doesn't rely on a user configured DNS server. Instead computers on a network create their own local DNS records and store them locally in memory cache.

### mDNS on Linux and Windows

On Linux, including the Raspberry Pi, it is usually installed automatically and uses Avahi packages. You can also download the utilities which can be useful for troubleshooting using:

```
sudo apt -y avahi-utils
```

On Windows and Apple the mDNS services are provided by the Bonjour packages.

You can download the Bonjour SDK here <https://developer.apple.com/bonjour/> which will allow you to use the dns-sd command line tool.

**Note:** You will need to create a developer account to download the SDK.

### How mDNS Works

It is useful to compare mDNS with the traditional DNS service especially if you are familiar with DNS. When a host requires to know the IP address of another host it sends a DNS query to a DNS server using a unicast UDP message on port 53.

The address of this DNS server is part of the network configuration of the host.

With mDNS no DNS server is preconfigured, and the host uses a multicast query on IPv4 address 224.0.0.251 or IPv6 address ff02::fb and UDP port 5353.

**Note:** Multicast is a method on IP networks of sending data to a group of computers.

All mDNS hosts see this query and the host that has this network name responds to the query using a multicast message that contains its IP address.

Because messages were exchanged using multicast all other mDNS hosts see this exchange and can make a note of the network name and IP address, and store this in their local cache.

All mDNS hosts now know the name and IP address of the requested host.

So the process is:

1. Host A sends query for IP address of Host Z.
2. All mDNS hosts see this query.
3. Host Z responds with its IP address
4. All mDNS hosts see this response.
5. All mDNS hosts update their local cache with the name host Z and its IP address.

### The .local Domain Name

All DNS entries have a domain name and for devices and services on local networks that aren't part of the global DNS namespace the .local domain name was reserved.

This means that all hosts on a local network will have a name of the form.

```
host1.local
```

The mDNS resolver on the client registers the host with the .local prefix. So when you do a ping to a computer called `ws6` on your local network you would use:

```
ping ws6.local
```

The local name resolver will automatically choose mDNS as a first choice to resolve the name because of the presence of the .local suffix.

## mDNS Records

mDNS supports both host name records (A and AAAA) record types (and SRV records) as in standard DNS. Because of this, mDNS can be used for locating hosts (and services) on a local network.

**Note:** The service record structure can appear confusing at first glance for any one new to DNS service records.

The structure is shown below along with an example record:

```
_service._proto.name. TTL class SRV priority weight port target
_sip._tcp.example.com. 86400 IN SRV 0 5 5060 sipserver.example.com.
```

Here the service is `_sip`, the proto name is `tcp`, protocol service used. The name is `example.com.`, domain name for which it is valid. Notice the use of the underscores in the structure. At the end of the record we have `port` and `target`. The `port` is `5060`, the port the service runs on and the `target` is `sipserver.example.com`, the host that runs this service.

For details on how to add service records, read <http://www.dns-sd.org/serverstaticsetup.html>

## Announcements

As part of the mDNS protocol mDNS devices will make announcements containing their mDNS records on start and in response to network changes on the host machine.

These announcements will be received by all mDNS clients on the local network and used to update their own records.

## Finding Hosts Using mDNS

This can be demonstrated using the `avahi-resolve` utility as shown below

```
# avahi-resolve --name oldies.local
Oldies.local 192.168.1.42
```

On Windows use the `dns-sd -Q hostname` command

```
> dns-sd -Q MeRasPi4B-Test.local
Timestamp A/R Flags if Name T C Rdata
9:39:33.633 Add 2 20 MeRasPi4B-Test.local. 1 1 192.168.1.43
```

## Service Discovery Using mDNS

mDNS can be used to discover services like MQTT on your local network. An mDNS client issues a query for that service as shown.

On Raspberry Pi/Linux, by using the `avahi-browse` utility

```
# avahi-browse _mqtt._tcp
+ eth0 IPv4 Mosquitto MQTT server on Oldies _mqtt._tcp local
```

You can see that this service is available on machine called `Oldies` and on IPv4.

On Windows use the `dns-sd -B service name` command

```
>dns-sd -B _mqtt._tcp
Browsing for _mqtt._tcp
Timestamp A/R Flags if Domain Service Type Instance Name
9:39:33.633 Add 2 4 local. _mqtt._tcp Mosquitto MQTT
```

## Note:

- You can still use a local domain name server like DNSMasq if clients use mDNS as they can co-exist on the same network.
- MDNS can not be used across VLANs without extra configuration in your switches and routers
- You can use a host announce services that are available on other machines on the network. On Linux you will need to create and add a `avahi services` file.