



DNSMasq

Home Network DNS Server Using DNSMasq

Records of Domain names and IP addresses are maintained on DNS servers located in many locations.

Home Network Devices and DNS

Home network devices like computers don't have a public domain name and hence don't have, or need an entry in the global, public, DNS database.

So how do you locate devices on your home network if , for example, you need to access a local file share or printer?

Well DNS isn't the only name resolution option and others methods are possible, and used. Available methods are : broadcast, hosts, wins, etc

However with the growing use of smart devices using a local DNS server for local name resolution is becoming popular.

The DNSMasq DNS Server

DNSMasq is a free and DNS and DHCP server for small computer networks and is provided with most Linux distributions.

Understanding The DNS Name Resolution Process

When you type in a web address in your web browser the web browser will first use the DNS protocol to contact the DNS server that has been configured for that device.

The DNS server will respond to the resolution request with the IP address of the website or return a not found message. Once the browser has the IP address of the website it can connect to it. If the DNS process fails you get a server not found displayed in the browser.

This process is the same for all applications e.g. email, Skype etc. They all rely on DNS working in the background.

DNS Server Addresses

The addresses of the DNS servers are usually given to the client machine by a DHCP server, but can also be assigned manually.

Two DNS server addresses are normally assigned for redundancy purposes.

On Windows you can view the addresses using the `ipconfig /all` command.

```
IPv4 Address. . . . . : 192.168.1.42 (Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 27 January 2021 03:31:26
Lease Expires . . . . . : 28 January 2021 03:31:29
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DNS Servers . . . . . : 192.168.1.1
                        8.8.4.4
                        195.130.130.2
```

On linux you obtain that info using

```
# cat /etc/resolv.conf
# Generated by resolvconf
domain engrie.local
nameserver 192.168.1.1
nameserver 8.8.4.4
nameserver 195.130.130.2
```

Setting Up and Using Your Own Local DNS Server

Installing DNSMasq

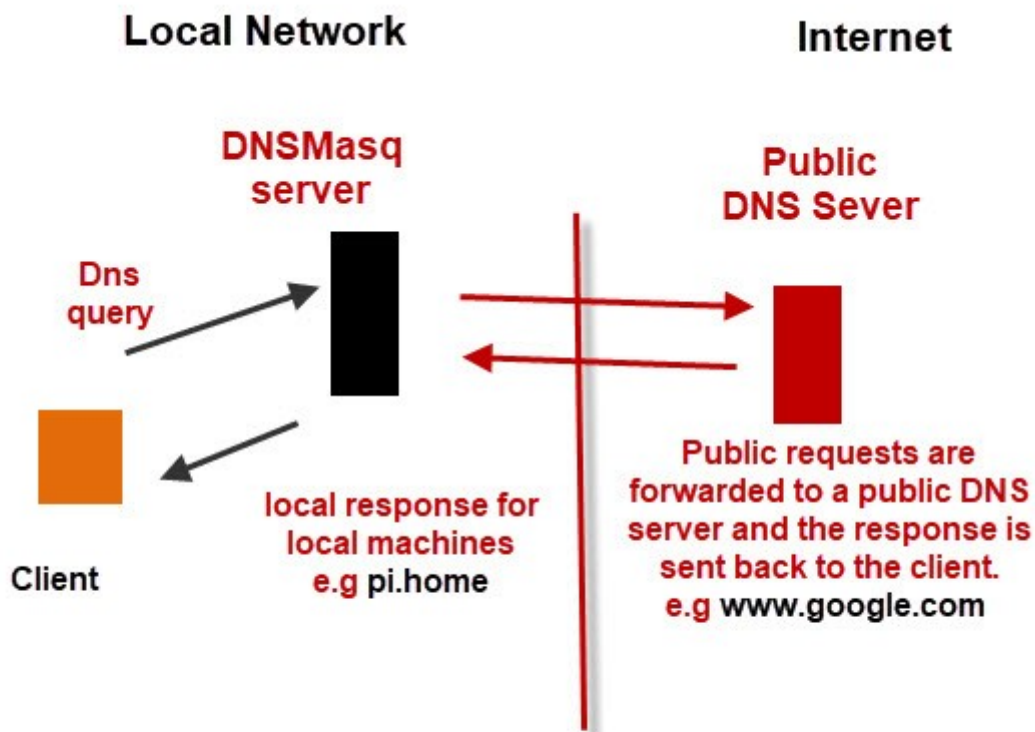
Before you install DNSMasq it is important that your machine has a fixed IP address. You can do this locally by editing the `dhcp.conf` file or using the network GUI utility. However, the best way is to configure it at the router level. Almost all routers let you assign a fixed IP address to a device.

To install on Raspberry Pi use:

```
sudo apt -y install dnsmasq
```

Configuring DNSMasq

Like almost all Linux applications, configuration is via a text file `/etc/dnsmasq.conf`. The way it is configured is that all local DNS requests are handled directly by the DNSMasq server and others that are for external resources are forwarded to the DNS servers you normally use. This is shown in the diagram below:



DNSMasq on A local Network

Although DNSMasq can be used as a DHCP server as well, I use only for a DNS server and so the DHCP configuration isn't used.

Although you can use the `/etc/dnsmasq.conf` file and uncomment the settings you need I prefer to use my own and copy the original file for safe keeping.

```
cp /etc/dnsmasq.conf /etc/dnsmasq.conf.orginal
```

Local Network Domain Name

Although you don't actually need one I think it best to use one.

Private DNS Namespaces recommends these names for internal networks

- .intranet
- .internal
- .private
- .corp
- .home
- .lan

Note you should not use .local as it is use by mDNS.

Example Configuration File

This is a sample configuration file

```
nano /etc/dnsmasq.conf
```

```
# Add other name servers here, with domain specs if they are for
# non-public domains.
server=8.8.8.8
server=195.130.130.2
```

```
# Add local-only domains here, queries in these domains are answered
# from /etc/hosts or DHCP only.
local=/home/
```

```
# Set the cachesize here.
cache-size=150
```

you see a .home domain name is used. So all machines will be of the form name.home this you can see in the hosts file.

You can speed up DNS requests for you home network by increasing the cache size using

```
cache-size=5000
```

The default is 150 records. You can even use a large number as each record is only 100 bytes but I'm not sure if you will notice the difference.

Hosts File

DNSMasq uses the local `/etc/hosts` file for the machine names so you will need to edit this with the names of your local machines. Below is a sample `/etc/hosts` file.

```
nano /etc/hosts

127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

127.0.1.1     MeRasPi4B-Test
127.0.1.1     MeRasPi4B-Test.home
192.168.1.42  Oldies
192.168.1.42  Oldies.home

# fake for testing to make sure
192.168.255.255 test
192.168.255.255 test.home
```

If you make changes to the hosts file you will need to restart DNSMasq to pick up these changes.

```
#start the program
sudo systemctl start dnsmasq

#stop the program
sudo systemctl stop dnsmasq

#restart the program
sudo systemctl restart dnsmasq

#start dnsmasq on boot
sudo systemctl enable dnsmasq

#get status of dnsmasq
sudo systemctl status dnsmasq
```

```
root@MeRasPi4B-Test:/home/pi # sudo systemctl status dnsmasq
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-01-27 10:39:06 CET; 19s ago
     Process: 4097 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
     Process: 4098 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=0/SUCCESS)
     Process: 4108 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf (code=exited, status=0/SUCCESS)
   Main PID: 4107 (dnsmasq)
     Tasks: 1 (limit: 4915)
   CGroup: /system.slice/dnsmasq.service
           └─4107 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -r
             /run/dnsmasq/resolv.conf -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-ne

Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: reading /run/dnsmasq/resolv.conf
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using local addresses only for domain home
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using nameserver 195.130.130.2#53
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using nameserver 8.8.8.8#53
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using nameserver 192.168.1.1#53
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using nameserver 8.8.4.4#53
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: using nameserver 195.130.130.2#53
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4107]: read /etc/hosts - 8 addresses
Jan 27 10:39:06 MeRasPi4B-Test dnsmasq[4108]: Too few arguments.
Jan 27 10:39:06 MeRasPi4B-Test systemd[1]: Started dnsmasq - A lightweight DHCP and caching DNS server.
```

Testing DNSMasq

Before you configure you clients to use it you will need to test that it works as expected. To do this, use the `nslookup` tool on Windows. The follwing screen shot is taken from a Windows 10 machine.

The first thing to do is to choose to use the DNSMasq server which is at IP address 192.168.1.43.

```
>nslookup
Default Server:  UnKnown
Address:  192.168.1.1

> server 192.168.1.43
Default Server:  [192.168.1.43]
Address:  192.168.1.43
```

I just enter a few names that I know are configured and then test external domain names using eg. google.

```
> oldies.home
Server:  [192.168.1.43]
Address:  192.168.1.43

Name:    oldies.home
Address:  192.168.1.42

> test.home
Server:  [192.168.1.43]
Address:  192.168.1.43

Name:    test.home
Address:  192.168.255.255

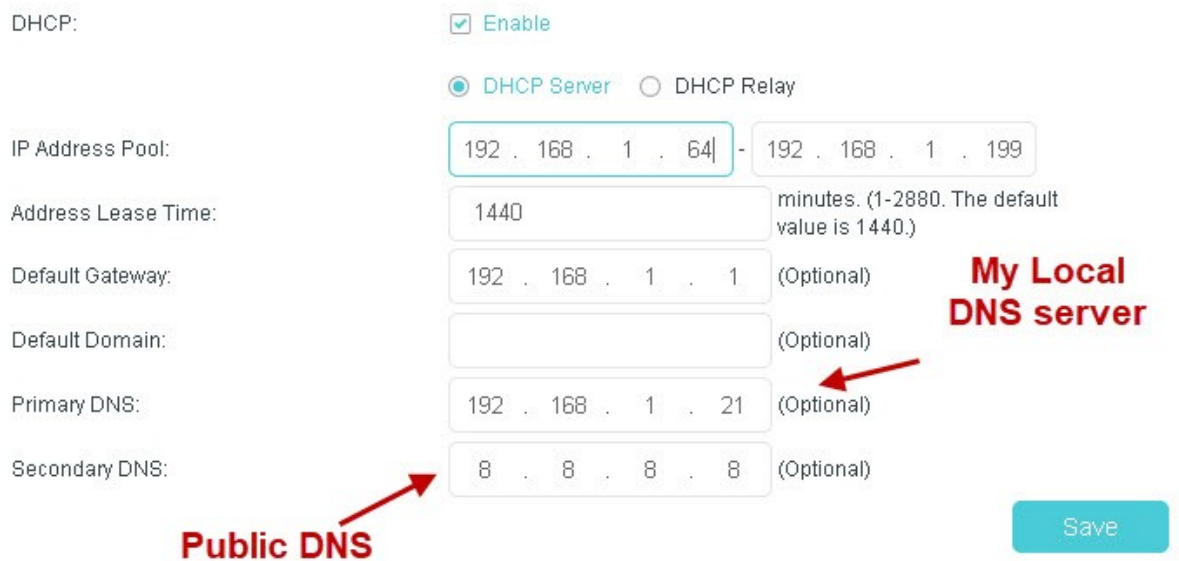
> www.google.com
Server:  [192.168.1.43]
Address:  192.168.1.43

Non-authoritative answer:
Name:    www.google.com
Addresses:  2a00:1450:400e:807::2004
           172.217.17.132

> quit
```

Configuring the Clients

The easiest way and recommended way is to use your DHCP server to assign the DNS address. Because two addresses are assigned you will assign the local address and an Internet server address.



The screenshot shows a DHCP configuration interface with the following fields and values:

- DHCP: Enable
- DHCP Server DHCP Relay
- IP Address Pool: 192 . 168 . 1 . 64 - 192 . 168 . 1 . 199
- Address Lease Time: 1440 minutes. (1-2880. The default value is 1440.)
- Default Gateway: 192 . 168 . 1 . 1 (Optional)
- Default Domain: (Optional)
- Primary DNS: 192 . 168 . 1 . 21 (Optional)
- Secondary DNS: 8 . 8 . 8 . 8 (Optional)

Annotations in red text with arrows:

- Public DNS** with an arrow pointing to the Secondary DNS field (8 . 8 . 8 . 8).
- My Local DNS server** with an arrow pointing to the Primary DNS field (192 . 168 . 1 . 21).

A **Save** button is located at the bottom right of the form.

You then need to wait until the clients renew their IP address or reboot your clients and they will pick up the DNS server.

Although a Local DNS server isn't really necessary for most home networks, it can make Internet activity faster as many addresses will be cached locally.

Also, it is not necessary for Home Automation but it will make it easier.

If your local DNS server is unavailable, clients will use the public DNS server that you configured but not all will be resolved if you added local devices in the hosts file.

It would be ideal to just use mDNS but not all clients support it.