resolv.conf

The configuration file for DNS resolvers is `/etc/resolv.conf`.
From **resolv.conf(5) (https://jlk.fjfi.cvut.cz/arch/manpages/man/resolv.conf.5)**:

> The resolver is a set of routines in the C library that provide access to the Internet Domain Name System (DNS). The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.
>
> If this file does not exist, only the name server on the local machine will be queried; the domain name is determined from the hostname and the domain search path is constructed from the domain name.

Related articles

Improving performance#Networ

Contents

DNS in Linux

Your ISP (usually) provides working **DNS** servers, and a router may also add an extra DNS server in case it has its own cache server. Switching between DNS servers does not represent a problem for Windows users, because if a DNS server is slow or does not work it will

immediately switch to a better one. However, Linux usually takes longer to timeout, which could be the reason why you are getting a delay.

Testing

Use *drill* (provided by package **ldns (https://www.archlinux.org/packages/?name=ldns)**) before any changes, repeat after making the adjustments and compare the query time(s). The following command uses the nameservers set in `/etc/resolv.conf`:

```
$ drill www.archlinux.org
```

You can also specify a specific nameserver's ip address, bypassing the settings in your `/etc/resolv.conf`:

```
$ drill @ip.of.name.server www.archlinux.org
```

For example to test Google's name servers:

```
$ drill @8.8.8.8 www.archlinux.org
```

To test a local name server (such as **unbound**) do:

```
$ drill @127.0.0.1 www.archlinux.org
```

Alternative DNS servers

To use alternative DNS servers, edit `/etc/resolv.conf` and add them at the top of the list so they are used first, optionally removing or commenting out other servers. Currently, you may include a maximum of three nameservers.

Note: Changes made to `/etc/resolv.conf` take effect immediately.

Tip: If you require more flexibility, e.g. more than three nameservers, you can use a local DNS resolver like **dnsmasq** or **unbound**. In this case the nameserver IP address will likely be `127.0.0.1`.

OpenNIC

OpenNIC (http://www.opennicproject.org/) provides free uncensored nameservers located in multiple countries. The full list of public servers is available at **servers.opennic.org (https://servers.opennic.org/)** and a shortlist of nearest nameservers for optimal performance is generated on their **home page (https://www.opennic.org/)**.

To retrieve a list of nearest nameservers, an API is also available and returns, based on the **URL parameters (https://wiki.opennic.org/api/geoip)** provided, a list of nameservers in the desired format. For example to get the 200 nearest IPv4 servers, one can use **https://api.opennicproject.org/geoip/?list&ipv=4&res=200&adm=0&bl&wl**.

Alternatively, the anycast servers below can be used; while reliable their latency **fluctuates a lot (https://wiki.opennic.org/opennic/dont_anycast)**.

```
# OpenNIC IPv4 nameservers (Worldwide Anycast)
nameserver 185.121.177.177
nameserver 185.121.177.53
```

```
# OpenNIC IPv6 nameservers (Worldwide Anycast)
nameserver 2a05:dfc7:5::53
nameserver 2a05:dfc7:5::5353
```

For an automated renewal of your DNS servers with the most responsive OpenNIC servers, the script **nm-opennic (https://github.com/kewlfft/nm-opennic/)** can be used if you have **NetworkManager**.

Note: Use of OpenNIC DNS servers will allow host name resolution in the traditional Top-Level Domain (TLD) registries, but also in OpenNIC or afiliated operated namespaces (.o, .libre, .dyn...)

DNS.WATCH

DNS.WATCH (https://dns.watch/) focuses on neutrality and security and provides two servers located in Germany with no logging and with DNSSEC enabled. Note they welcome commercial sponsorship.

```
# dns.watch IPv4 nameservers
nameserver 84.200.69.80    # resolver1.dns.watch
```

```
nameserver 84.200.70.40    # resolver2.dns.watch
```

UncensoredDNS

UncensoredDNS (http://censurfridns.dk) is a free uncensored DNS service. It is run by a private individual and consists in one anycast served by multiple servers and one unicast node hosted in Denmark.

```
# censurfridns.dk IPv4 nameservers
nameserver 91.239.100.100    ## anycast.censurfridns.dk
nameserver 89.233.43.71      ## unicast.censurfridns.dk
```

```
# censurfridns.dk IPv6 nameservers
nameserver 2001:67c:28a4::   ## anycast.censurfridns.dk
nameserver 2a01:3a0:53:53::  ## unicast.censurfridns.dk
```

Note: Its servers listen to port 5353 as well as the standard port 53. This can be used in case your ISP hijacks port 53.

Cisco Umbrella (formerly OpenDNS)

OpenDNS (https://www.opendns.com/home-internet-security/) provided free alternative nameservers, was **bought by Cisco in Nov. 2016 (https://umbrella.cisco.com/products/features/opendns-cisco-umbrella)** and continues to offer OpenDNS as end-user product of its "Umbrella" product suite with focus on Security Enforcement, Security Intelligence and Web

Filtering. The old nameservers **still work (https://www.opendns.com/setupguide/)** but are **pre-configured to block adult content (https://www.opendns.com/home-internet-security/)**:

```
# OpenDNS IPv4 nameservers
nameserver 208.67.222.222
nameserver 208.67.220.220
```

```
# OpenDNS IPv6 nameservers
nameserver 2620:0:ccc::2
nameserver 2620:0:ccd::2
```

Google

Google's nameservers (https://developers.google.com/speed/public-dns/) can be used as an alternative:

```
# Google IPv4 nameservers
nameserver 8.8.8.8
nameserver 8.8.4.4
```

```
# Google IPv6 nameservers
nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
```

Comodo

Comodo (http://securedns.dnsbycomodo.com/) provides another IPv4 set, with optional (non-free) web-filtering. Implied in this feature is that the service hijacks the queries.

```
# Comodo nameservers
nameserver 8.26.56.26
nameserver 8.20.247.20
```

Yandex

Yandex.DNS (https://dns.yandex.com/advanced/) has servers in Russia, Eastern and Western Europe and has three options, *Basic*, *Safe* and *Family*:

```
# Basic Yandex.DNS - Quick and reliable DNS
nameserver 77.88.8.8              # Preferred IPv4 DNS
nameserver 77.88.8.1              # Alternate IPv4 DNS

nameserver 2a02:6b8::feed:0ff     # Preferred IPv6 DNS
nameserver 2a02:6b8:0:1::feed:0ff # Alternate IPv6 DNS
```

```
# Safe Yandex.DNS - Protection from virus and fraudulent content
nameserver 77.88.8.88             # Preferred IPv4 DNS
nameserver 77.88.8.2              # Alternate IPv4 DNS

nameserver 2a02:6b8::feed:bad     # Preferred IPv6 DNS
nameserver 2a02:6b8:0:1::feed:bad # Alternate IPv6 DNS
```

```
# Family Yandex.DNS - Without adult content
nameserver 77.88.8.7              # Preferred IPv4 DNS
nameserver 77.88.8.3              # Alternate IPv4 DNS

nameserver 2a02:6b8::feed:a11     # Preferred IPv6 DNS
nameserver 2a02:6b8:0:1::feed:a11 # Alternate IPv6 DNS
```

Yandex.DNS' speed is the same in the three modes. In *Basic* mode, there is no traffic filtering. In *Safe* mode, protection from infected and fraudulent sites is provided. *Family* mode enables protection from dangerous sites and blocks sites with adult content.

Quad9

Quad9 (https://quad9.net/#/) is a free DNS service founded by **IBM (https://www.ibm.com/security)**, **Packet Clearing House (https://www.pch.net)** and **Global Cyber Alliance (https://www.globalcyberalliance.org)**; its primary unique feature is a blocklist which avoids resolving known malicious domains. The addresses below are worldwide anycast.

```
# Quad9 IPv4 nameservers
nameserver 9.9.9.9    ## "secure", with blocklist and DNSSEC
nameserver 9.9.9.10    ## no blocklist, no DNSSEC
```

```
# Quad9 IPv6 nameservers
nameserver 2620:fe::fe    ## "secure", with blocklist and DNSSEC
nameserver 2620:fe::10    ## no blocklist, no DNSSEC
```

Preserve DNS settings

dhcpcd, **netctl**, **NetworkManager**, and various other processes can overwrite `/etc/resolv.conf`. This is usually desirable behavior, but sometimes DNS settings need to be set manually (e.g. when using a static IP address). There are several ways to accomplish this.

- If you are using *dhcpcd*, see **#Modify the dhcpcd config** below.
- If you are using **netctl** and static IP address assignment, do not use the `DNS*` options in your profile, otherwise *resolvconf* is called and `/etc/resolv.conf` overwritten.

Prevent NetworkManager modifications

To stop *NetworkManager* from modifying `/etc/resolv.conf`, edit `/etc/NetworkManager/NetworkManager.conf` and add the following in the `[main]` section:

```
dns=none
```

`/etc/resolv.conf` might be a broken symlink that you will need to remove after doing that. Then, just create a new `/etc/resolv.conf` file.

Use openresolv

openresolv (https://www.archlinux.org/packages/?name=openresolv) provides a utility *resolvconf*, which is a framework for managing multiple DNS configurations. See **resolvconf(8) (https://jlk.fjfi.cvut.cz/arch/manpages/man/resolvconf.8)** and **resolvconf.conf(5) (https://jlk.fjfi.cvut.cz/arch/manpages/man/resolvconf.conf.5)** for more information.

The configuration is done in `/etc/resolvconf.conf` and running `resolvconf -u` will generate `/etc/resolv.conf`.

Modify the dhcpcd config

dhcpcd's configuration file may be edited to prevent the *dhcpcd* daemon from overwriting `/etc/resolv.conf`. To do this, add the following to the last section of `/etc/dhcpcd.conf`:

```
nohook resolv.conf
```

Alternatively, you can create a file called `/etc/resolv.conf.head` containing your DNS servers. *dhcpcd* will prepend this file to the beginning of `/etc/resolv.conf`.

Or you can configure dhcpcd to use the same DNS servers every time. To do this, add the following line at the end of your `/etc/dhcpcd.conf`, where *`dns-server-ip-addressses`* is a space separated list of DNS IP addresses.

```
static domain_name_servers=dns-server-ip-addresses
```

For example, to set it to Google's DNS servers:

```
static domain_name_servers=8.8.8.8 8.8.4.4
```

Write-protect /etc/resolv.conf

Another way to protect your `/etc/resolv.conf` from being modified by anything is setting the immutable (write-protection) attribute:

```
# chattr +i /etc/resolv.conf
```

Limit lookup time

If you are confronted with a very long hostname lookup (may it be in **pacman** or while browsing), it often helps to define a small timeout after which an alternative nameserver is used. To do so, put the following in `/etc/resolv.conf`.

```
options timeout:1
```

Tips and tricks

Local domain names

If you want to be able to use the hostname of local machine names without the fully qualified domain names, then add a line to `resolv.conf` with the local domain such as:

```
domain example.com
```

That way you can refer to local hosts such as `mainmachine1.example.com` as simply `mainmachine1` when using the *ssh* command, but the *drill* command still requires the fully qualified domain names in order to perform lookups.

Retrieved from "https://wiki.archlinux.org/index.php?title=Resolv.conf&oldid=500789"