Network Debugging

This article handles the steps needed for basic network troubleshooting.

Related articles

VLAN

Contents

Network Interfaces

The first step in troubleshooting network issues will be to identify which network interfaces are present on the system. See **Network configuration#Get current interface names** for details.

Link status

In the overview of `ip a`, the link status will already be displayed. But it can also be displayed by running:

```
$ ip link show dev eth0
```

This will provide an output along the lines of:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
   link/ether 70:5a:b6:8a:a0:87 brd ff:ff:ff:ff:ff:ff
```

Bringing up an interface can be done by issuing:

```
# ip link set dev eth0 up
```

RTNETLINK answers: Cannot assign requested address

If you get this error when trying to set an interface up, its most probably because you've got an invalid MAC address. To set a working MAC, see **MAC address spoofing**.

IP address

In the overview provided by `ip a`, the ip address will already be displayed. But it can also be displayed by running:

```
$ ip addr show dev eth0
```

This will provide an output along the lines of:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
  link/ether 70:5a:b6:8a:a0:87 brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.143/24 brd 192.168.1.255 scope global eth0
  inet6 fe80::725a:b6ff:fe8a:a087/64 scope link
     valid_lft forever preferred_lft forever
```

Adding a temporary ip address:

```
# ip addr add 192.168.1.143/24 dev eth0
```

Removing an ip address:

```
# ip addr del 192.168.1.143/24 dev eth0
```

Route table

The route table can be displayed by running:

```
$ ip route show
```

Route table for a specific interface:

```
$ ip route show dev eth0
```

This will provide an output along the lines of:

```
default via 192.168.1.1  proto static
192.168.1.0/24  proto kernel  scope link  src 192.168.1.143
```

Configuring the default gateway:

```
# ip route add 0/0 via 192.168.1.1 dev eth0
```

Removing the default gateway:

```
# ip route del 0/0 via 192.168.1.1 dev eth0
```

DNS Servers

Dns is responsible for converting hostnames to an ip address. When connectivity towards ip addresses is working, but the system is unable to connect to a hostname; there is a fair chance that this will be related to the dns configuration. The configuration can be displayed by running:

```
$ cat /etc/resolv.conf
```

This will provide an output among the lines of:

```
domain example.com
search example.com
nameserver 192.168.1.1
```

- The rule '`nameserver`' is the relevant section. Configuring multiple nameservers is supported.
- The '`domain`' and '`search`' rules are optional.
- Often the '`nameserver`' is the same as your default gateway.
- In case of doubt there is always the possibility to use the Google DNS servers as your default DNS servers:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Testing your dns configuration can be done through the `drill` command (from the **ldns (https://www.archlinux.org/packages/?name=ldns)** package):

```
$ drill www.archlinux.org @8.8.4.4
```

The above command will perform a dns lookup of www.archlinux.org using the 8.8.4.4 dns server and return output as follows:

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 35144
;; flags: qr rd ra ; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; www.archlinux.org.   IN      A

;; ANSWER SECTION:
www.archlinux.org.      966     IN      CNAME   gudrun.archlinux.org.
gudrun.archlinux.org.   18966   IN      A       66.211.214.131

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:

;; Query time: 38 msec
;; SERVER: 8.8.4.4
;; WHEN: Wed Jun 17 20:21:47 2015
;; MSG SIZE  rcvd: 72
```

As an alternative to *drill*, there are also the `dig`, `host`, and `nslookup` tools from the **bind-tools (https://www.archlinux.org/packages/?name=bind-tools)** package.

Ping & Tracepath/Traceroute

The ping command can help test connectivity towards a specific host.

The first step would be verifying connectivity towards the default gateway (replace the ip address with your own default gateway):

```
$ ping -c4 192.168.1.1
```

When erasing the "-c4" parameter, the ping will continue endlessly. It can be aborted by hitting "Control-C".

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=0.193 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=0.190 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=64 time=0.192 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=64 time=0.189 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.165/0.184/0.193/0.014 ms
```

The output above indicated the default gateway is reachable. When instead a "`Destination Host Unreachable`" message is displayed, doublecheck the ip address, netmask and default gateway config. This message can also be displayed when ICMP traffic is not permitted towards the default gateway (blocked by a firewall, router,...).

The next step is verifying connectivity towards the configured dns server(s). When no reply is received, `tracepath` or `traceroute` can be used to verify the routing towards said server and get an idea of where the issue lies.

```
$ traceroute 8.8.4.4
```

Traceroute also used ICMP to determine the path and hence there can be "no reply" answers as well when ICMP traffic is blocked.

Retrieved from "https://wiki.archlinux.org/index.php?title=Network_Debugging&oldid=507389"
