Network bridge

1 language

- 日本語

Related articles

Bridge with netctl

Network configuration#Bonding or LAG

A bridge is a piece of software used to unite two or more network segments. A bridge behaves like a virtual network switch, working transparently (the other machines do not need to know about its existence). Any real devices (e.g. `eth0`) and virtual devices (e.g. `tap0`) can be connected to it.

This article explains how to create a bridge that contains at least an ethernet device. This is useful for things like the bridge mode of QEMU, setting a software based access point, etc.

1 Creating a bridge

There are a number of ways to create a bridge.

1.1 With iproute2

This section describes the management of a network bridge using the *ip* tool from the `iproute2 (https://archlinux.org/packages/?name=iproute2)` package, which is required by the `base (https://archlinux.org/packages/?name=base)` meta package.

Create a new bridge and change its state to up:

```
# ip link add name bridge_name type bridge
# ip link set dev bridge_name up
```

To add an interface (e.g. eth0) into the bridge, its state must be up:

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

Adding the interface into the bridge is done by setting its master to *`bridge_name`*:

```
# ip link set eth0 master bridge_name
```

To show the existing bridges and associated interfaces, use the *bridge* utility (also part of `iproute2 (https://archlinux.org/packages/?name=iproute2)`). See `bridge(8) (https://man.archlinux.org/man/bridge.8)` for details.

```
# bridge link
```

This is how to remove an interface from a bridge:

```
# ip link set eth0 nomaster
```

The interface will still be up, so you may also want to bring it down:

```
# ip link set eth0 down
```

To delete a bridge issue the following command:

```
# ip link delete bridge_name type bridge
```

This will automatically remove all interfaces from the bridge. The slave interfaces will still be up, though, so you may also want to bring them down after.

1.2 With bridge-utils

This section describes the management of a network bridge using the legacy *brctl* tool from the **bridge-utils (https://archlinux.org/packages/?name=bridge-utils)** package. See **brctl(8) (https://man.archlinux.org/man/brctl.8)** for full listing of options.

Note: The use of *brctl* is deprecated and is considered obsolete. See the Notes section in **brctl(8) § NOTES (https://man.archlinux.org/man/brctl.8#NOTES)** for details.

Create a new bridge:

```
# brctl addbr bridge_name
```

Add a device to a bridge, for example `eth0` :

Note: Adding an interface to a bridge will cause the interface to lose its existing IP address. If you are connected remotely via the interface you intend to add to the bridge, you will lose your connection. This problem can be worked around by scripting the bridge to be created at system startup.

```
# brctl addif bridge_name eth0
```

Show current bridges and what interfaces they are connected to:

```
$ brctl show
```

Set the bridge device up:

```
# ip link set dev bridge_name up
```

Delete a bridge, you need to first set it to *down*:

```
# ip link set dev bridge_name down
# brctl delbr bridge_name
```

Note: To enable the **bridge-netfilter (http://ebtables.netfilter.org/documentation/bridge-nf.html)** functionality, you need to manually load the `br_netfilter` module:

```
# modprobe br_netfilter
```

You can also **load the module at boot**.

1.3 With netctl

See **Bridge with netctl**.

1.4 With systemd-networkd

See **systemd-networkd#Bridge interface**.

1.5 With NetworkManager

GNOME's Network settings can create bridges, but currently will not auto-connect to them or slave/attached interfaces. Open Network Settings, add a new interface of type Bridge, add a new bridged connection, and select the MAC address of the device to attach to the bridge.

KDE's **`plasma-nm` (https://archlinux.org/packages/?name=plasma-nm)** can create bridges. In order to view, create and modify bridge interfaces open the Connections window either by right clicking the Networks applet in the system tray and selecting *Configure Network Connections...* or from *System Settings > Connections*. Click the *Configuration* button in the lower left corner of the module and enable "Show virtual connections". A session restart will be necessary to use the enabled functionality.

`nm-connection-editor` (https://archlinux.org/packages/?name=nm-connection-editor) can create bridges in the same manner as GNOME's Network settings. **This (https://www.xmodulo.com/configure-linux-bridge-network-manager-ubuntu.html)** page shows these steps with screenshots.

`nmcli` from **`networkmanager` (https://archlinux.org/packages/?name=networkmanager)** can create bridges. Creating a bridge with **STP** disabled (to avoid the bridge being advertised on the network):

```
$ nmcli connection add type bridge ifname br0 stp no
```

Making interface `enp30s0` a slave to the bridge:

```
$ nmcli connection add type bridge-slave ifname enp30s0 master br0
```

Setting the existing connection as down (you can get it with `nmcli connection show --active`):

```
$ nmcli connection down Connection
```

Setting the new bridge as up:

```
$ nmcli connection up bridge-br0
$ nmcli connection up bridge-slave-enp30s0
```

If NetworkManager's default interface for the device you added to the bridge connects automatically, you may want to disable that by clicking the gear next to it in Network Settings, and unchecking "Connect automatically" under "Identity."

2 Assigning an IP address

When the bridge is fully set up, it can be assigned an IP address:

2.1 With iproute2

```
# ip address add dev bridge_name 192.168.66.66/24
```

2.2 With NetworkManager

Give it the desired address:

```
# nmcli connection modify Connection ipv4.addresses desired_IP
```

Set up a DNS server (this will also avoid not being able to load any pages after you apply the changes):

```
# nmcli connection modify Connection ipv4.dns DNS_server
```

Set the IP address to static:

```
# nmcli connection modify Connection ipv4.method manual
```

Apply the changes:

```
# nmcli connection up Connection
```

3 Tips and tricks

3.1 Wireless interface on a bridge

To add a wireless interface to a bridge, you first have to assign the wireless interface to an access point or start an access point with **hostapd**. Otherwise the wireless interface will not be added to the bridge.

See also **Debian:BridgeNetworkConnections#Bridging with a wireless NIC**.

3.2 Speeding up traffic destinated to the bridge itself

In some situations the bridge not only serves as a bridge box, but also talks to other hosts. Packets that arrive on a bridge port and that are destinated to the bridge box itself will by default enter the iptables INPUT chain with the logical bridge port as input device. These packets will be queued twice by the network code, the first time they are queued after they are received by the network device. The second time after the bridge code examined the destination MAC address and determined it was a locally destinated packet and therefore decided to pass the frame up to the higher protocol stack.**[1] (http://ebtables.netfilter.org/examples/basic.html#ex_speed)**

The way to let locally destinated packets be queued only once is by brouting them in the BROUTING chain of the broute table. Suppose br0 has an IP address and that br0's bridge ports do not have an IP address. Using the following rule should make all locally directed traffic be queued only once:

```
# ebtables -t broute -A BROUTING -d $MAC_OF_BR0 -p ipv4 -j redirect --redirect-
target DROP
```

The replies from the bridge will be sent out through the br0 device (assuming your routing table is correct and sends all traffic through br0), so everything keeps working neatly, without the performance loss caused by the packet being queued twice.

The redirect target is needed because the MAC address of the bridge port is not necessarily equal to the MAC address of the bridge device. The packets destinated to the bridge box will have a destination MAC address equal to that of the bridge br0, so that destination address must be changed to that of the bridge port.

4 Troubleshooting

4.1 No networking after bridge configuration

It may help to remove all IP addresses and routes from the interface (e.g. `eth0`) that was added to the bridge and configure these parameters for the bridge instead.

First of all, make sure there is no **dhcpcd** instance running for `eth0` , otherwise the deleted addresses may be reassigned.

Remove address and route from the `eth0` interface:

```
# ip addr del address dev eth0
# ip route del address dev eth0
```

Now IP address and route for the earlier configured bridge must be set. This is usually done by starting a DHCP client for this interface. Otherwise, consult **Network configuration** for manual configuration.

4.2 No networking on hosted servers after bridge configuration

As the MAC address of the bridge is not necessarily equal to the MAC address of the networking card usually used by the server, the server provider might drop traffic coming out from the bridge, resulting in a loss of connectivity when bridging e.g. the server ethernet interface. Configuring the bridge to clone the mac address of the ethernet interface might therefore be needed for hosted servers.

4.3 Cannot connect to bridge connection after connecting to usual connection

In Network Manager applet, if you have usual ethernet/wireless connection (not a bridge slave connection), and if you first connect to it, and then try to connect to bridged connection (with or without disconnecting from usual connection first), then you are not able to connect to it. For some reason, the bridge slave connection (it is not listed in network applet) is not activated, even when the auto connect checkbox is enabled.

The workaround is to activate it manually via terminal:

```
nmcli connection up br1\ slave\ 1
```

Then immediately your bridge connections works.

5 See also

- **Official documentation for bridge-utils (https://www.linuxfoundation.org/collaborate/workgroups/networking/bridge)**[**dead link** 2022-09-22]
- **Official documentation for iproute2 (https://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2)**
- **ebtables/iptables interaction on a Linux-based bridge (http://ebtables.netfilter.org/br_fw_ia/br_fw_ia.html)**

Retrieved from "**https://wiki.archlinux.org/index.php?title=Network_bridge&oldid=758008**"