

# IPVLAN Driver HOWTO

Initial Release:

Maresh Bandewar <mareshb AT google.com>

## 1. Introduction:

This is conceptually very similar to the macvlan driver with one major exception of using L3 for mux-ing /demux-ing among slaves. This property makes the master device share the L2 with it's slave devices. I have developed this driver in conjunction with network namespaces and not sure if there is use case outside of it.

## 2. Building and Installation:

In order to build the driver, please select the config item CONFIG\_IPVLAN. The driver can be built into the kernel (CONFIG\_IPVLAN=y) or as a module (CONFIG\_IPVLAN=m).

## 3. Configuration:

There are no module parameters for this driver and it can be configured using IProute2/ip utility.

```
ip link add link <master> name <slave> type ipvlan [ mode MODE ] [ FLAGS ]
where
  MODE: l3 (default) | l3s | l2
  FLAGS: bridge (default) | private | vepa
```

e.g.

- a. Following will create IPvlan link with eth0 as master in L3 bridge mode:

```
bash# ip link add link eth0 name ipv10 type ipvlan
```

- b. This command will create IPvlan link in L2 bridge mode:

```
bash# ip link add link eth0 name ipv10 type ipvlan mode l2 bridge
```

- c. This command will create an IPvlan device in L2 private mode:

```
bash# ip link add link eth0 name ipvlan type ipvlan mode l2 private
```

- d. This command will create an IPvlan device in L2 vepa mode:

```
bash# ip link add link eth0 name ipvlan type ipvlan mode l2 vepa
```

## 4. Operating modes:

IPvlan has two modes of operation - L2 and L3. For a given master device, you can select one of these two modes and all slaves on that master will operate in the same (selected) mode. The RX mode is almost identical except that in L3 mode the slaves wont receive any multicast / broadcast traffic. L3 mode is more restrictive since routing is controlled from the other (mostly) default namespace.

### 4.1 L2 mode:

In this mode TX processing happens on the stack instance attached to the slave device and packets are switched and queued to the master device to send out. In this mode the slaves will RX/TX multicast and broadcast (if applicable) as well.

### 4.2 L3 mode:

In this mode TX processing up to L3 happens on the stack instance attached to the slave device and packets are switched to the stack instance of the master device for the L2 processing and routing from that instance will be used before packets are queued on the outbound device. In this mode the slaves will not receive nor can send multicast / broadcast traffic.

### 4.3 L3S mode:

This is very similar to the L3 mode except that iptables (conn-tracking) works in this mode and hence it is L3-symmetric (L3s). This will have slightly less performance but that shouldn't matter since you are choosing this mode over plain-L3 mode to make conn-tracking work.

## 5. Mode flags:

At this time following mode flags are available

## 5.1 bridge:

This is the default option. To configure the IPvlan port in this mode, user can choose to either add this option on the command-line or don't specify anything. This is the traditional mode where slaves can cross-talk among themselves apart from talking through the master device.

## 5.2 private:

If this option is added to the command-line, the port is set in private mode. i.e. port won't allow cross communication between slaves.

### 5.3 vepa:

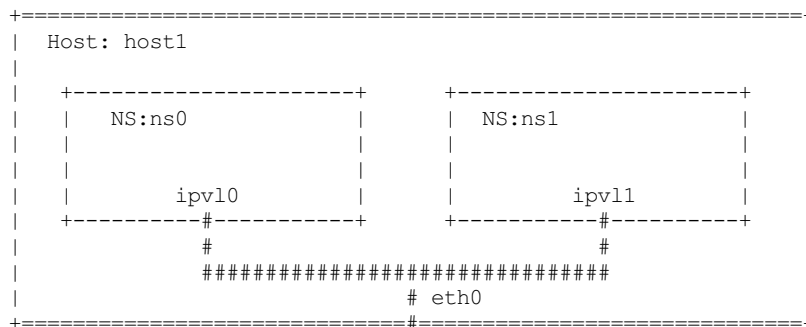
If this is added to the command-line, the port is set in VEPA mode. i.e. port will offload switching functionality to the external entity as described in 802.1Qbg Note: VEPA mode in IPvlan has limitations. IPvlan uses the mac-address of the master-device, so the packets which are emitted in this mode for the adjacent neighbor will have source and destination mac same. This will make the switch / router send the redirect message.

## 6. What to choose (macvlan vs. ipvlan)?

These two devices are very similar in many regards and the specific use case could very well define which device to choose. if one of the following situations defines your use case then you can choose to use `ipvlan`:

- The Linux host that is connected to the external switch / router has policy configured that allows only one mac per port.
- No of virtual devices created on a master exceed the mac capacity and puts the NIC in promiscuous mode and degraded performance is a concern.
- If the slave device is to be put into the hostile / untrusted network namespace where L2 on the slave could be changed / misused.

## 6. Example configuration:



- a. Create two network namespaces - ns0, ns1:

```
ip netns add ns0
ip netns add ns1
```

- b. Create two `ipvlan` slaves on `eth0` (master device):

```
ip link add link eth0 ipv10 type ipvlan mode 12
ip link add link eth0 ipv11 type ipvlan mode 12
```

- c. Assign slaves to the respective network namespaces:

```
ip link set dev ipv10 netns ns0
ip link set dev ipv11 netns ns1
```


- d. Now switch to the namespace (ns0 or ns1) to configure the slave devices

- For ns0:

```
(1) ip netns exec ns0 bash
(2) ip link set dev ipv10 up
(3) ip link set dev lo up
(4) ip -4 addr add 127.0.0.1 dev lo
(5) ip -4 addr add $IPADDR dev ipv10
(6) ip -4 route add default via $ROUTER dev ipv10
```

- For ns1:

```
(1) ip netns exec ns1 bash
(2) ip link set dev v1 up
```



```
(3) ip link set dev lo up
(4) ip -4 addr add 127.0.0.1 dev lo
(5) ip -4 addr add $IPADDR dev ipv11
(6) ip -4 route add default via $ROUTER dev ipv11
```

