netdevsim devlink support

This document describes the `devlink` features supported by the `netdevsim` device driver.

Parameters

Generic parameters
implemented

Name	Mode
`max_macs`	driverinit

The `netdevsim` driver also implements the following driver-specific parameters.

Driver-specific parameters implemented

Name	Type	Mode	Description
`test1`	Boolean	driverinit	Test parameter used to show how a driver-specific devlink parameter can be implemented.

The `netdevsim` driver supports reloading via `DEVLINK_CMD_RELOAD`

Regions

The `netdevsim` driver exposes a `dummy` region as an example of how the devlink-region interfaces work. A snapshot is taken whenever the `take_snapshot` debugfs file is written to.

Resources

The `netdevsim` driver exposes resources to control the number of FIB entries, FIB rule entries and nexthops that the driver will allow.

```
$ devlink resource set netdevsim/netdevsim0 path /IPv4/fib size 96
$ devlink resource set netdevsim/netdevsim0 path /IPv4/fib-rules size 16
$ devlink resource set netdevsim/netdevsim0 path /IPv6/fib size 64
$ devlink resource set netdevsim/netdevsim0 path /IPv6/fib-rules size 16
$ devlink resource set netdevsim/netdevsim0 path /nexthops size 16
$ devlink dev reload netdevsim/netdevsim0
```

Rate objects

The `netdevsim` driver supports rate objects management, which includes:

- registerging/unregistering leaf rate objects per VF devlink port;
- creation/deletion node rate objects;
- setting tx_share and tx_max rate values for any rate object type;
- setting parent node for any rate object type.

Rate nodes and it's parameters are exposed in `netdevsim` debugfs in RO mode. For example created rate node with name `some_group`:

```
$ ls /sys/kernel/debug/netdevsim/netdevsim0/rate_groups/some_group
rate_parent  tx_max  tx_share
```

Same parameters are exposed for leaf objects in corresponding ports directories. For ex.:

```
$ ls /sys/kernel/debug/netdevsim/netdevsim0/ports/1
dev  ethtool  rate_parent  tx_max  tx_share
```

Driver-specific Traps

List of Driver-specific Traps Registered by `netdevsim`

Name	Type	Description
`fid_miss`	`exception`	When a packet enters the device it is classified to a filtering indentifier (FID) based on the ingress port and VLAN. This trap is used to trap packets for which a FID could not be found