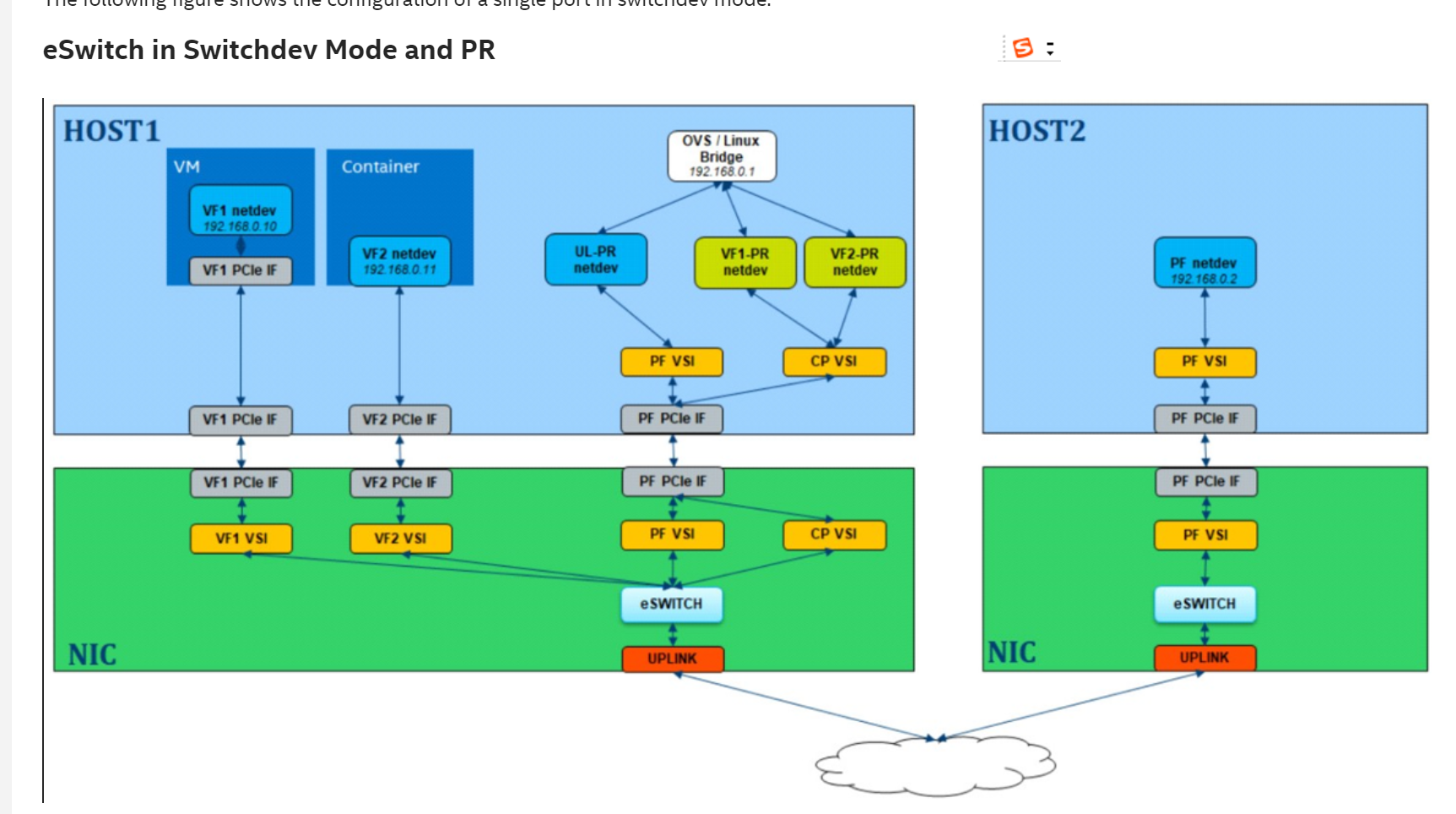
# [Intel® Ethernet 800](https://www.intel.com/content/www/us/en/products/details/ethernet/800-network-adapters.html)



In switchdev mode, the 800 Series Controller’s PF driver supports a standard Linux kernel abstraction layer (switchdev API) to expose the control plane of the Ethernet controller's eSwitch to the software vSwitch (OVS/Linux Bridge). This switchdev API was originally developed in the Linux Kernels to configure switch hardware ASICs. The switchdev API enables limited vSwitch functionality offload to the Ethernet controller's eSwitch to allow hardware switch assist for OVS/Linux Bridge.

As detailed in [Technical Details: eSwitch Legacy Mode](https://edc.intel.com/content/www/us/en/design/products/ethernet/appnote-e810-eswitch-switchdev-mode-config-guide/technical-details-eswitch-legacy-mode/" \l "GUID-F96F0887-126F-4C9A-8B79-F85A393B96F9), in Legacy mode (without switchdev), only MAC/VLAN are configured and used in VEB (eSwitch) for hardware switching. Based on policies configured by the host admin or SDN controller, flows are configured in the software switch (vSwitch) through a *slow path* mechanism. Flows, in this case, refer to (filter-match)/action tables for different filter/classifier (L2/L3 and L4 fields). To make use of hardware switching capabilities for vSwitch, it is required for the eSwitch to have the same flows as the vSwitch.

When the PF's eSwitch mode is set as *switchdev*, Port Representor netdevs are created for the PF and each VF associated to that eSwitch. The PF netdev is replaced by an UpLink Port Representor (UL\_PR) netdev and a VF Port Representor (VF\_PR) netdev is created for each VF.

PRs are similar to hardware accelerated networking ports and function like standard Linux network interfaces. The PF\_VSI backs up the PF (UL\_PR) netdev and a new Control Plane VSI (CP\_VSI) is created; and, it backs up all the VF\_PR netdevs. UL\_PR can be used for two-way communication ports with VFs. VF\_PRs are used for control plane communications through the OVS control plane.

These PRs enable exposing statistics as well as configuring and monitoring link state, MTU, filters, FDB/VLAN entries, and so on. These PRs plug into existing kernel software switching subsystems (such as TC and OVS) and allow offload of software traffic rules (flows) to the Ethernet controller (hardware).

To offload a data forwarding path (software vSwitch flows) from the kernel to Ethernet controller hardware (eSwitch), the bridge Forwarding Database (FDB) entries are mirrored down to the Ethernet controller. By default, the hardware FDB has entries for {MAC, VLAN, PORT} tuples.

With switchdev mode, the 800 Series Ethernet Network Controller PF driver also supports the following switch rule programming that can be used for hardware switching.

For example, when a route is added in OVS, a function calls the Intel *switchdev* driver, which then determines whether this route needs to be offloaded to the hardware. Routes that do not involve the eSwitch are typically not offloaded.

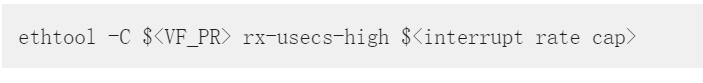
The following provides a summary on Port Representor capabilities:

* Netdevs for all switching ports created on an eSwitch in switchdev mode:
  + Uplink ports (UL\_PR)
  + Virtual ports (VF\_PR)
* PR reflects the originating item's link state.
* PR should not be used to control Receive Side Scaling (RSS) input set or Intel® Ethernet Flow Director (Intel® Ethernet FD) settings for the VF.
* PR supports default/exception paths.
* When packets are sent from the PR netdev, they are sourced routed directly to the port.
* Scope is limited to single Uplink port in one switching domain/switchdev instance.

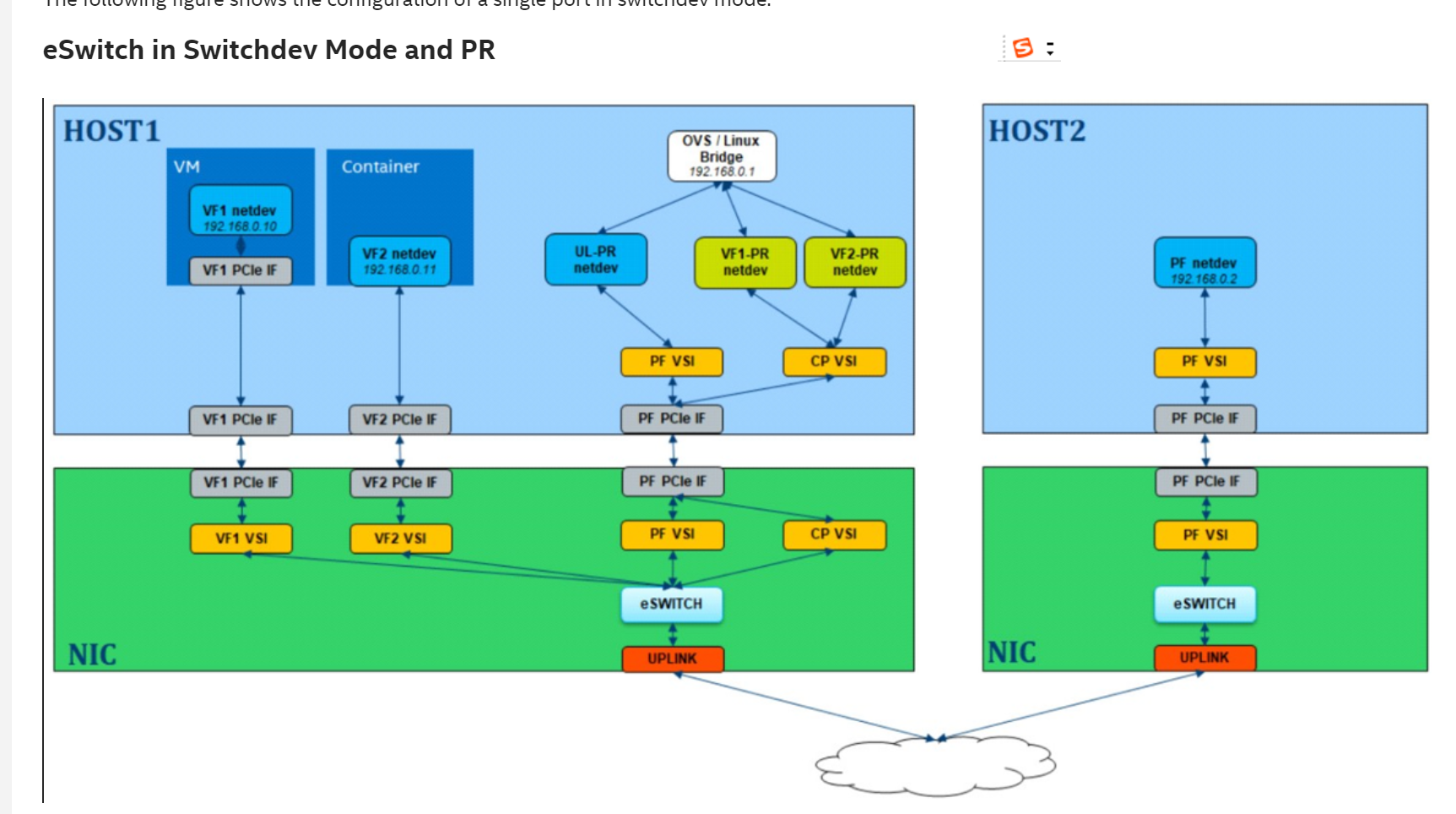
****VF Port Representor:****

Each VF must have a unique Port Representor. Intel recommends that you use the corresponding VF\_PRs for any VF-related configuration.

For example, if you want to limit a VF's interrupt rate for Rx and Tx (*Bounding interrupt rates using rx-usecs-high*), you apply the command using the VF\_PR, as shown here:



## Default/Exception/Slow Path



In switchdev mode, the flow rules are programmed by the control plane. An exception path is enabled through CP\_VSI to allow a virtual switch like OVS or Linux bridge to receive any packet that does not match any hardware filter and program the flow rules based on the policy configured by the host admin or SDN controller. The packet is also reinjected to the right port. This path is known as the slow path.

Initially by default, the hardware eSwitch has only MAC/VLAN entries. Other configurations and filters are not present on the hardware eSwitch. For every first packet received on the external LAN port, there is no matching flow rule in the hardware eSwitch, so the first packet always follows the Default or exception path through CP\_VSI to reach the software vSwitch (OVS or Linux bridge).

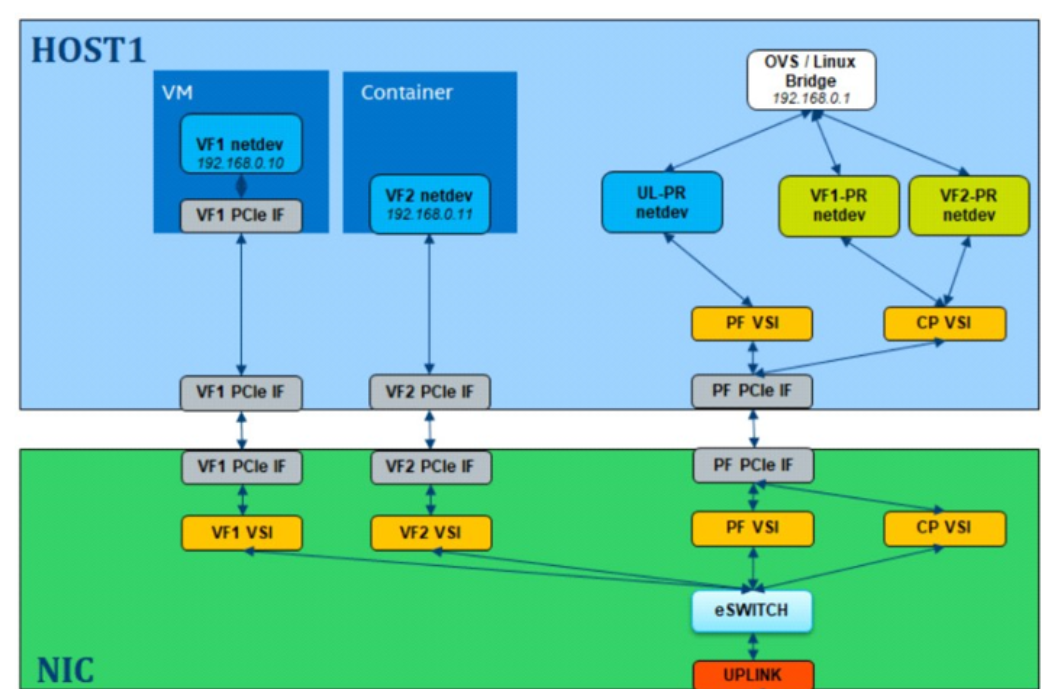
The vSwitch has flow rules configured as per host admin or SDN controller. When a matching flow entry is hit in the software switch control plane (OVS/Linux Bridge), a predefined match action is performed. At the same time, that match/action flow rule is also configured in the hardware eSwitch (classifier engine). This way, a mirror of software control plane/FDB is created in hardware eSwitch.

When the next packet hits the LAN Port, there is a matching entry in the hardware eSwitch's control plane/FDB, and hardware performs the required match action. This enables hardware switching for filter/flows configured in software, and is known as the fast path. From this packet onwards, all similar packets go through this fast path (or vSwitch acceleration path).

To support an exception path, CP\_VSI is configured as the default VSI for the eSwitch for all packets received from the VFs. Packets from uplink are directed to the PF\_VSI. Packets received on CP\_VSI are directed to the corresponding VF\_PR netdev based on the Source VSI in the RX descriptor. The PF\_VSI is configured as the default VSI for uplink packets, and the frames received on PF\_VSI are directed to UL\_PR netdev.

Transmits from PR netdevs are treated as directed transmits. Transmits from UL\_PR are directed to the network by setting the switch control tag to indicate uplink packet and bypass any hardware filters.

### 默认配置

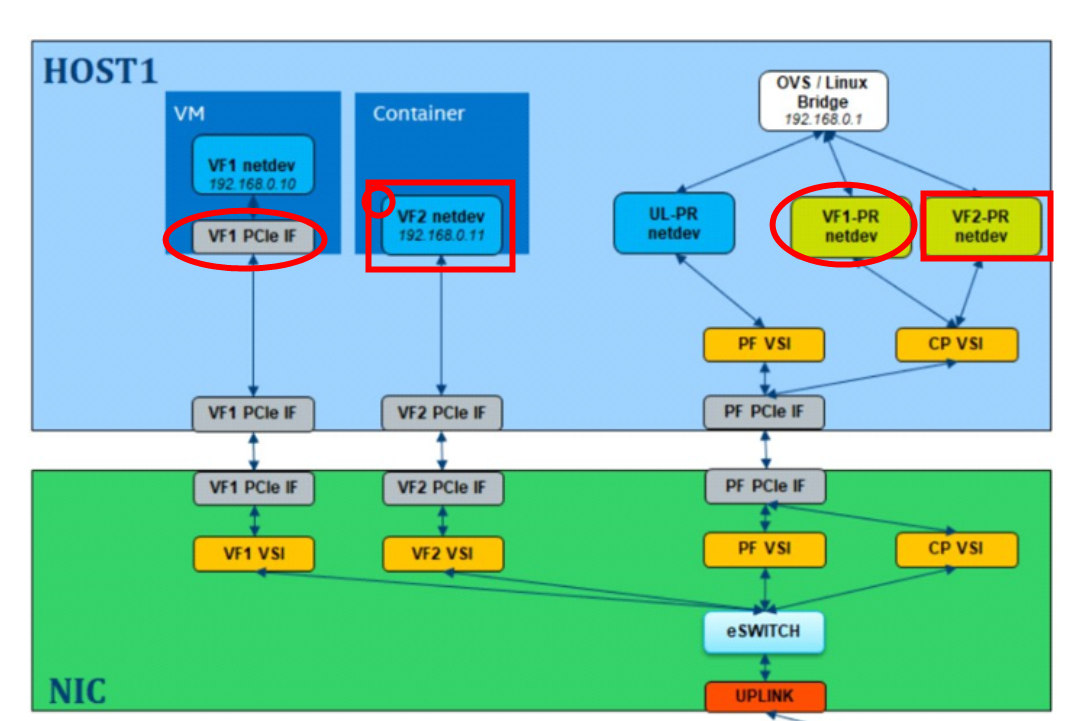


* VFs发出的报文--> CP\_VSI --> VF\_PRX(根据vsi id)
* uplink接收的报文 --> PF\_VSI --> UL\_PR
* UL\_PR 发出的报文--> network

### 举例

* ****Exception path for VM-to-VM packets:****

**VF1netdev(左边椭圆)和左边VF2 netdev（方块）通信**



VF1\_netdev → VF1\_VSI → eSwitch → CP\_VSI → VF1\_PR netdev → OVS/Linux Bridge → VF2\_PR netdev → CP\_VSI → eSwitch → VF2\_VSI → VF2\_netdev

* ****Exception path for uplink to VM packets:****

PF\_netdev → PF\_VSI(H2) → eSwitch(H2) → Uplink(H2) → Uplink(H1) → eSwitch(H1) → PF\_VSI(H1) → UL\_PR netdev → OVS/Linux Bridge → VF1\_PR netdev → CP\_VSI → eSwitch(H1) → VF1\_VSI → VF1 netdev

* ****Exception path for VM to uplink packets:****

VF1 netdev → VF1 VSI → eSwitch → CP VSI → UL\_PR netdev → OVS/Linux Bridge → UL\_PR netdev → PF VSI → eSwitch(H1) → Uplink(H1) → Uplink(H2) → eSwitch(H2) → PF VSI → PF netdev

# Switchdev

Port Representator netdevs are created for each PF and VF if the switch mode is set to 'switchdev'. These netdevs can be used to control and configure VFs and PFs when they are moved to a different namespace.They enable exposing statistics, configure and monitor link state, mtu, filters,fdb/vlan entries etc.

Sample script to create port representors

1. # rmmod i40e; modprobe i40e
2. # devlink dev eswitch set pci/0000:42:00.0 mode switchdev
3. # echo 2 > /sys/class/net/p4p1/device/sriov\_numvfs
4. # ip l show

122: p4p1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

link/ether 3c:fd:fe:a3:18:f8 brd ff:ff:ff:ff:ff:ff

vf 0 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off

vf 1 MAC 00:00:00:00:00:00, spoof checking on, link-state auto, trust off

124: p4p1-pf: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

link/ether 72:8e:34:b2:d0:44 brd ff:ff:ff:ff:ff:ff

125: p4p1-vf0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

link/ether 02:57:a0:18:2b:ce brd ff:ff:ff:ff:ff:ff

126: p4p1-vf1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

link/ether 32:7c:77:5f:3e:e3 brd ff:ff:ff:ff:ff:ff

127: p4p1\_0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

link/ether 26:51:28:54:69:43 brd ff:ff:ff:ff:ff:ff

128: p4p1\_1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000

p4p1 is the PF. p4p1-pf is the port netdev for PF.

p4p1\_0, p4p1\_1 are VFs and p4p1-vf0, p4p1-vf1 are the port netdev's for the 2 VFs.

# 参考

|  |  |  |
| --- | --- | --- |
|  | **[eSwitch Switchdev Mode](https://edc.intel.com/content/www/us/en/design/products/ethernet/appnote-e810-eswitch-switchdev-mode-config-guide/technical-details-eswitch-switchdev-mode/" \l "GUID-02744BEC-8444-41B6-A955-A832EC8F4153)** |  |
|  | [Default/Exception/Slow Path](https://edc.intel.com/content/www/de/de/design/products/ethernet/appnote-e810-eswitch-switchdev-mode-config-guide/default-exception-slow-path/) |  |
|  | [i40e: Introduce Port Representor netdevs and switchdev mode](https://patchwork.ozlabs.org/project/intel-wired-lan/patch/1490833375-2788-3-git-send-email-sridhar.samudrala@intel.com/) |  |