Содержание

18 Bypass network card questions

Why do we recommend to use SILICOM network cards?
Here are the reasons:

- these cards support bypass functionality;
- drivers' licenses for DNA & Libzero can be included into delivery package. These drivers allow to get the highest productivity. The cards with included drivers are marked by -SQ1 (for 10Gb card).

Does bypass work in SILICOM cards on power off?

- Fiber optics bypass does work on power off. It was verified on card PE210G2BPI9-SR-SQ1 short range/fiber;
- The copper bypass does work with no power. It was verified on card PEG6BPi6).

Do SILICOM cards have manual bypass control?
Basically DPI controls bypass by itself.
bpctl_util utility provides manual bypass control, if required:

- bpctl_util all get_bypass - get bypass status;
- bpctl_util all set_bypass on - activate bypass;
- bpctl_util all set_bypass off - disactivate bypass.

We have got second-hand card. bypass does not work. What can we do?
The problem is caused by configuring this card as a standard. It means bypass functionality is off.
To diagnose:

```
bpctl_util all get_std_nic
07:00.0 dna0 standard
07:00.1 dna1 slave
07:00.2 dna2 standard
07:00.3 dna3 slave
```

The correct result is: non-standard.

To set the card in bypass mode, one should execute the following instructions:

```
bpctl_util all set_std_nic off
```

This instruction switches the card into non-standard mode: it means, with bypass mode.

Explanation of bypass switching time
The bypass is activated in a short time interval of about 0.5 seconds (by default), but for reasons of interfaces overlapping it may take a longer time interval. Below is the manufacturer's explanation of bypass switching.
Such duration of switching can affect BGP, OSPF and other mechanisms, due to a brief disconnection (duration may vary, see description below) or several disconnections as in case of server or service restart (*service→break→bypass→break→service*). In this case the session recovery time (BGP, OSPF) depends on their settings and may last up to several tens of seconds. To reduce this interval you need

to configure yourself to reduce the session recovery time after disconnection. **For example**, on Juniper equipment hold-timer down 500ms is configured to avoid BGP session breakup and routing tables rebuilding:

```
set interfaces <ifname> hold-time up 500 down 500
```

where 500 ms is the timeout before the interface's operational status changes

```
Basically, the time for the bypass mechanism to switch from one mode to
another is 10mS.
The timing that you are seeing relates to re-establish the link and then re-
establish
the connection (with new routing tables in switches and devices).
This switch to bypass mode is done in our product by physically connecting
the pair of
the ports together (wire to wire). This means that when this happen our
product is actually out
of the picture and the start of the traffic with this new connection will
depend on
the two networking devices (router / switch / device) on how they link
together and how
they establish the connection again. You can try to  force fix mode (not
auto-neg,
change to force 1G FD or so) this might reduce the time needed for the
negotiation.
Not sure how much.
For the change from bypass mode to normal mode - all the above also stand as
well.
The networking devices (router / switch / device) loss the link with each
other and
starts establish the connection with the Silicom NIC . Here you have more
control as
the link is done between the two devices and your system (Check that all the
devices
are set to the same speed settings)
From our customer and our experience a 1-3sec is a reasonable time to get
the Copper 1G link
to be establish between 2 network devices.
```

However, one port on the NIC is in bypass mode and does not filter traffic.
If it is configured (in/out_dev), but does not switch, try resetting the bypass switch in the card to its initial state:

```
bpctl_util all set_bypass off
bpctl_util all set_dis_bypass off
bpctl_util all set_bypass_pwoff on
bpctl_util all set_bypass_pwup on
bpctl_util all set_std_nic off
bpctl_util all get_bypass_change on
bpctl_util all get_tx on
bpctl_util all get_tpl off
```

```
bpctl_util all get_wait_at_pwup off
bpctl_util all get_hw_reset off
bpctl_util all get_disc off
bpctl_util all get_disc_change off
bpctl_util all get_dis_disc off
bpctl_util all get_disc_pwup off
bpctl_util all get_wd_exp_mode bypass
bpctl_util all get_wd_autoreset disable
```

If it doesn't work, it means the card is defective, replace it under warranty.

Setting up Juniper so that switching to and from bypass does not cause routes to be re-routed.

```
set interfaces <ifname> hold-time up 500 down 500

show xe-5/2/0
 description "-= 20G UPLINK LAGG =-";
hold-time up 1000 down 1000;
gigether-options {
 802.3ad ae1;
}
```

Setting up Cisco so that switching to and from bypass does not cause routes to be re-routed.

```
int fa0/0
ip bgp fast-external-fallover deny
```

Note:

```
BGP Fast-external-fallover command terminates external BGP sessions of any
directly adjacent peer if the link used to reach the peer goes down; without
waiting for the hold-down timer to expire
```

List of all dna interfaces and their MAC addresses
grep ^ /sys/class/net/dna?/address

How to check if the card can perform bypass
You can check for bypass by running the command:

```
lspci -v|grep -A1 Eth
```

For cards with bypass, the Subsystem field will indicate:

```
Subsystem: Silicom Ltd. Device
```