Table of Contents

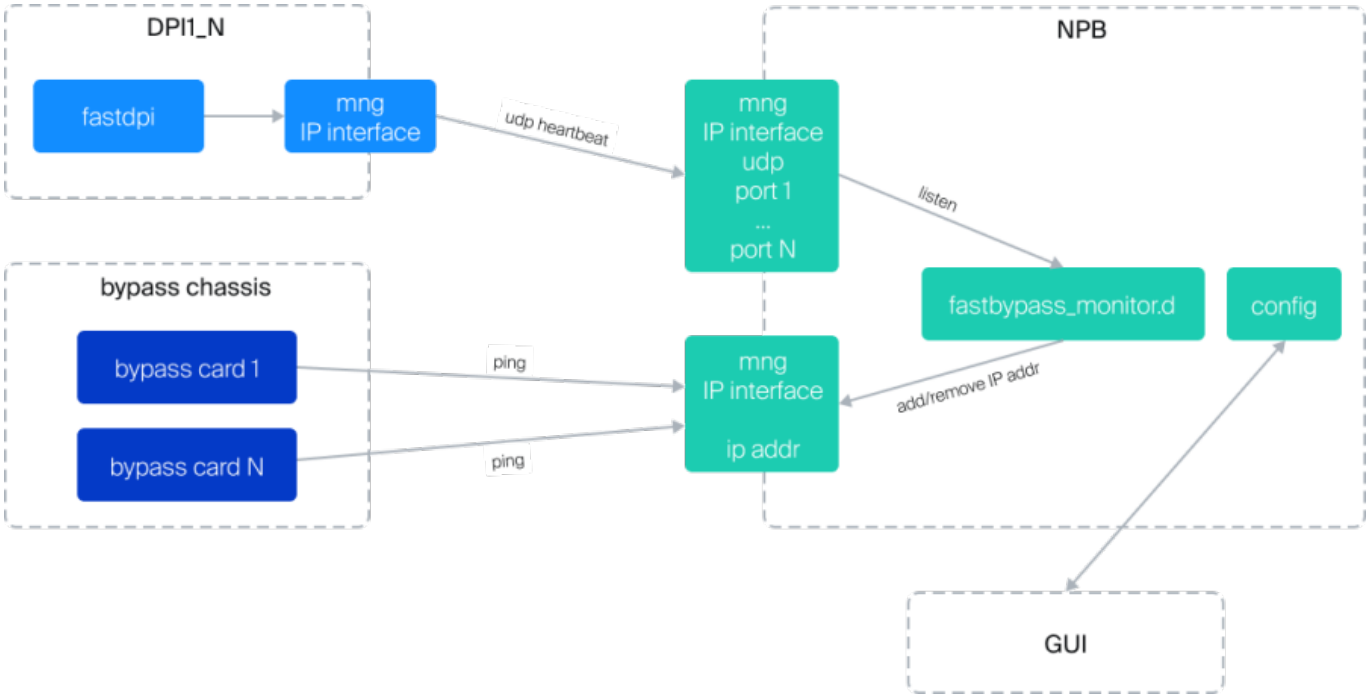
FastBypass monitor 3
- ***Hardware Requirements*** 3
- ***Key Features*** 3
- ***Installation*** 4
- ***Usage*** 4
  - Local and Global States: Bypass Mode 6
- ***Configuration*** 7
  - Minimal Configuration 7
  - General Configuration 8
  - Listener Configuration 10
  - Bypass Network Card Interface Configuration 11

FastBypass monitor

`fastbypass_monitor` allows implementing a case of DPI operation with external BYPASS systems.

`fastbypass_monitor` (referred to as "daemon" further in the documentation and script) is a tool for monitoring and managing the state of network interfaces connected to Bypass network cards.

The daemon reacts to HEARTBEAT signals received from DPI on specific ports defined in the configuration file. If HEARTBEAT signals are not received according to the configuration rules, the daemon performs specific actions such as deleting or creating IP addresses connected to the Bypass cards and enabling or disabling certain network interfaces.



If there is a software failure on DPI, the NPB removes the DPI from the stack and redistributes the load among the remaining DPIs.
If more than two DPI nodes fail, the entire system switches to bypass mode.
If the link on a DPI fails, NPB redistributes the load among the remaining DPIs.

Hardware Requirements

OS: OpenSwitch 2+ / Debian 9+
Python: 2.7.9

Key Features

- Monitoring HEARTBEAT signals from DPI on specified ports.
- Dynamic management of IP addresses and network interfaces.

Installation

1. Copy the installation package `fastbypass_monitor-X.X.XX.deb` to the host machine.
2. Run the following command from the directory where the package is located:

```
sudo dpkg -i fastbypass_monitor-X.X.XX.deb
```

After installation, the daemon becomes manageable through the system manager (`systemctl`).

The configuration file is available at `/var/fastbypass_monitor/backend/.env`
A sample configuration file can be found at `/var/fastbypass_monitor/backend/sample.env`
Daemon logs are stored at `/var/fastbypass_monitor/backend/logs/`

Usage

After installation, the daemon runs automatically. Upon reboot, it starts after the `network` service has successfully launched.
Manage the daemon using system manager commands.


Aliases (short command equivalents) can only be used with sudo. Use `sudo su -` and enter the password to enable this mode.

Start the daemon:

```
sudo systemctl start fastbypass_monitor
```

Alias:

```
fbypass_ctl start
```


The service starts in an unknown state, meaning it does not initially enable or disable bypass mode. After all receivers are initialized and their statuses are determined, the system switches to either normal or bypass mode depending on configuration and receiver status.

Restart the daemon:

```
sudo systemctl restart fastbypass_monitor
```

Alias:

```
fbypass_ctl restart
```

Reload the daemon without stopping:

```
sudo systemctl reload fastbypass_monitor
```

Alias:

```
fbypass_ctl reload
```

Stop the daemon:

```
sudo systemctl stop fastbypass_monitor
```

Alias:

```
fbypass_ctl stop
```

Check the daemon’s status:

```
sudo systemctl status fastbypass_monitor
```

Alias:

```
fbypass_ctl status
```

View the last few lines of the log file in real-time:

```
tail -f /var/fastbypass_monitor/backend/logs/fastbypass_monitor.log
```

Alias:

```
fbypass_ctl tailf
```

Output the last 100 lines of the log:

```
tail -n 100 /var/fastbypass_monitor/backend/logs/fastbypass_monitor.log
```

Alias:

```
fbypass_ctl tail 100
```

Stop the daemon and remove IPs from Bypass cards, forcing the system into bypass

mode:

```
fbypass_ctl force_on
```

Stop the daemon and add IPs to Bypass cards, forcing the system into normal mode:

```
fbypass_ctl force_off
```

Add the daemon to startup:

```
fbypass_ctl enable
```

Remove the daemon from startup:

```
fbypass_ctl disable
```

To configure and launch the daemon with new settings, edit the configuration file and restart or stop and start the daemon.

The daemon configuration is located at `/var/fastbypass_monitor/backend/.env`

Upon startup and reload, the daemon reads the configuration file. If it can successfully configure the specified IPs and interfaces, it continues operation. Otherwise, it only launches successfully configured components.

In case of a critical error, the daemon will restart automatically.

Using `sudo systemctl reload fastbypass_monitor` will reload the configuration without stopping the daemon, shutting down removed components, and adding new ones.

During startup and reload, the daemon does not manage interfaces and IPs until all listeners report their statuses. After a restart, the daemon remains in its previous state until receiving updates from all listeners.

Local and Global States: Bypass Mode

The daemon manages interfaces based on either a **global** state (depending on all listeners) or a **local** state (specific to individual listeners).
For instance, if you list interfaces in the global settings, they will be enabled or disabled based on the daemon's overall state. If the daemon fails to receive enough signals, the interfaces are disabled.

Example:

```
LISTEN_CUBRO_IFS=<interface list>
LISTEN_SHUTDOWN_CUBRO_IFS_WHEN_BYPASS=1
```

Each listener can also have its own interface list that it manages based on its state.

Example:

```
LISTEN_CUBRO_IFS[0]=<interface list>
LISTEN_SHUTDOWN_CUBRO_IFS_WHEN_BYPASS[0]=1
```

If the interfaces specified in `LISTEN_CUBRO_IFS[N]` are duplicated in multiple listener, they go into bypass mode if one of the corresponding listener stops receiving a signal. The interfaces are only bypassed to normal mode if all the corresponding listener receives signals.

In the case where interfaces are specified in both local and global settings, the interfaces are in bypass mode until the corresponding listener starts receiving signals and the daemon enters normal mode.

Configuration

Minimal Configuration

The minimum configuration for daemon operation includes specifying at least one interface, IP address, and port to receive HEARTBEAT signals, and one interface and IP address to connect the Bypass card.

Example:

```
# Logging level - error and information messages
LOG_LEVEL=INFO

# interface for default listener operation
LISTEN_HEARTBEAT_IFS=eth0
# interface for working with Bypass cards by default
BYPASS_CARD_IFS=eth0

# number of unsuccessful HEARTBEAT listener for switching to Bypass mode
LISTEN_HEARTBEAT_FAILED=1
# number of attempts to receive the default HEARTBEAT signal for the
listener
LISTEN_HEARTBEAT_ATTEMPTS=1
# default HEARTBEAT signal waiting time in milliseconds for listener
LISTEN_HEARTBEAT_TIMEOUT=3000

# IP address where the daemon is waiting for HEARTBEAT signals
LISTEN_HB_HOST[0]=192.168.1.202
# port on which the daemon expects HEARTBEAT signals
LISTEN_HB_PORT[0]=3000
```

```
# IP address where the daemon is waiting for HEARTBEAT signals
LISTEN_HB_HOST[1]=192.168.1.202
# port on which the daemon expects HEARTBEAT signals
LISTEN_HB_PORT[1]=3100

# IP address of the Bypass card to which the daemon will be connected
BYPASS_CARD_HOST[0]=192.168.1.211

# IP address of the Bypass card to which the daemon will be connected
BYPASS_CARD_HOST[1]=192.168.1.212
```

Above is a sample configuration for receiving HEARTBEAT signals using interface `eth0` on IP address `192.168.1.202` and ports `3000` and `3100`

Bypass cards are connected on interface `eth0` to IP addresses `192.168.1.211` and `192.168.1.212`.

The default values for all listener are set to:

`LISTEN_HEARTBEAT_ATTEMPTS` — number of attempts to receive the signal: `1`
`LISTEN_HEARTBEAT_TIMEOUT` — signal waiting time: `3000` milliseconds

When the listener does not receive a signal after one attempt within `3000` milliseconds, it is considered to have failed.

If the number of failed listeners equals or exceeds the set threshold (`LISTEN_HEARTBEAT_FAILED`), the daemon enters Bypass mode and removes the IP addresses specified for Bypass maps.

When signals are restored, the listener is considered operational.

If the total number of failed listeners becomes less than the threshold, the daemon returns to NORMAL mode and restores the specified IP addresses to the Bypass cards.

General Configuration

The configuration settings below apply to the overall functioning of the daemon.

The listener settings block serves as default values for those listeners that do not have specific configurations defined. The same applies to interface settings for connecting Bypass network cards.

Additionally, the daemon allows the integration of custom commands to manage interfaces and IP addresses. This makes it possible to tailor the daemon to the specifics of the network and implement custom scripts optimized for specific network environment requirements.

```
# logging level (optional setting, default is INFO):
# INFO - error and informational messages
# DEBUG - error, informational, and debug messages
LOG_LEVEL=

# network mask for specified IP addresses (optional setting, default is 32)
```

```
//NETWORK_MASK=//
```

```
# default interface for listeners that do not specify an interface name for
listening to HEARTBEAT (optional)
LISTEN_HEARTBEAT_IFS=

# default IP address for listeners that do not specify an IP address for
listening to HEARTBEAT signals (optional)
//LISTEN_HEARTBEAT_HOST=//

# default number of attempts to receive a HEARTBEAT signal for listeners
without a specific configuration (optional, default is 3)
//LISTEN_HEARTBEAT_ATTEMPTS=//

# default timeout in milliseconds for HEARTBEAT signal for listeners without
a specific configuration (optional, default is 3000)
//LISTEN_HEARTBEAT_TIMEOUT=//

# number of failed HEARTBEAT listeners to trigger Bypass mode (optional,
default is 1)
//LISTEN_HEARTBEAT_FAILED=//
```

```
# default interface for working with Bypass cards that do not specify an
interface name (optional)
BYPASS_CARD_IFS=
```

```
# list of interfaces to shut down when entering Bypass mode (optional)
LISTEN_CUBRO_IFS=

# setting for handling interfaces listed in //LISTEN_CUBRO_IFS// (optional,
default is 0)
# 1 - shut down specified interfaces when entering Bypass mode
# 0 - take no action with specified interfaces when entering Bypass mode
LISTEN_SHUTDOWN_CUBRO_IFS_WHEN_BYPASS=
```

```
# absolute path to the script for bringing up an interface (optional,
default – see Note 2)
CMD_SET_UP_INTFS=

# absolute path to the script for bringing down an interface (optional,
default – see Note 2)
CMD_SET_DOWN_INTFS=

# absolute path to the script for adding an IP address (optional, default –
see Note 2)
CMD_ADD_IP=

# absolute path to the script for removing an IP address (optional, default
– see Note 2)
CMD_DEL_IP=
```

Note 1

All IP addresses specified in the configuration may be presented in the format `192.168.1.202` or with a network mask, e.g., `192.168.1.202/16`. By default, the network mask is 32 (if not set in the global `NETWORK_MASK` setting or in a specific listener or Bypass card configuration).

Important: If the IP address matches the management IP address used for SSH connections, then the network mask is not changed and remains as defined in the operating system.

This important condition should be considered when configuring IP addresses to avoid conflicts with the management IP and unintended network mask changes.

Note 2

The daemon configuration allows specifying custom scripts for performing basic operations such as enabling/disabling interfaces and creating/removing IP addresses.

The daemon expects the absolute path to a shell script in the respective configuration, with variables specified at the end of the line in the format `%(<variable_name>)s`

Used variables:

- `intfs` — interface name
- `ip` — IP address
- `netmask` — network mask

Default configurations:

- `CMD_SET_UP_INTFS=/var/fastbypass_monitor/backend/app_bash/cmd_set_up_intfs.sh %(intfs)s`
- `CMD_SET_DOWN_INTFS=/var/fastbypass_monitor/backend/app_bash/cmd_set_down_intfs.sh %(intfs)s`
- `CMD_ADD_IP=/var/fastbypass_monitor/backend/app_bash/cmd_add_ip.sh %(ip)s %(netmask)s %(intfs)s`
- `CMD_DEL_IP=/var/fastbypass_monitor/backend/app_bash/cmd_del_ip.sh %(ip)s %(netmask)s %(intfs)s`

Listener Configuration

Each listener enables receiving HEARTBEAT signals from various DPI devices. Each subsequent listener is specified with the next index (e.g., [0], [1], [2]).

A listener has the following parameters for complete configuration:

```
# listener ID (optional, default is the index)
```

```
LISTEN_HB_ID[0]=0

# name of the interface where the listener expects HEARTBEAT signals
LISTEN_HB_IFS[0]=eth0

# IP address/subnet mask for listening to HEARTBEAT signals
LISTEN_HB_HOST[0]=192.168.1.202/32

# port for listening to HEARTBEAT signals
LISTEN_HB_PORT[0]=3000

# number of attempts to receive a HEARTBEAT signal (optional, default: 3)
LISTEN_HB_ATTEMPTS[0]=3

# HEARTBEAT signal timeout in milliseconds (optional, default: 3000)
LISTEN_HB_TIMEOUT[0]=3000

# setting for immediate switch to Bypass mode (optional)
# 1 - if no HEARTBEAT signal received, switch to Bypass immediately
# 0 - if no HEARTBEAT signal received, switch after all attempts (default)
LISTEN_HB_SWITCH_IMMEDIATELY[0]=0

# list of interfaces to shut down when entering Bypass mode (optional)
LISTEN_CUBRO_IFS[0]=e101-001-0,e101-002-0

# setting to shut down interfaces specified in LISTEN_CUBRO_IFS[N]
(optional)
# 1 - shut down specified interfaces when entering Bypass mode
# 0 - take no action with specified interfaces when entering Bypass mode
(default)
LISTEN_SHUTDOWN_CUBRO_IFS_WHEN_BYPASS[0]=1
```

Bypass Network Card Interface Configuration

The daemon automatically manages (adds/removes) IP addresses on the respective interfaces when switching to BYPASS or NORMAL mode according to the Bypass card settings.
Each Bypass card is specified with the next index (e.g., [0], [1], [2]).

Bypass card interfaces have the following parameters for full configuration:

```
# Bypass card ID (optional, default is the index)
BYPASS_CARD_ID[0]=

# IP address/subnet mask for listening to HEARTBEAT signals
BYPASS_CARD_HOST[0]=

# Bypass card operation mode
# 0 - remove specified IP address when Bypass mode is enabled
# 1 - remove specified IP address when Bypass mode is enabled, add it back
when Bypass mode is disabled (default)
```

```
BYPASS_CARD_ACTIVE[0]=

# forced Bypass card mode
# 0 - disable forced mode (default)
# 1 - enable forced mode, IP address remains active regardless of daemon
state
BYPASS_CARD_FORCE[0]=
```