Содержание

Administration Questions

1. How to check the current release (CCC)?

By using the command

```
fastdpi -re
```

2. How to check the current version?

By using the command

```
fastdpi -ve
```

3. How to roll back to a previous version?

Example of rolling back from version 2.7 to 2.6:

```
yum downgrade fastdpi-2.6
```

4. What does the error 'error loading DSCP settings, res=-4' mean?

The error occurs due to the absence of DSCP by autonomous systems. It can be ignored.

5. What to do if not all commands are processed and the error 'ERROR : Can't connect to 127.0.0.1:29000, errcode=99 : Cannot assign requested address Autodetected fastdpi params : dev='lo', port=29000 connecting 127.0.0.1:29000 ...' appears?

`fdpi_ctrl` uses the regular Linux stack to communicate with DPI, so tuning recommendations are similar to those for web servers (like nginx) under high load.
The settings are similar to those recommended for nginx and should be added to the `/etc/sysctl.conf` file to persist after reboot:

```
# OS network stack optimization
net.core.netdev_max_backlog=10000
net.core.somaxconn=262144
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_max_syn_backlog = 262144
net.ipv4.tcp_max_tw_buckets = 720000
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_keepalive_probes = 7
net.ipv4.tcp_keepalive_intvl = 30
net.core.wmem_max = 33554432
net.core.rmem_max = 33554432
net.core.rmem_default = 8388608
net.core.wmem_default = 4194394
net.ipv4.tcp_rmem = 4096 8388608 16777216
```

```
net.ipv4.tcp_wmem = 4096 4194394 16777216
```

for a 1 Gbps interface:

```
net.core.netdev_max_backlog=10000
```

for a 10 Gbps interface:

```
net.core.netdev_max_backlog=30000
```

To apply the changes without rebooting, you can change them on the fly by using the command

```
sysctl -w <setting>
```

For example:

```
sysctl -w net.ipv4.tcp_tw_reuse=1
```

This should resolve the issue.

For CentOS 7
Example:

```
# OS network stack optimization
net.core.netdev_max_backlog=65536
net.core.optmem_max=25165824
net.core.somaxconn=1024
net.ipv4.tcp_max_orphans = 60000
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_sack = 1
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_max_syn_backlog = 262144
net.ipv4.tcp_max_tw_buckets = 720000
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_keepalive_probes = 7
net.ipv4.tcp_keepalive_intvl = 30
net.core.wmem_max = 33554432
net.core.rmem_max = 33554432
net.core.rmem_default = 8388608
net.core.wmem_default = 4194394
net.ipv4.tcp_rmem = 4096 8388608 16777216
net.ipv4.tcp_wmem = 4096 4194394 16777216
```

Update command:

```
sysctl –system
```

More information on CentOS7

Scripts for migrating from SCE SM to SSG database, description inside

6. How to check the load by cores and understand why they are unevenly loaded?

To view CPU load by cores in the `top` utility, press 1.
To view load by DPI tasks, use the command:

```
ps -p `pidof fastdpi` H -o %cpu,lwp,pri,psr,comm
```

Example output:

```
  %CPU   LWP PRI PSR COMMAND
 0.0  23141  41   0 fastdpi_main
 0.0  23146  41   0 fastdpi_dl
 0.3  23147  41   0 fastdpi_ctrl
 35.8 23148  41   0 fastdpi_ajb
 32.7 23152  41   1 fastdpi_rx_1
 34.1 23165  41   2 fastdpi_wrk0
 34.1 23170  41   3 fastdpi_wrk1
```

In DPI, the COMMAND tasks are functionally separated by cores (PSR) to avoid interfering with each other's work:

1. `wrk` threads perform data analysis in network packets
2. `rx` thread handles data transit between network ports
3. other threads perform application and auxiliary tasks (generating Netflow, receiving control commands, loading lists, writing PCAP, etc.) and can create peak loads on the CPU, so they are assigned to a separate core.

7. What to do if the error in fastdpi_alert.log '[CRITICAL][2017/10/06-16:36:44:616019][0x7fdb297ac700] metadata_storage : Can't allocate memory [repeat 1], cntr=188889, allocated=188889' appears?

In DPI, everything is pre-allocated, by default for the number of subscribers specified in the error (188889). This is controlled by the parameter in the configuration file `mem_ip_metadata_recs`.
For example, to increase to 500000 subscribers, set in the `/etc/dpi/fastdpi.conf` configuration file:

```
mem_ip_metadata_recs=500000
```

After changing the parameter, a restart is required:

```
service fastdpi restart
```

8. Which files are recommended to be archived?

```
cp /etc/pf_ring/ /BACKUPDIR/pf_ring
cp /etc/dpi /BACKUPDIR/etc/
```

```
mdb_copy /var/db/dpi /BACKUPDIR/db/
```

With `mdb_copy`, you can make a backup while fastDPI is running.

9. What to do if ipmi uses 100% CPU and interferes with DPI operation?

Run the command

```
echo 100 > /sys/module/ipmi_si/parameters/kipmid_max_busy_us
```

To ensure the setting is not lost after a server reboot, add this command to `/etc/rc.local`

10. What to do if an error in the alert log '[ERROR] bpm : thread #1 - does not change self-monitoring counters', DPI restarted and generated a core or switched to bypass?

DPI performs self-diagnostics during operation, and if one of the working threads hangs and can no longer process traffic, DPI detects this state and restarts with core generation on Abort signal.

Important: trace and dbg settings in `fastdpi.conf` are intended for diagnostics and debugging, not for permanent operation. If disk write is blocked by another process (e.g., log rotation, which typically occurs from 3 to 4 AM), enabling tracing may cause the working thread to block on writing to the diagnostic (slave) log, leading to DPI switching to bypass or restarting. After diagnostics, remember to disable these settings.

The problem occurs only on certain servers. If your server is affected, we recommend changing the standard disk scheduler to deadline:

```
echo deadline > /sys/block/sda/queue/scheduler
echo deadline > /sys/block/sdb/queue/scheduler
```

11 . Why does the memory consumption of the process increase during operation?

DPI allocates memory statically: at process start and when creating certain service profiles (such as NAT, black and white lists). During operation, additional memory is not allocated. So why does consumption increase?
Linux distinguishes between resident (RES in top) and virtual (VIRT in top) process memory. The peculiarity is that while memory is uninitialized (actually initialized to zero), Linux does not record it as resident and moves it there as it is initialized.
Setting `mem_preset=1` in `/etc/dpi/fastdpi.conf` instructs DPI to initialize all allocated memory (almost all), preventing the resident part from growing during operation. This option slows down the startup and is good when there is enough physical RAM. It is better to consider this factor and monitor virtual (VIRT) and resident (RES) memory separately.

12. What to do if there are many "zombie" processes with names ''wd_*'' on one of the SSG?

```
166206 ?        Z      0:00  \_ [wd_fastdpi.sh] <defunct>
166219 ?        Z      0:00  \_ [wd_fastpcrf.sh] <defunct>
```

Simply restart the watchdog:

```
service watchdog restart
```

13. Protocol or signature detection problem

In case of protocol or signature detection issues, perform three tests on each of the following devices:

- Personal computer
- Smartphone on iOS
- Smartphone on Android

The following recommendations help eliminate unnecessary traffic:

- It is recommended to conduct the test on a PC in incognito mode.
- When testing on a smartphone, enable power-saving mode.

Test execution:

1. Check if the following parameters are enabled in the `/etc/dpi/fastdpi.conf` file:

   ```
   trace_ip="subscriber's IP"
   ajb_save_ip="subscriber's IP"
   plc_trace_ip="subscriber's IP"
   ```

   If any of these parameters are enabled, comment them out and perform `service fastdpi reload`.

2. Execute the command

   ```
   find /var/log/dpi -type f -name "fastdpi_slave_*.log" -exec sh -c 'cat /dev/null > {}'  \;
   ```

   The command should clear the `fastdpi_slave_*.log` files.

3. Delete all files from `/var/dump/dpi/`.
4. Open the `/etc/dpi/fastdpi.conf` file in a text editor. Add the parameters:

   ```
   trace_ip="subscriber's IP"
   ajb_save_ip="subscriber's IP"
   plc_trace_ip="subscriber's IP" #For this parameter to work, the test subscriber must have a policing profile
   ```

5. Prepare the test subscriber for launch to generate problematic traffic.
6. Perform `service fastdpi reload`.
7. Start generating traffic. Record traffic for 1 minute.
8. Open the `fastdpi.conf` file. Comment out the parameters:

   ```
   trace_ip="subscriber's IP"
   ajb_save_ip="subscriber's IP"
   plc_trace_ip="subscriber's IP"
   ```

9. Perform `service fastdpi reload`.
10. Prepare the output of the following commands into files:

```
“fastdpi -ve”
“dscp2lst /etc/dpi/protocols.dscp”
“fdpi_ctrl list --policing --ip “subscriber's IP”
“dscp2as /etc/dpi/asnum.dscp”.
```

11. Prepare an archive with the files from point 10 and the `fastdpi.conf` file.
    From `/var/log/dpi` — `fastdpi_stat.log`, `fastdpi_slave_*.log`.
    From `/var/dump/dpi` — `udp_*.pcap`.
12. Repeat the required number of tests with different devices. Indicate in the archive name or in a `readme.txt` file within the archive the types of devices used for testing.
13. Attach the archives to the ticket. If the archives are too large, upload them to any cloud file sharing service and send us the link.