Table of Contents

Description of NAT statistics	3
Common for all profiles	3
CG-NAT Profile	
Information on the data converter	
1:1 Profile	
Output of public addresses statistics	

4 Description of NAT statistics

The output of statistics in fastdpi_stat.log is set by dbg_log_mask.

0x40000 - output of NAT initialization details (profiles, public addresses, etc.) to the alert log 0x100000 - output of statistics by blocks of public addresses (if 0x2000000 is set) 0x2000000 - NAT statistics output.

Example:

dbg log mask=0x2000000

Common for all profiles

```
[STAT ][2021/12/22-17:33:17:513859] NAT statistics : itrnsl=0, iprof=2,
profile 'cgnat', nttype=0, ref_cnt=1, cidr=94.140.198.84/30
[STAT ][2021/12/22-17:33:17:513869] NAT statistics : itrnsl=1, iprof=3,
profile 'nat1_1', nttype=1, ref_cnt=1, cidr=16.35.121.0/24

itrnsl - internal index of the private<-->public proile converter.
   iprof - internal index of the profile
   profile - profile name
   nttype - profile type (0 - cgnat, 1 - 1:1)
   ref_cnt - counter of references of converter usage by profiles
   (Profiles can use one set of CIDR, but different number of restrictions
per session)
   cidr - list of profile's public addresses CIDR
```

CG-NAT Profile

Information on the data converter

```
k=0, itrnsld=0, cidr=94.140.198.84/30
k - itrnsld number
itrnsld - internal converter data index - the one serving the CIDR
cidr - exact cidr
```

```
total - total statistics actual - статистика изменеий счетчиков за период вывода статистики (delta_alarm parameter, 15 seconds by default) total TCP: 30/20/0/7/17 0/0/0 50/20/0/50/0 5516/8/121
```

Four groups группы:

1 group -- operations with public address ports

30/20/0/7/17:

- 30 allocation of a new public port
- 20 public port reuse
- 0 errors of allocating a new public port
- 7 performs decrement of number of subscriber sessions on flow releasing
- 17 performs decrement of the number of subscriber sessions on public port reuse

2 group -- general statistics

0/0/0:

- θ calculated CRC by IP when accessed to allocate a public address. Should be == θ
- O excessive number of sessions for subscribers
- 0 different public addresses in flow and converters Should be == 0

3 group -- cache statistics *private* -→ *public*

50/20/0/50/0:

- 50 added records to cache
- 20 deleted records from cache
- 0 found records in cache when new public port is allocated
- 50 no public address found for private
- 0 errors of adding to cache

4 group -- conversion statistics *public* -→ *private* (*inet*-→*subs*)

5516/8/121:

- 5516 successful translation public --> private
- 8 port is not in the allocated range of public ports
- 121 translation public --> private was not found

For TCP/UDP and total/actual statistics are the same.

For GRE – it is GRE by default (when session is not found in PPTP). Only one such session can be created per public address.

```
total GRE : 0/0
0 - address used
0 - number of attempts to create sessions on an already allocated public address
```

1:1 Profile

```
[STAT ][2021/12/22-17:17:28:749622] NAT statistics : itrnsl=1, iprof=3, profile 'nat1_1', nttype=1, ref_cnt=1, cidr=16.35.121.0/24 k=0, itrnsld=1, cidr=16.35.121.0/24 total 256/256/0/0/0/0 0/0
```

Group 2 statistics: 1 group - 256/256/0/0/0/0 2 group - 0/0

Example:

dbg log mask=0x2100000

Output of public addresses statistics

TCP whiteblck ip mask=0x0, nwhaddr=2:

awhb=4, fwhb=1004, puwhb=0.40%

```
[STAT
         [2021/12/22-21:14:48:385991] NAT statistics : itrnsl=0, iprof=2,
profile 'cgnat', nttype=0, ref cnt=1, cidr=94.140.198.84/30
        k=0, itrnsld=0, cidr=94.140.198.84/30
                total TCP: 26/4/0/4/2/ 0/0/0 30/4/0/30/0 3045/1/36
                actual TCP : 0/0/0/0/0 0/0/0 0/0/0/0/0 0/0/0
                        TCP whiteblck ip mask=0x0, nwhaddr=2
                                whip=94.140.198.84 : sb=64, lsb=64, nb=1008,
whpa=64512, whpb=0, whpf=64512, awhb=4, fwhb=1004, puwhb=0.40%
                                        thr=0, ublock=1, uport=0
                                        thr=1, ublock=1, uport=0
                                        thr=2, ublock=1, uport=0
                                        thr=3, ublock=1, uport=0
                                whip=94.140.198.86 : sb=64, lsb=64, nb=1008,
whpa=64512, whpb=26, whpf=64486, awhb=4, fwhb=1004, puwhb=0.40%
                                        thr=0, ublock=1, uport=0
                                        thr=1, ublock=1, uport=0
                                        thr=2, ublock=1, uport=13
                                        thr=3, ublock=1, uport=13
```

```
ip_mask - addresses mask
nwhaddr - the number of public addresses that are under the mask
whip=94.140.198.84 : sb=64 ( 64 ), nb=1008, whpa=64512, whpb=0, whpf=64512,
```

```
whip=94.140.198.84 - public address
sb=64 - port block size
lsb=64 - size of the last block
nb=1008 - number of port blocks
whpa=64512 - total ports
whpb=0 - ports occupied
whpf=64512 - free ports
awhb=4 - blocks issued
fwhb=1004 - free blocks
puwhb=0.40% - percentage of blocks occupied
```

Added in version 12.1.0

```
whp_salfs - how many ports are in the 'short' queue
whp_lalfs - how many ports are in the 'long' queue
whp_ruse - how many ports can be reused
whp_ruse_salfs - how many ports can be reused from the 'short' queue
whp_ruse_lalfs - how many ports can be reused from the 'long' queue
whp_dthr - how many ports were created in one worker thread but used in
another thread
whp_dthr_salfs - how many ports were created in one worker thread but used
in another from the 'short' queue
whp_dthr_lalfs - how many ports were created in one worker thread but used
in another from the 'long' queue
```

Within the public address you can see the distribution of captured ports/blocks by worker threads

```
thr_salfs - how many ports are in the 'short' queue
thr_lalfs - how many ports are in the 'long' queue
thr_ruse - how many ports can be reused
thr_ruse_salfs - how many ports can be reused from the 'short' queue
thr_ruse_lalfs - how many ports can be reused from the 'long' queue
thr_dthr - how many ports were created in one worker thread but used in
another thread
thr_dthr_salfs - how many ports were created in one worker thread but used
in another from the 'short' queue
thr_dthr_lalfs - how many ports were created in one worker thread but used
in another from 'long' queue
```

The output format is the same.

```
fdpi_ctrl list status --service 11 --ip 192.168.4.20
Autodetected fastdpi params : dev='em1', port=29001
connecting 94.140.198.68:29001 ...
```

Output:

```
- private IP
192.168.4.20
crcip=0xd649d853
                    - CRC of the private IP
nttype=0
                    - NAT type: 0 - cgnat, 1 - 1:1
profile='cgnat' - profile name
itrnsl=0
                - internal index of the converter private<-->public profile.
                    - internal converter data index
itrnsld=0
whiteip=94.140.198.86 - public address
sess tcp=127

    number of TCP sessions

                  - number of UDP sessions
sess udp=108
indmtd=4
                       - internal index of subscriber data (subscriber
metadata)
fdpi ctrl list status --service 11 --ip 192.168.4.20 --outformat=json
fdpi ctrl list status --service 11 --ip 192.168.4.20 --outformat=json
                                                                       ljq
fdpi ctrl list all status --service 11
fdpi ctrl list all status --service 11 --outformat=json
```

The format is the same.

Key to command to view NAT profile statistics via fdpi ctrl

Command:

```
fdpi_ctrl list status --service 11 --profile.name cgnat
```

Output:

```
profile='test nat cgnat'
                                                itrnsl=0
                                                                nitrnsld=1
nttype=0
        itrnsld=0
                        cidr=94.140.198.84/30
                proto=TCP
                                ip mask=0x0
                                                nwhaddr=2
                        proto=TCP
                                        ip mask=0x0
                                                        whip=94.140.198.84
       lsb=64 nb=1008 whpa=64512
                                        whpb=0
                                                whpf=64512
sb=64
                                                                awhb=4
fwhb=1004
                puwhb=0.40%
                                whp salfs=0
                                                whp lalfs=0
                                                                whp ruse=0
whp ruse salfs=0
                        whp ruse lalfs=0
                                                whp dthr=0
whp dthr salfs=0
                        whp dthr lalfs=0
                                nthr=0 ublock=1
                                                        uport=0 thr salfs=0
thr_lalfs=0
                thr_ruse=0
                                thr_ruse_salfs=0
                                                        thr_ruse_lalfs=0
thr dthr=0
                thr dthr salfs=0
                                        thr dthr lalfs=0
                                nthr=1 ublock=1
                                                        uport=0 thr salfs=0
                                                        thr ruse lalfs=0
thr_lalfs=0
                thr ruse=0
                                thr ruse salfs=0
```

thr_dthr=0 thr_dthr_salfs=0 thr_dthr_lalfs=0

Key:

```
- profile type (0 - cgnat, 1 - 1:1)
nttype
profile
               - profile name
itrnsl
               - internal index of private<-->public profile converter
               - number of profile converter data (number of CIDR)
nitrnsld
               - internal converter data index - the one serving the CIDR
itrnsld
               - exact CIDR
cidr
               - TCP/UDP
proto
               - addresses mask
ip mask
               - the number of public addresses which fall under the mask or
nwhaddr
CRC (depends on rx_dispatcher parameter)
whip
               - public address
sb
               - size of the block of ports to be allocated
              - size of the last block
lsb
               - number of prot blocks
nb
whpa
               - ports in total
               - ports occupied
whpb
               - free ports
whpf
              - blocks issued
awhb
               - free blocks
fwhb
dhwuq
              - percentage of blocks occupied
whp_salfs
              - is in the 'short' queue
whp_lalfs - In the can be used
               - in the 'long' queue
whp_ruse_salfs - can be used in 'short' queue
whp ruse lalfs - can be used in 'long' queue
whp dthr - number of elements ithr owner != ithr by queue
whp_dthr_salfs - number of elements ithr_owner != ithr by 'short' queue
whp_dthr_lalfs - number of elements ithr owner != ithr by 'long' queue
```

Key to command to view the dump NAT profile statistics

Command:

```
fdpi_cli -r 127.0.0.1:29001 nat dump whaddr queue test_nat_cgnat
```

Output:

Key:

The implementation of the used ports queue for public addresses uses a single array - let's call it WHP, of size 0xffff. It is used to build a list of used ports for the worker thread. Index 0 is used as a stub (empty).

Thread queues cannot be output as a list, because records in the queue are moved in the process, which may cause the output to loop. Therefore, the WHP array is output 'as is' for occupied entries.

```
entryp : sets entry points to the list of public ports of the worker thread
  ithr  - worker thread number
  ihead - top of the list
  itail - the last element of the list
```

```
data : WHP white port array data (only occupied entries are output)
    sind
                 - record index
   inext
                     - next record index
                     - previous record index
   iprev
   whport
                     - public port
                     - private address, which the public address is assigned
   graddr
to
   tml
                     - time of the last record access
                     - timeout, time in seconds, the record lifetime
   lifetime
(depends on the parameters for short/long queue)
                     - sign that the record can be used again
    canreuse
   ialf
                     - number of the processing queue :
                            en nalfs shrt = 0, # queue with a short
lifetime
```

```
en_nalfs_long = 1, # long queue
                    - internal index of the subscriber's data (subscriber's
   indmtd
metadata)
                    - owner thread which has created the record
   iown
                    - number of the thread which has last accessed the
   ilst
record
                    - the protocol the record was allocated for from UDP
   subproto
                       typedef enum en_nat_borrw_udp: u_int8_t
                           ennatborwu ORG = 0, # UDP/TCP
                           ennatborwu_DFLTGRE = 1, # общий GRE
                           ennatborwu MAX = 2, # ICMP
                       } en nat borrw udp t;
   decr_sess
                    - a mark that the port usage counter on the private
address has decremented
   ind gcache slice - index of the 'private --> public' transcoding cache-
slice
   igcache - index in the corresponding cache-slice recoding
'private --> public'
```