

# Содержание

<b>Configuring Full NetFlow Export in IPFIX Format</b> .....	3
<b>General Required NetFlow Configuration Settings</b> .....	3
Configuration Example .....	4
<b>General Additional NetFlow Configuration Settings</b> .....	4
Sending a template via IPFIX .....	4
<b>Full NetFlow Configuration</b> .....	5
<b>IPFIX (Netflow v10) export template for the IPv4 protocol</b> .....	6
<b>IPFIX (Netflow v10) export template for IPv6</b> .....	6
<b>Configuring NetFlow v5</b> .....	6



# Configuring Full NetFlow Export in IPFIX Format

## General Required NetFlow Configuration Settings

Enabling statistics collection and export:

```
netflow=1
```

- 0 or not specified - option disabled
- 1 - export statistics by protocol (port numbers); see the section [Set up NetFlow export by protocols, directions and billing](#) for details
- 2 - export statistics by destination (autonomous system numbers); see the section [Set up NetFlow export by protocols, directions and billing](#) for details
- 4 - Export statistics for billing; see the section [Set up NetFlow export by protocols, directions and billing](#) for details
- 8 - Export full session statistics, Full NetFlow in NetFlow v5 or IPFIX format



Simultaneous export of Full NetFlow and billing statistics is enabled via the bitmask `netflow=12` (8 + 4). Billing statistics are converted to RADIUS Accounting via FastPCRF when `enable_acct=1` is enabled. [RADIUS Accounting Configuration](#)



You must assign a separate collector for each type to prevent data from getting mixed up!

The name of the network interface through which NetFlow statistics will be sent:

```
netflow_dev=eth2
```



IPFIX/Netflow settings can be changed without restarting fastDPI. The `ipfix_reserved` configuration parameter allows you to reserve the necessary memory to enable or modify IPFIX/Netflow settings. If IPFIX/Netflow settings are specified in the configuration file, memory reservation for IPFIX/Netflow is automatically enabled, and IPFIX/Netflow settings and new IPFIX/Netflow exporter types can be changed without rebooting fastDPI.



For receiving, processing, and storing IPFIX, we recommend using the [QoE Store statistics collection software](#) and the [DPIUI2 graphical interface](#).

Any universal IPFIX collector that supports templates, or the [IPFIX Receiver](#) utility, is suitable for collecting information in the IPFIX format.

## Configuration Example

See the [QoE Stor: DPI Configuration](#) section for a configuration example

## General Additional NetFlow Configuration Settings

Data export interval (in seconds):

```
netflow_timeout=10
```

The default value is 30 seconds.

Session timeout:

- `netflow_passive_timeout` — the timeout period (in seconds) for session activity; after this period, if there has been no activity, the session is considered terminated and data is transmitted via it. The default value is 30 seconds.
- `netflow_active_timeout` — the time (in seconds) after which information is reported for long sessions (i.e., long sessions are effectively broken down into segments of this duration). The default value is 300 seconds.

To smooth out spikes and distribute the load more evenly across the collector, set the configuration parameter

```
netflow_rate_limit=900
```

, where 900 is the maximum NetFlow rate in Mbps.  
The default value for this parameter is 0 (unlimited).



The parameter value should be set based on the following calculation: each DPI generates an IPFIX stream at a rate of 0.5% to 1% of the actual traffic rate. Setting a value that is too low will result in data being discarded at the DPI side. Information about this event will be recorded in the log file `/var/log/dpi/fastdpi_alert.log`.

## Sending a template via IPFIX

1. TCP transport protocol.  
The template is sent once after a TCP session is established.
2. UDP transport protocol.  
By default, the template is sent every 20 seconds. This can be adjusted using the `ipfix_udp_template_timer` parameter.

# Full NetFlow Configuration

Specify the IP address and port number of the **Full NetFlow** collector. You must assign a separate collector to each FastDPI to ensure that the data is not mixed with other statistics:

```
netflow_full_collector=192.168.0.1:9996
```

Specify the **Full NetFlow** export format:

```
netflow_full_collector_type=2
```

Possible values:

- **0** - Export in NetFlow5 format (default).
- **1** - Export IPFIX to a UDP collector.
- **2** - Export IPFIX to a TCP collector.

## We recommend using Full NetFlow transmission in IPFIX format over TCP (parameter value 2).

The NetFlow protocol does not guarantee packet delivery (since it operates over UDP), and if the collector cannot handle the incoming data, some packets will simply be lost. Transmitting **Full NetFlow** for 10 Gbps of DPI traffic requires the collector to be capable of receiving data at a rate of at least 60 Mbps.

Check your collector's capabilities before sending **Full NetFlow** statistics to it. At the same time, when transmitting **Full NetFlow** from DPI, short-term peaks of up to 100 Mbps may occur during spikes in the number of sessions.



When sending IPFIX over UDP, errors may occur due to incorrect settings on the receiving end, which can result in data loss. Examples of such errors:

```
[ERROR      ][2026/03/12-11:52:53:559204][0x7fdeba84b400]
IPFIX_ClickStream : udp:10.16.20.183:1502 : Error socket send to
collector, rc=-1, errno=113 : No route to host
[ERROR      ][2026/03/12-11:52:53:559243][0x7fdeba84b400]
IPFIX_ClickStream : udp:10.16.20.183:1502 : Error socket send to
collector ( repeat error 2 ), now ok.
```

The `netflow_plc_stat` parameter defines the set of statistics on dropped packets to be transmitted in accordance with policing or drop rules. The parameter is a bit mask.

By default, the mask has the value "**0x07**"—statistics on dropped data for session, subscriber, and virtual channel policing are transmitted.

⚠ Affects the calculation of the `DROPPED_BYTES` and `DROPPED_PACKETS` counters.

Values used to construct the mask:

- **0xff** - any drop is counted
- **0** - do not count
- **1** - count for session-based policing
- **2** - count for subscriber-based policing
- **4** - count for virtual channel policing

- **8** - count when packets are dropped by the protocol
- **16** - count in all other cases

The `ipfix_mtu_limit` parameter specifies the maximum size of a UDP packet when sending IPFIX. By default, it is set to the minimum MTU size of the interfaces used for transmission.

The parameter `tethering_ttl_allowed = 128:64` specifies a list of valid TTL values for traffic from the subscriber that is not considered tethering. Values are separated by a colon ':'. The number of values can be up to 256 (0-255).

## **IPFIX (Netflow v10) export template for the IPv4 protocol**

## **IPFIX (Netflow v10) export template for IPv6**

## **Configuring NetFlow v5**

In the Netflow v5 format, the original port numbers are retained in the full statistics, and information about the detected protocols is transmitted in the normally unused bytes 46-47. If you need to analyze the protocols in use, you can configure the system so that protocol information is transmitted in the port number:

```
netflow_full_port_swap=1
```

For backward compatibility with older collectors, this setting also applies to the IPFIX format; however, using it in conjunction with IPFIX is not recommended, as protocol information is transmitted in IPFIX in a separate, dedicated field.