Internet Draft
Document: <draft-molina-flow-selection-00.txt>
M. Molina
NEC Europe Ltd.
Expires: April 2004

October 2003

**Flow selection support in IPFIX**

Status of this Memo



Abstract

Flow selection is the process of selecting only a limited number of flows out of all the flows observable/observed at an IPFIX device. The selection can be done in the metering process, by selectively accounting only some of the incoming packets, in the flow recording process, by keeping only some of the created flow records, or in the exporting process, by exporting only some of the flow records of the flow recording process. This document describes the scenarios where flow selection can be applied, discusses what information about the flow selection process is beneficial to export and provides an information model for it.

Table of Contents

## 1. Introduction

The flow records exported out of an IPFIX device may be only a limited subset of the flows observable/observed at the observation points of the device. This may happen for several reasons, including resource limitations and/or explicit policies of the metering process, the flow recording process and the exporting process (functionalities of these processes are described in [Sad03]). For applications receiving and parsing flow information, it may be important to know details about the applied flow selection.
As an analogy, consider what happens with packet sampling. An application receiving counters relative to a flow whose packets were sampled needs to know details about the packet sampling procedure (e.g. the sampling ratio), in order to re-normalize the counters or simply to adjust its level of trust of the received information.

## 2. Limitations and scope

This document does not address the flow selection that can result from the sampling of packets in the metering process before flow classification. As an example, if 1 out of N sampling is applied before flow classification, some of the most short flows may not have even a single packet of them sampled for measurement, and therefore may be totally invisible to the IPFIX device.
Although the information about the number of packets not reaching the flow classification function because of sampling may be available, it is not in the scope of this document to describe if and how to export it.
This document is also not concerned about packets associated with a flow that are dropped (i.e. not forwarded) at an observation point, but that are anyway accounted for measurements, i.e. that account for the droppedPacketCount and droppedByteCount defined in [Cal03].
Dropping of packets but correct accounting of them may happen, for example, because of firewalling rules.
On the contrary, this document considers:

- the packet sampling that may be done in the metering process 1) after flow classification and 2) considering the flow state information contained in the flow recording process (if this block is present). This type of sampling is defined in [ZeSamp03] as flow state dependent sampling, but should be considered also in the IPFIX WG, because of its flow driven nature, as will be clarified later.
- the flow selection that can be done in the flow recording process (when present) and in the flow exporting process.

## 3. Causes of flow selection and relevant exportable information

We identify and describe some possible causes of flow selection, along with the information that can be beneficial to make available to applications about it.

### 3.1 Policies/resource limitations in the metering process

The main reason for applying in the metering process a packet selection driven by the state of the flow recording process (flow state dependent sampling) is that the flow recording process may not have, at a certain point in time, enough positions to record all observable flows. Another reason may be that there may not be enough processing resources to create and manage a new flow record.
To cope with these limitation, a number of possible policies can be applied, the simplest one being not to consider for measurement the new packets that do not belong to already existing flow records (i.e. that would require the creation of a new one).
More refined policies are however possible, mainly aimed at the so called elephant flow detection, i.e. to give priority in the flow recording process to flows carrying more traffic. For instance, [EsVa01] propose criteria to define a packet eligible to create a new flow record (sample and hold, multistage filters). [Molina03] proposes a method to prioritize the occupancy of flow recording process position according to a metric related to flowsÆ dynamic. In [Molina03] it may also happen that a flow record in the flow recording process is deleted in order to make room for a flow record opened by a newly arriving packet (we will explicitly consider the consequences of this case later in this document).
Independently of the specific algorithms, we are concerned here about defining what information it makes sense to keep about the flow state dependent packet sampling and make available to applications.
It is certainly possible to keep a cumulative counter of the total number of packets and bytes that were not considered for measurement because of flow state dependent sampling. Also, it is possible to keep a timestamp for the first and last of these non measured packets. This means, in practice, to aggregate all these packets in a macro flow, and keep track of its volume and duration.
Imagining keeping more detailed information about packets not measured because of flow state dependent sampling would contradict the fact that the sampling is done because of lack of memory and/or processing resources.

### 3.2 Policies/resource limitations in the flow recording process

This block is optional in the IPFIX framework architecture. However, we address here the case where it is present.
We already described in the previous section that because of lack of memory positions in the flow recording process some incoming packets may be discarded if they lead to the opening of a new flow record. However, under certain circumstances, it may be advantageous to discard an existing flow record in the flow recording process to

make room for the new record opened by an arriving packet. For example, an algorithm for taking the decision whether to discard the new arriving packet or an existing flow record is described in [Molina03]. Once again, we are not concerned here about the

algorithm details but about what information to store about this record removal.
For the same reasons expressed before, we argue that it does not make sense to store separate information for each discarded flow record, as it would contradict the motivation itself for which the discarding is done (lack of memory resources).
The information that is certainly possible to keep with a limited effort is a cumulative counter of the total number of not yet exported packets and bytes belonging to flow records that were eliminated from the flow recording process.
Ideally, we would also like to keep a timestamp for the first (T_fd) and last (T_ld) not yet exported packets belonging to all these discarded flow records. This would mean, in practice, to aggregate all these packets in a macro flow, and keep track of its volume and duration. To do so precisely, we would need to keep in each flow record a timestamp for the first and last non-exported packets, and whenever a record is discarded look at these timestamps to see if they are smaller or larger (respectively) of T_fd and T_ld and if yes update them. We further discuss in section 5 the requirement put on implementations by the information model described here.
Another information that can be easily kept is the number of these discarding events, along with a timestamp of the first and last of them. This information should not be used by applications to re-normalize their received per flow statistics (because a flow may be discarded and re-created multiple times) but rather to keep under control the good functioning of the implemented policy. Note that we consider a discarding event only when the discarded flow record contains some not exported traffic. Otherwise, the removal of a record whose traffic was fully exported (after a timeout or after the arrival of specific packets, e.g. TCP FIN or RST) is part of the normal functioning of an IPFIX flow metering system.
Note also that we consider only the case when an elimination of a flow record from the flow recording process leads to the complete loss of all the information contained in the flow record. If on the contrary another policy is implemented, like immediate exporting of the flow record before elimination, or freezing of the flow record and moving it in an area of memory different from which is considered the flow recording process for later exporting, this is not considered an elimination and therefore is out of the scope of this document.
In parallel to the information about the number of discarded flow records and associated packets and bytes, it is useful to keep cumulative information about the number of flow records containing not yet exported traffic that exist in the flow recording process, along with the cumulative number of not exported packets and bytes contained in them. This information is useful also for exporting process related reasons, as clarified in the following paragraph.

### 3.3 Policies/resource limitations in the exporting process

The exporting process may implement policies for not exporting the whole set of flow records of the flow recording process. In case of

absence of the flow recording process, when the metering process directly feeds the exporting process (i.e. directly compiles the export packets in IPFIX format), the following reasoning does not apply.

The motivation for not exporting some flow records (containing non exported traffic) can be two: there are explicit configured policies or the exporting process faces resource limitation.

An example of explicit policy can be not to export the flows whose accounted traffic is below a certain threshold, or a more complex mechanism such as the one described in [DuLuTh1] or [DuLuTh2]. An example of resource limitation is that the exporting process has an assigned, limited time slot to operate or a limited predefined number of export packets that it can send. There can also by hybrid cases where there are resource limitations and policies are applied in order to optimize the exported information (e.g. given that we want to export only N flow records, select a subset so that the overall number of reported packets and bytes belonging to the subset is maximized).

Coming to the issue of which information it makes sense to keep about this flow selection, there are two cases to consider.

If a flow is not exported and because of this decision is deleted from the flow recording process, we are in the same case described before (where the deletion was triggered by the need to make room for another record). The information to keep is then naturally the same as described before (cumulative packets and bytes for all the flows not exported, timestamps of the first and last packets belonging to non exported flow records, counter of dropping events and timestamp of first and last dropping event). Only the reason for this removal is different.

If on the contrary a record eligible for exporting is not exported but it remains in the flow recording process it has always a chance to be exported in the future. For an application, however, it would be beneficial to know what it is not currently being exported because of exporting process policies/resource limitations, in terms of flow records, packets and bytes. This, not to re-normalize its estimates (it would be dangerous and error prone because the exporting of these records may be simply delayed), but rather to keep under control whatÆs happening: for example, understand if there are pathologic situations where a large number of flow records and/or associated traffic are never exported, or if the number of flow records in the flow recording process is growing, etc.

When it comes to understanding if this information can be easily available, however, we recognize that there is the problem that in order to be aware that it has not exported a flow record, an exporting process should at least have browsed through it. In other words, we would have to assume that there is always a full scanning of the flow recording process associated to the exporting process selection decision. However, there may be more efficient

implementations where this does not happen. Therefore, even if we provide support in the information model for this information, defining it as mandatory in the protocol definition would put a

constraint on the exporting process implementation, which is undesirable.
However, the above information (i.e. what is in the flow recording process and has not been selected for exporting) can also be derived by an application by difference. An application knowing the count of the flow records in the flow recording process containing not yet exported traffic and the number of not exported packet and bytes belonging to them (we defined this info in the paragraph above), can in fact get it subtracting to these figures the number of flow records/packets/bytes it received.

## 4. Information model for flow selection

We formally define the elements to contain the information described in the previous section. Some elements have an associated couple of timestamps, which we reference for brevity (when it is not ambiguous) as Tfirst and Tlast (instead of element_nameTfirst, element_nameTlast).
Note that only packet or flow related counts have associated timestamps, while bytes related counts do not.
Note also that all the following information elements are aimed at describing macro flows (e.g. the total number of packets and bytes contained in all dropped flow records). Some of these macro flows are additive only, in the sense that they only add contributions to them, but never subtract. E.g. the macro flow of the packets contained in flow records that are discarded from the flow reporting process receives a contribution when a flow record is discarded, and this contribution can never be subtracted. On the contrary, some of the macro flows can dynamically receive and loose contributions. E.g. the macro flows of packets not yet exported receives a contribution when a new packets arrives, and looses some contribution when there is an exporting event. Associating a timestamp for the oldest and most recent contributions to additive only flow is easy, while for the others is not (would require to maintain full state) and that is why we did not define timestamps for these information elements.

### 4.1 Meter process related

#### 4.1.1 FsMeter_UnmeasPacketCount

Contains the count of packets that were not measured because of flow state dependent sampling

TsFirst: timestamp of the first packet not measured because of flow state dependent sampling

TsLast: timestamp of the last packet not measured because of flow

state dependent sampling

#### 4.1.2 FsMeter_UnmeasBytesCount

Contains the count of bytes that were not measured because of flow state dependent sampling

### 4.2 Flow recording process related

#### 4.2.1 FsFrec_PacketInDroppedRecsCount

Contains the count of non exported packets that were contained in flow records eliminated from the flow recording process because of resource limitations/policies in the flow recording process

TsFirst: timestamp of the first non-exported packet belonging to a eliminated flow record

TsLast: timestamp of the last non-exported packet belonging to a eliminated flow record

#### 4.2.2 FsFrec_ByteInDroppedRecsCount

Contains the count of non exported bytes that were contained in flow records eliminated from the flow recording process because of resource limitations/policies in the flow recording process

#### 4.2.3 FsFrec_FrecDroppedCount

Contains the count of flow records containing non exported packets eliminated from the flow recording process because of resources limitations/policies in the flow recording process

TsFirst: timestamp of the first flow record elimination event from the flow recording process

TsLast: timestamp of the last flow record elimination event from the flow recording process

#### 4.2.4 FsFrec_UnexportedFrecCount

Contains the count of the flow records currently existing in the flow recording process containing at least one non exported packet

#### 4.2.5 FsFrec_UnexportedPacketInFrecCount

Contains the count of non exported packets contained in flow records of the flow recording process

#### 4.2.6 FsFrec_UnexportedBytesInFrecCount

Contains the count of non exported bytes contained in flow records of

the flow recording process

### 4.3 Flow exporting process related

#### 4.3.1 FsExp_PacketInDroppedRecsCount

Contains the count of non exported packets that were contained in flow records eliminated from the flow recording process because of resource limitations/policies in the exporting process

TsFirst: timestamp of the first non exported packet belonging to a eliminated flow record

TsLast: timestamp of the last non exported packet belonging to a eliminated flow record

#### 4.3.2 FsExp_ByteInDroppedRecsCount

Contains the count of non exported bytes that were contained in flow records eliminated from the flow recording process because of resource limitations/policies in the exporting process

#### 4.3.3 FsExp_FrecDroppedCount

Contains the count of flow records containing non exported packets eliminated from the flow recording process because of resource limitations/policies in the exporting process

TsFirst: timestamp of the first flow record elimination event from the flow recording process

TsLast: timestamp of the last flow record elimination event from the flow recording process

#### 4.3.4 FsExp_UnexportedCount

Contains the count of the flow records currently existing in the flow recording process containing non-exported traffic and not being exported because of exporting process resource lmitations/policies

#### 4.3.5 FsExp_UnexportedPacketCount

Contains the count of non exported packets contained in flow records of the flow recording process not being exported because of exporting process resource limitations/policies

#### 4.3.6 FsFrec_UnexportedByteInFrecCount

Contains the count of non exported bytes contained in flow records of the flow recording process not being exported because of exporting process resource limitations/policies

## 4.4 Relationship between counts

As mentioned in 3.3, depending on the implementation of the exporting process it may be difficult to get reliable information about the number of flow records containing non-exported traffic and not exported because of policies/resource limitations in the exporting process.
However, the information elements defined in 4.3.4, 4.3.5, 4.3.6 can be also obtained by difference. For example, the number of exported flow records in the flow recording process and containing non exported traffic (4.2.4) plus the number of flow records deleted from the flow recording process when they still contained non exported traffic (4.2.3 and 4.3.3) minus the number of received flow records (not defined in this model) is equal to the number of flow records not being exported because of exporting process policies/resource limitations, i.e. 4.3.4. The same reasoning applies to non-exported packets and bytes.

## 5. Requirements put on implementations

To support the described information model an implementation must keep, in the flow records, counts for non-exported packets and bytes. Sometimes these are referred as delta counts. An implementation may also keep absolute counts for scopes not specified in this information model (it appears that both delta and absolute counters can be exported in the IPFIX information model, see [Cal03], par. 6.10).
In addition, to fully support this information model, it would be required to keep in a flow record a timestamp for the first and last non-exported packets. An implementation may need to keep timestamps for the first and last exported packets as well for scopes not specified in this information model, or to join the two timers for the last exported and first exported packets (which is of course an approximation) or to approximate them with the time of the exporting event.

## 6. Exporting of flow selection information

As it appears evident from the described information model, the flow selection information is not relative to a single flow, but rather to the behavior of a whole metering / flow recording / exporting process. This is the same category of information like the one already described in the IPFIX information model for packet sampling, see [Cal03], 6.23 and 6.24 (SamplingInterval and SamplingAlgorithm information elements).
As for packet sampling related information, the way to export it should be through the option records. In fact, [Claise03], sec. 9.1, states:

> ææThe Options Template Record (and its corresponding Options Data

Record) is used to supply information about the Metering Process configuration or Metering Process specific data, rather than supplying information about IP FlowsÆÆ

The Options template record can contain a scope field that specifies ([Claise03], sec. 9.1)

> ææThe relevant portion of the Exporting Process/Metering Process to which the Options Template Record refers. Currently defined values are: can be the interface, the cache, etc.ÆÆ

An open issue is to identify whether the currently defined scope types are enough for flow sampling purposes.

## 7. References


[Cal03] P.Calato, J.Meyer, J.Quittek: Information Model for IP Flow Information Export, Internet Draft <draft-ietf-ipfix-info-01.txt>, work in progress, August 2003

[ZeSamp03] T.Zseby, M.Molina, F.Raspall, N.Duffield: Sampling and Filtering Techniques for IP Packet Selection, Internet Draft < draft-ietf-psamp-sample-tech-02.txt>, work in progress, June 2003

[Sad03] G.Sadasivan, N.Bownlee: Architecture Model for IP Flow Information Export, Internet Draft <draft-ietf-ipfix-arch-01.txt>, work in progress, June 2003

[Claise03] B.Claise, M.Fullmer, P.Calato, R.Penno: IPFIX Protocol Specifications, Internet Draft, <draft-ietf-ipfix-protocol-00.txt>, work in progress, June 2003

[DuLuTh1] N. Duffield, C. Lund, M.Thorup: Properties and Prediction of Flow Statistics from Sampled Packet Streams, ACM SIGCOMM Internet Measurement Workshop 2002

[DuLuTh2] N. Duffield, C. Lund, M.Thorup: Learn More, Sample Less: Control of Volume and Variance in Network Measurement - http://www.research.att.com/~duffield/pubs/DLT02-optimal.pdf

[EsVa01] C. Estan and G. Varghese: New Directions in Traffic Measurement and Accounting, ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco (CA) Nov. 2001

[Molina03] M.Molina: A scalable and efficient methodology for flow monitoring in the internet, International Teletraffic Congress (ITC-18), Berlin, Sep. 2003


## 8. Author's Addresses


Maurizio Molina
NEC Europe Ltd., Network Laboratories

Kurfuersten-Anlage 36
69115 Heidelberg

Germany
Phone: +49 6221 90511-18
Email: molina@ccrle.nec.de

## 9. Full Copyright Statement