Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2013

S. D'Antonio
University of Napoli
"Parthenope"
T. Zseby
CAIDA/FhG FOKUS
C. Henke
Tektronix Communication Berlin
L. Peluso
University of Napoli

February 23, 2013

Flow Selection Techniques
draft-ietf-ipfix-flow-selection-tech-13.txt

Abstract

Intermediate Flow Selection Process is the process of selecting a subset of Flows from all observed Flows. The Intermediate Flow Selection Process may be located at an IPFIX Exporter, Collector, or within an IPFIX Mediator. It reduces the effort of post-processing Flow data and transferring Flow Records. This document describes motivations for using the Intermediate Flow Selection process and presents Intermediate Flow Selection techniques. It provides an information model for configuring Intermediate Flow Selection Process techniques and discusses what information about an Intermediate Flow Selection Process should be exported.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This Internet-Draft will expire on August 27, 2013.

Table of Contents

1. Scope

This document describes Intermediate Flow Selection Process techniques for network traffic measurements. A Flow is defined as a set of packets with common properties as described in [RFC5101]. An Intermediate Flow Selection Process can be executed to limit the resource demands for capturing, storing, exporting and post-processing of Flow Records. It also can be used to select a particular set of Flows that are of interest to a specific application. This document provides a categorization of Intermediate Flow Selection Process techniques and describes configuration and reporting parameters for them. In order to be compliant with this document, at least the Property Match Filtering MUST be implemented.

This document also addresses configuration and reporting parameters for Flow-state Dependent Packet Selection as described in [RFC5475], although this technique is categorized as packet selection. The reason is that Flow-state Dependent Packet Selection techniques often aim at the reduction of resources for Flow capturing and Flow processing. Furthermore, these techniques were only briefly discussed in [RFC5475]. Therefore configuration and reporting considerations for Flow-state Dependent Packet Selection techniques have been included in this document.

2. Terminology

This document is consistent with the terminology introduced in [RFC5101], [RFC5470], [RFC5475] and [RFC3917]. As in [RFC5101] and [RFC5476], the first letter of each IPFIX-specific and PSAMP-specific term is capitalized along with the Intermediate Flow Selection Process specific terms defined here.

* Packet Classification

  Packet Classification is a process by which packets are mapped to specific Flow Records based on packet properties or external properties (e.g. interface). The properties (e.g. header information, packet content, AS number) make up the Flow Key. In case a Flow Record for a specific Flow Key already exists the Flow Record is updated, otherwise a new Flow Record is created.

* Intermediate Flow Selection Process

  An Intermediate Flow Selection Process is an Intermediate Process as in [RFC6183] that takes Flow Records as its input and selects a subset of this set as its output. Intermediate Flow Selection Process is a more general concept than Intermediate Selection

Process as defined in [RFC6183]. While an Intermediate Selection Process selects Flow Records from a sequence based upon criteria-evaluated Flow record values and passes only those Flow Records that match the criteria, an Intermediate Flow Selection Process selects Flow Records using selection criteria applicable to a larger set of Flow characteristics and information.

* Flow Selection State

  An Intermediate Flow Selection Process maintains state information for use by the Flow Selector. At a given time, the Flow Selection State may depend on Flows and packets observed at and before that time, as well as other variables. Examples include:

  (i) sequence number of packets and accounted Flow Records;

  (ii) number of selected Flows;

  (iii) number of observed Flows;

  (iv) current Flow cache occupancy;

  (v) Flow specific counters, lower and upper bounds;

  (vi) Intermediate Flow Selection Process timeout intervals.

* Flow Selector

  A Flow Selector defines the action of an Intermediate Flow Selection Process on a single Flow of its input. The Flow Selector can make use of the following information in order to establish whether a Flow has to be selected or not:

  (i) the content of the Flow Record;

  (ii) any state information related to the Metering Process or Exporting Process;

  (iii) any Flow Selection State that may be maintained by the Intermediate Flow Selection Process.

* Complete Flow

  A Complete Flow consists of all the packets that enter the Intermediate Flow Selection Process within the Flow time-out interval, and which belong to the same Flow as defined by the Flow definition in [RFC5470]. For this definition only packets that arrive at the Intermediate Flow Selection Process are considered.

* Flow Filtering

  Flow Filtering selects flows based on a deterministic function on the Flow Record content, Flow Selection State, external properties (e.g. ingress interface) or external events (e.g violated Access Control List). If the relevant parts of the Flow Record content can already be observed at packet level (e.g. Flow Keys from packet header fields) Flow Filtering can be performed at packet level by Property Match Filtering as described in [RFC5475].

* Hash-based Flow Filtering

  Hash-based Flow Filtering is a deterministic Flow filter function that selects flows based on a Hash Function. The Hash Function is calculated over parts of the Flow Record content or external properties which are called the Hash Domain. If the hash value falls into a predefined Hash Selection Range the Flow is selected. Hash-based Flow Filtering can already applied at packet level, in which case the Hash Domain MUST contain the Flow Key of the packet. In case Hash-based Flow Filtering is used to select the same subset of flows at different Observation Points, the Hash Domain MUST comprise parts of the packet or Flow thar are invariant on the packet/Flow path. Also refer to the according Trajectory Sampling Application Example on packet level in [RFC5475]

* Flow-state Dependent Intermediate Flow Selection Process

  Flow-state Dependent Intermediate Flow Selection Process is a selection function that selects or drops Flows based on the current Flow Selection State. The selection can be either deterministic, random or non-uniform random.

* Flow-state Dependent Packet Selection

  Flow-state Dependent Packet Selection is a selection function that selects or drops packets based on the current Flow Selection State. The selection can be either deterministic, random or non-uniform random. Flow-state Dependent Packet Selection can be used to prefer the selection of packets belonging to specific Flows. For example the selection probability of packets belonging to Flows that are already within the Flow Cache may be higher than for packets that have not been recorded yet.

* Flow Sampling

Flow Sampling selects flows based on Flow Record sequence or arrival times (e.g. entry in Flow cache, arrival time at Exporter or Mediator). The selection can be systematic (e.g. every n-th Flow) or based on a random function (e.g. select each Flow Record with probability p, or randomly select n out of N Flow Records).

3. Difference between Intermediate Flow Selection Process and Packet Selection

Intermediate Flow Selection Process differs from packet selection described in [RFC5475]. Packet selection techniques consider packets as the basic element and the parent population consists of all packets observed at an Observation Point. In contrast to this the basic elements in Flow selection are the Flows. The parent population consists of all observed Flows and the Intermediate Flow Selection Process operates on the Flows. The major characteristics of Intermediate Flow Selection Process are the following:

- Intermediate Flow Selection Process takes Flows as basic elements. For packet selection, packets are considered as basic elements.

- Intermediate Flow Selection Process can only take place after Packet Classification, because the classification rules determine to which Flow a packet belongs. Packet selection can be applied before and after Packet Classification. As an example, packet selection before Packet Classification can be random packet selection whereas packet selection after Packet Classification can be Flow-state Dependent Packet Selection (as described in [RFC5475])

- Intermediate Flow Selection Process operates on Complete Flows. That means that after the Intermediate Flow Selection Process either all packets of the Flow are kept or all packets of the Flow are discarded. That means that if the Intermediate Flow Selection Process is preceded by a packet selection process the Complete Flow consists only of the packets that were not discarded during the packet selection.

There are some techniques that are difficult to unambiguously categorize into one of the categories. Here some guidance is given on how to categorize such techniques:

- Techniques that can be considered as both packet selection and Intermediate Flow Selection Process: some packet selection techniques result in the selection of Complete Flows and therefore can be considered as packet selection or

as Intermediate Flow Selection Process at the same time. An example is Property Match Filtering of all packets to a specific destination address. If Flows are defined based on destination addresses, such a packet selection also results in a Intermediate Flow Selection Process and can be considered as packet selection or Intermediate Flow Selection Process.

- Flow-state Dependent Packet Selection: there exist techniques that select packets based on the Flow state, e.g. based on the number of already observed packets belonging to the Flow. Examples of these techniques from the literature are "Sample and Hold" [EsVa01] "Fast Filtered Sampling" [MSZC10] or the "Sticky Sampling" algorithm presented in [MaMo02]. Such techniques can be used to influence which Flows are captured (e.g. increase the selection of packets belonging to large Flows) and reduce the number of Flows that need to be stored in the Flow cache. Nevertheless, such techniques do not necessarily select Complete Flows, because they do not ensure that all packets of a selected Flow are captured. Therefore Flow-state Dependent Packet Selection techniques that do not ensure that either all or no packets of a Flow are selected strictly speaking have to be considered as packet selection techniques and not as Intermediate Flow Selection Process techniques.

4. Intermediate Flow Selection Process within the IPFIX Architecture

An Intermediate Flow Selection Process can be deployed at any of three places within the IPFIX architecture. As shown in Figure 1 Intermediate Flow Selection Process can occur

1. in the Metering Process at the IPFIX Exporter

2. in the Exporting Process at the Collector

3. within a Mediator

```
+===========================================+
|  IPFIX Exporter        +----------------+ |
|                        | Metering Proc. | |
| +-----------------+    +----------------+ |
| |    Metering     |    |  Intermediate  | |
| |    Process      | or | Flow Selection | |
| |                 |    |     Process    | |
| +-----------------+----+----------------+ |
| |           Exporting Process           | |
| +----|-------------------------------|--+ |
+======|===============================|====+
       |                               |
       |                               |
+======|=======================+       |
|      |  Mediator              |      |
|    +-V-------------------+    |      |
|    | Collecting Process  |    |      |
|    +---------------------+    |      |
|    | Intermediate Flow   |    |      |
|    | Selection Process   |    |      |
|    +---------------------+    |      |
|    |  Exporting Process  |    |      |
|    +-|-------------------+    |      |
+======|=======================+       |
       |                               |
       |                               |
+======|===============================|=====+
|      |         Collector             |     |
| +----V-------------------------------V-+   |
| |         Collecting Process           |   |
| +--------------------------------------+   |
| | Intermediate Flow Selection Process  |   |
| +--------------------------------------+   |
| |           Exporting Process          |   |
| +------------------------------|-------+   |
+================================|===========+
                                 |
                                 |
                                 V
                          +------------------+
                          |      IPFIX       |
                          +------------------+
```

Figure 1: Potential Intermediate Flow Selection Process locations

In contrast to packet selection, Intermediate Flow Selection Process is always applied after the packets are classified into Flows.

4.1. Intermediate Flow Selection Process in the Metering Process

Intermediate Flow Selection Process in the Metering process uses packet information to update the Flow Records in the Flow cache. Intermediate Flow Selection Process before Packet Classification can be based on the fields of the Flow Key (also on a hash value over these fields), but not based on characteristics that are only available after Packet Classification (e.g. Flow size, Flow duration). An Intermediate Flow Selection Process is here applied to reduce resources for all succeeding processes or to select specific Flows of interest in case such Flow characteristics are already observable at packet level (e.g. Flows to specific IP addresses). In contrast, Flow-state Dependent Packet Selection is a packet selection technique, because it does not necessarily select Complete Flows.

4.2. Intermediate Flow Selection Process in the Exporting Process

Intermediate Flow Selection Process in the Exporting Process works on Flow Records. An Intermediate Flow Selection Process in the Exporting Process can therefore depend on Flow characteristics that are only visible after the classification of packets, such as Flow size and Flow duration. The Exporting Process may implement policies for exporting only a subset of the Flow Records which have been stored in the system memory in order to unload Flow export and Flow post-processing. An Intermediate Flow Selection Process in the Exporting Process may select only the subset of Flow Records which are of interest to the users application, or select only as many Flow Records as can be handled by the available resources (e.g. limited export link capacity).

4.3. Intermediate Flow Selection Process as a function of the IPFIX Mediator

As shown in Figure 1, Intermediate Flow Selection Process can be performed within an IPFIX Mediator [RFC6183]. The Intermediate Flow Selection Process takes Flow Record stream as its input and selects Flow Records from a sequence based upon criteria-evaluated record values. The Intermediate Flow Selection Process can again apply an Intermediate Flow Selection Process technique to obtain Flows of interest to the application. Further, the Intermediate Flow Selection Process can base its selection decision on the correlation of data from different IPFIX Exporters, e.g. by only selecting Flows that were at least recorded on two IPFIX Exporters.

5. Intermediate Flow Selection Process Techniques

An Intermediate Flow Selection Process technique selects either all or none of the packets of a Flow, otherwise the technique has to be considered as packet selection. A difference is recognized between Flow Filtering and Flow Sampling.

5.1. Flow Filtering

Flow Filtering is a deterministic function on the IPFIX Flow Record content. If the relevant Flow characteristics are already observable at packet level (e.g. Flow Keys), Flow Filtering can be applied before aggregation at packet level. In order to be compliant with this document, at least the Property Match Filtering MUST be implemented.

5.1.1. Property Match Filtering

Property Match Filtering can be performed similarly to Property Match Filtering for packet selection described in [RFC5475]. The difference is that, instead of packet fields, Flow Record fields are here used to derive the selection decision. Property Match Filtering is typically used to select a specific subset of the Flows that are of interest to a particular application (e.g. all Flows to a specific destination, all large Flows, etc.). Properties on which the filtering is based can be Flow Keys, Flow Timestamps, or Per-Flow Counters described in [RFC5102]. Examples of properties are the Flow size in bytes, the number of packets in the Flow, the observation time of the first or last packet, or the maximum packet length. An example is to select Flows with more than a threshold number of observed octets. The selection criteria can be a specific value, a set of specific values, or an interval. For example, a Flow is selected if destinationIPv4Address and the total number of packets of the Flow equal two predefined values. Property Match Filtering can be applied in the Metering Process if the properties are already observable at the packet level (e.g. Flow Key fields). For example, a Flow is selected if sourceIPv4Address and sourceIPv4PrefixLength equal, respectively, two specific values.

There are content-based Property Match Filtering techniques that require a computation on the current Flow cache. An example is the selection of the largest Flows or a percentage of Flows with the longest lifetime. This type of Property Match Filtering is also used in Intermediate Flow Selection Proces techniques that react to external events (e.g. resource constraint). For example when the Flow cache is full, the Flow Record with the lowest Flow volume per current Flow life time may be deleted.

5.1.2. Hash-based Flow Filtering

Hash-based Flow Filtering uses a Hash Function h to map the Flow Key c onto a Hash Range R. A Flow is selected if the hash value h(c) is within the Hash Selection Range S, which is a subset of R. Hash-based Flow Filtering can be used to emulate a random sampling process but still enable the correlation between selected Flow subsets at different Observation Points. Hash-based Flow Filtering is similar to Hash-based Packet Selection, and in fact is identical when Hash-based Packet Selection uses the Flow Key that defines the Flow as the hash input. Nevertheless there may be the incentive to apply Hash-based Flow Filtering not on the packet level in the Metering Process, for example when the size of the selection range and therefore the sampling probability is dependent on the number of observed Flows.

5.2. Flow Sampling

Flow Sampling operates on Flow Record sequence or arrival times. It can use either a systematic or a random function for the Intermediate Flow Selection Process. Flow Sampling usually aims at the selection of a representative subset of all Flows in order to estimate characteristics of the whole set (e.g. mean Flow size in the network).

5.2.1. Systematic sampling

Systematic sampling is a deterministic selection function. Systematic sampling may be a periodic selection of the N-th Flow Record which arrives at the Intermediate Flow Selection Process. Systematic sampling MAY be applied in the Metering Process. An example would be to create, besides the Flow cache of selected Flows, an additional data structure that saves the Flow Keys of the Flows that are not selected. The selection of a Flow would then be based on the first packet of a Flow. Everytime a packet belonging to a new Flow (which is neither in the data structure of the selected or not selected Flows) arrives at the Observation Point, a counter is increased. In case the counter is increased to a multiple of N a new Flow cache entry is created, and in case the counter is not a multiple of N the Flow Key is added to the data structure for not selected Flows.

Systematic sampling can also be time-based. Time-based systematic sampling is applied by only creating Flows that are observed between time-based start and stop triggers. The time interval may be applied at packet level in the Metering Process or after aggregation on Flow level, e.g. by selecting a Flow arriving at the Exporting Process every n seconds.

5.2.2. Random Sampling

Random Flow sampling is based on a random process which requires the calculation of random numbers. One can differentiate between n-out-N and probabilistic Flow sampling.

5.2.2.1. n-out-of-N Flow Sampling

In n-out-of-N Sampling, n elements are selected out of the parent population that consists of N elements. One example would be to generate n different random numbers in the range [1,N] and select all Flows that have a Flow position equal to one of the random numbers.

5.2.2.2. Probabilistic Flow Sampling

In probabilistic Sampling, the decision whether or not a Flow is selected is made in accordance with a predefined selection probability. For probabilistic Sampling, the Sample Size can vary for different trials. The selection probability does not necessarily have to be the same for each Flow. Therefore, a difference is recognized between uniform probabilistic sampling (with the same selection probability for all Flows) and non-uniform probabilistic sampling (where the selection probability can vary for different Flows). For non-uniform probabilistic Flow Sampling the sampling probability may be adjusted according to the Flow Record content. An example would be to increase the selection probability of large volume Flows over small volume Flows as described in the Smart Sampling technique [DuLT01].

5.3. Flow-state Dependent Intermediate Flow Selection Process

Flow-state Dependent Intermediate Flow Selection Process can be a deterministic or random Intermediate Flow Selection Process based on the Flow Record content and the Flow state which may be kept additionally for each of the Flows. External processes may update counters, bounds and timers for each of the Flow Records and the Intermediate Flow Selection Process utilises this information for the selection decision. A review of Flow-state Dependent Intermediate Flow Selection Process techniques that aim at the selection of the most frequent items by keeping additional Flow state information can be found in [CoHa08]. Flow-state Dependent Intermediate Flow Selection Process can only be applied after packet aggregation, when a packet has been assigned to a Flow. The Intermediate Flow Selection Process then decides based upon the Flow state for each Flow if it is kept in the Flow cache or not. Two Flow-state Dependent Intermediate Flow Selection Process Algorithms are here described:

The frequent algorithm [KaPS03] is a technique that aims at the selection of all flows that at least exceed a 1/k fraction of the Observed Packet Stream. The algorithm has only a Flow cache of size k-1 and each Flow in the cache has an additional counter. The counter is incremented each time a packet belonging to the Flow in the Flow cache is observed. In case the observed packet does not belong to any Flow all counters are decremented and if any of the Flow counters has a value of zero the Flow is replaced with a Flow formed from the new packet.

Lossy counting is a selection technique that identifies all Flows whose packet count exceeds a certain percentage of the whole observed packet stream (e.g. 5% of all packets) with a certain estimation error e. Lossy counting separates the observed packet stream in windows of size N=1/e, where N is an amount of consecutive packets. For each observed Flow an additional counter will be held in the Flow state. The counter is incremented each time a packet belonging to the Flow is observed and all counters are decremented at the end of each window and all Flows with a counter of zero are removed from the Flow cache.

5.4. Flow-state Dependent Packet Selection

Flow-state Dependent Packet Selection is not an Intermediate Flow Selection Process technique but a packet selection technique. Nevertheless configuration and reporting parameters for this technique will be described in this document. An example is the "Sample and Hold" algorithm [EsVa01] that tries to prefer large volume Flows in the selection. When a packet arrives it is selected when a Flow Record for this packet already exists. In case there is no Flow Record, the packet is selected by a certain probability that is dependent on the packet size.

6. Configuration of Intermediate Flow Selection Process Techniques

This section describes the configuration parameters of the Flow selection techniques presented above. It provides the basis for an information model to be adopted in order to configure the Intermediate Flow Selection Process within an IPFIX Device. The actual information model with the Information Elements (IEs) for the configuration is described together with the reporting IEs in section 7. The following table gives an overview of the defined Intermediate Flow Selection Process techniques, where they can be applied and what their input parameters are. Depending on where the Flow selection techniques are applied different input parameters can be configured.

Overview of Intermediate Flow Selection Process Techniques:

Location	Selection Technique	Selection Input
In the Metering Process	Flow-state Dependent Packet Selection	packet sampling probabilities, Flow Selection State, packet properties
In the Metering Process	Property Match Flow Filtering	Flow record IEs, Selection Interval
In the Metering Process	Hash-based Flow Filtering	selection range, Hash Function, Flow Key, (seed)
In the Metering Process	Time-based Systematic Flow Sampling	Flow position (derived from arrival time of packets), Flow Selection State
In the Metering Process	Sequence-based Systematic Flow Sampling	Flow position (derived from packet position), Flow Selection State
In the Metering Process	Random Flow Sampling	random number generator or list and packet position, Flow state
In the Exporting Process/ within the IPFIX Mediator	Property Match Flow Filtering	Flow Record content, filter function
In the Exporting Process/ within the IPFIX Mediator	Hash-based Flow Filtering	selection range, Hash Function, hash input (Flow Keys and other Flow properties)
In the Exporting Process/ within the IPFIX Mediator	Flow-state Dependent Intermediate Flow Selection Process	Flow state parameters, random number generator or list
In the Exporting Process/ within the IPFIX Mediator	Time-based Systematic Flow Sampling	Flow arrival time, Flow state

In the Exporting Process/ within the IPFIX Mediator	Sequence-based Systematic Flow Sampling	Flow position, Flow state
In the Exporting Process/ within the IPFIX Mediator	Random Flow Sampling	random number generator or list and Flow position, Flow state

Table 1: Overview of Intermediate Flow Selection Process Techniques

6.1. Intermediate Flow Selection Process Parameters

This section defines what parameters are required to describe the most common Intermediate Flow Selection Process techniques.

Intermediate Flow Selection Process Parameters:

For Property Match Filtering:

- Information Element as specified in [iana-ipfix-assignments]): Specifies the Information Element which is used as the property in the filter expression.

- Selection Value or Value Interval: Specifies the value or interval of the filter expression. Packets and Flow Records that have a value equal to the Selection Value or within the Interval will be selected.

For Hash-based Flow Filtering:

- Hash Domain: Specifies the bits from the packet or Flow which are taken as the hash input to the Hash Function.

- Hash Function: Specifies the name of the Hash Function that is used to calculate the hash value. Possible Hash Functions are BOB [RFC5475], IPSX [RFC5475], CRC-32 [Bra75]

- Hash Selection Range: Flows that have a hash value within the Hash Selection Range are selected. The Hash Selection Range can be a value interval or arbitrary hash values within the Hash Range of the Hash Function.

- Random Seed or Initializer Value:
  Some Hash Functions require an initializing value. In order to make the selection decision more secure one can choose a random seed that configures the hash function.

For Flow-state Dependent Intermediate Flow Selection Process:

- frequency threshold:
  Specifies the frequency threshold s for Flow-state Dependent Flow Selection techniques that try to find the most frequent items within a dataset. All Flows which exceed the defined threshold will be selected.

- accuracy parameter:
  specifies the accuracy parameter e for techniques that deal with the frequent items problems. The accuracy parameter defines the maximum error, i.e. no Flows that have a true frequency less than (s - e) N are selected, where s is the frequency threshold and N is the total number of packets.

The above list of parameters for Flow-state Dependent Flow Selection techniques is suitable for the presented frequent item and lossy counting algorithms. Nevertheless a variety of techniques exist with very specific parameters which are not defined here.

For Systematic time-based Flow Sampling:

- Interval length (in usec)
  Defines the length of the sampling interval during which Flows are selected.

- Spacing (in usec)
  The spacing parameter defines the spacing in usec between the end of one sampling interval and the start of the next succeeding interval.

For Systematic count-based Flow Sampling:

- Interval length
  Defines the number of Flows that are selected within the sampling interval.

- Spacing
  The spacing parameter defines the spacing in number of observed Flows between the end of one sampling interval and the start of the next succeeding interval.

For random n-out-of-N Flow Sampling:

- Population Size N
  The Population Size N is the number of all Flows in the Population from which the sample is drawn.

- Sampling Size n
  The sampling size n is the number of Flows that are randomly drawn from the population N.

For probabilistic Flow Sampling:

- Sampling probability p
  The sampling probability p defines the probability by which each of the observed Flows is selected.

6.2. Description of Flow-state Dependent Packet Selection

The configuration of Flow-state Dependent Packet Selection has not been described in [RFC5475] therefore the parameters are defined here:

For Flow-state Dependent Packet Selection:

- packet selection probability per possible Flow state interval
  Defines multiple {Flow interval, packet selection probability} value pairs that configure the sampling probability depending on the current Flow state.

- additional parameters
  For the configuration of Flow-state Dependent Packet Selection additional parameters or packet properties may be required, e.g. the packet size ([EsVa01])

7. Information Model for Intermediate Flow Selection Process Configuration and Reporting

This section specifies the Information Elements (IEs) that MUST be exported by an Intermediate Flow Selection Process in order to support the interpretation of measurement results from Flow measurements. The information is mainly used to report how many packets and Flows have been observed in total and how many of them were selected. This helps for instance to calculate the Attained Selection Fraction (see also [RFC5476]), which is an important parameter to provide an accuracy statement. The IEs can provide reporting information about Flow Records, packets or bytes. The reported metrics are total number of elements and the number of selected elements. From this the number of dropped elements can be derived.

List of Intermediate Flow Selection Process Information Elements:

ID	Name	ID	Name
301	selectionSequenceID	302	selectorID
TBD1	flowSelectorAlgorithm	1	octetDeltaCount
TBD2	flowSelectedOctetDeltaCount	2	packetDeltaCount
TBD3	flowSelectedPacketDeltaCount	3	originalFlowsPresent
TBD4	flowSelectedFlowDeltaCount	TBD5	selectorIDTotalFlowsObserved
TBD6	selectorIDTotalFlowsSelected	TBD7	samplingFlowInterval
TBD8	samplingFlowSpacing	309	samplingSize
310	samplingPopulation	311	samplingProbability
TBD9	flowSamplingTimeInterval	TBD10	flowSamplingTimeSpacing
326	digestHashValue	TBD11	hashFlowDomain
329	hashOutputRangeMin	330	hashOutputRangeMax
331	hashSelectedRangeMin	332	hashSelectedRangeMax
333	hashDigestOutput	334	hashInitialiserValue
320	absoluteError	321	relativeError
336	upperCILimit	337	lowerCILimit
338	confidenceLevel		

Table 2: Intermediate Flow Selection Process Information Elements

7.1. flowSelectorAlgorithm

Description:

This Information Element identifies the Intermediate Flow Selection Process technique(e.g., Filtering, Sampling) that is applied by the Intermediate Flow Selection Process. Most of these techniques have parameters as decribed in Section 6. Further technique identifiers may be added to the list below. It might be necessary to define new Information Elements to specify their parameters. The flowSelectorAlgorithm registry is maintained by IANA. New assignments for the registry will be administered by IANA and are subject to Expert Review [RFC5226]. The registry can be updated when specifications of the new technique(s) and any new Information Elements are provided.

ID	Technique	Parameters
1	Systematic count-based Sampling	flowSamplingInterval flowSamplingSpacing
2	Systematic time-based Sampling	flowSamplingTimeInterval flowSamplingTimeSpacing
3	Random n-out-of-N Sampling	samplingSize samplingPopulation
4	Uniform probabilistic Sampling	samplingProbability
5	Property Match Filtering	Information Element Value Range
	Hash-based Filtering	hashInitialiserValue hashFlowDomain hashSelectedRangeMin hashSelectedRangeMax hashOutputRangeMin hashOutputRangeMax
6	using BOB	
7	using IPSX	
8	using CRC	
TBDx	Flow-state Dependent Intermediate Flow Selection Process	No agreed Parameters

Intermediate Flow Selection Process Techniques

Abstract Data Type: unsigned16

ElementId: TBD1

Data Type Semantics: identifier

Status: Proposed

7.2. flowSelectedOctetDeltaCount

Description:

This Information Element specifies the volume in octets of all Flows that are selected in the Intermediate Flow Selection Process since the previous report.

Abstract Data Type: unsigned64

ElementId: TBD2

Units: Octets

Status: Proposed

7.3. flowSelectedPacketDeltaCount

Description:

This Information Element specifies the volume in packets of all Flows that were selected in the Intermediate Flow Selection Process since the previous report.

Abstract Data Type: unsigned64

ElementId: TBD3

Units: Packets

Status: Proposed

7.4. flowSelectedFlowDeltaCount

Description:

This Information Element specifies the number of Flows that were selected in the Intermediate Flow Selection Process since the last

report.

Abstract Data Type: unsigned64

ElementId: TBD4

Units: Flows

Status: Proposed

7.5. selectorIDTotalFlowsObserved

Description:

This Information Element specifies the total number of Flows observed by a Selector, for a specific value of SelectorId. This Information Element should be used in an Options Template scoped to the observation to which it refers. See Section 3.4.2.1 of the IPFIX protocol document [RFC5101] .

Abstract Data Type: unsigned64

ElementId: TBD5

Units: Flows

Status: Proposed

7.6. selectorIDTotalFlowsSelected

Description:

This Information Element specifies the total number of Flows selected by a Selector, for a specific value of SelectorId. This Information Element should be used in an Options Template scoped to the observation to which it refers. See Section 3.4.2.1 of the IPFIX protocol document [RFC5101].

Abstract Data Type: unsigned64

ElementId: TBD6

Units: Flows

Status: Proposed

7.7. samplingFlowInterval

Description:

This Information Element specifies the number of Flows that are consecutively sampled. A value of 100 means that 100 consecutive Flows are sampled. For example, this Information Element may be used to describe the configuration of a systematic count-based Sampling Selector.

Abstract Data Type: unsigned64

ElementId: TBD7

Units: Flows

Status: Proposed

7.8. samplingFlowSpacing

Description:

This Information Element specifies the number of Flows between two "samplingFlowInterval"s. A value of 100 means that the next interval starts 100 Flows (which are not sampled) after the current "samplingFlowInterval" is over. For example, this Information Element may be used to describe the configuration of a systematic count-based Sampling Selector.

Abstract Data Type: unsigned64

ElementId: TBD8

Units: Flows

Status: Proposed

7.9. flowSamplingTimeInterval

Description:

This Information Element specifies the time interval in microseconds during which all arriving Flows are sampled. For example, this Information Element may be used to describe the configuration of a systematic time-based Sampling Selector.

Abstract Data Type: unsigned64

ElementId: TBD9

Units: microseconds

Status: Proposed

7.10. flowSamplingTimeSpacing

Description:

This Information Element specifies the time interval in microseconds between two "flowSamplingTimeInterval"s. A value of 100 means that the next interval starts 100 microseconds (during which no Flows are sampled) after the current "flowsamplingTimeInterval" is over. For example, this Information Element may used to describe the configuration of a systematic time-based Sampling Selector.

Abstract Data Type: unsigned64

ElementId: TBD10

Units: microseconds

Status: Proposed

7.11. hashFlowDomain

Description:

This Information Element specifies the Information Elements that are used by the Hash-based Flow Selector as the Hash Domain.

Abstract Data Type: unsigned16

ElementId: TBD11

Data Type Semantics: identifier

Status: Proposed

8. IANA Considerations

8.1. Registration of Information Elements

IANA will register the following IEs in the IPFIX Information Elements registry at http://www.iana.org/assignments/ipfix/ipfix.xml

Value	Name	Data Type	Data Type Semantics	Status	Description
TBD1	flowSelectorAlgorithm	unsigned16	identifier	Proposed	This Information Element identifies the Intermediate Flow Selection Process technique(e.g., Filtering, Sampling) that is applied by the Intermediate Flow Selection Process
TBD2	flowSelectedOctetDeltaCount	unsigned64	Octets	Proposed	This Information Element specifies the volume in octets of all Flows that are selected in the Intermediate Flow Selection Process since the previous report.

TBD 3	flowSelectedPacketDeltaCount	unsigned64	Packets	Proposed	This Information Element specifies the volume in packets of all Flows that were selected in the Intermediate Flow Selection Process since the previous report.
TBD 4	flowSelectedFlowDeltaCount	unsigned64	Flows	Proposed	This Information Element specifies the number of Flows that were selected in the Intermediate Flow Selection Process since the last report.

TBD 5	selectorIDTotalFlowsObserved	unsigned64	Flows	Proposed	This Information Element specifies the total number of Flows observed by a Selector, for a specific value of SelectorId. This Information Element should be used in an Options Template scoped to the observation to which it refers. See Section 3.4.2.1 of the IPFIX protocol document [RFC5101]

TBD 6	selectorIDTotalFlowsSelected	unsigned64	Flows	Proposed	This Information Element specifies the total number of Flows selected by a Selector, for a specific value of SelectorId. This Information Element should be used in an Options Template scoped to the observation to which it refers. See Section 3.4.2.1 of the IPFIX protocol document [RFC5101].

TBD7	samplingFlowInterval	unsigned64	Flows	Proposed	This Information Element specifies the number of Flows that are consecutively sampled. A value of 100 means that 100 consecutive Flows are sampled. For example, this Information Element may be used to describe the configuration of a systematic count-based Sampling Selector.

TBD8	samplingFlowSpacing	unsigned64	Flows	Proposed	This Information Element specifies the number of Flows between two "samplingFlowInterval"s. A value of 100 means that the next interval starts 100 Flows (which are not sampled) after the current "samplingFlowInterval" is over.For example, this Information Element may be used to describe the configuration of a systematiccount-based Sampling Selector.

TBD 9	flowSamplingTimeInterval	unsigned64	microseconds	Proposed	This Information Element specifies the time interval in microseconds during which all arriving Flows are sampled. For example, this Information Element may be used to describe the configuration of a systematic time-based Sampling Selector.

TBD 10	flowSamplingTimeSpacing	unsigned64	microseconds	Proposed	This Information Element specifies the time interval in microseconds between two "flowSamplingTimeInterval"s. Avalue of 100 means that the next interval starts 100 microseconds (during which no Flows are sampled) after the current "flowsamplingTimeInterval" is over. For example, this Information Element may used to describe the configuration of a systematictime-based Sampling Selector.
TBD 11	hashFlowDomain	unsigned16	identifier	Proposed	This Information Element specifies the Information Elements that are used by the Hash-based Flow Selector as the Hash Domain.

Table 3: Information Elements to be registered

IANA Note: please replace TBD1, TBD2, TBD3, TBD4, TBD5, TBD6, TBD7,

TBD8, TBD9, TBD10, TBD11 with the assigned values, throughout the document

8.2. Registration of Object Identifier

IANA will register the following OID in the IPFIX-SELECTOR-MIB Functions sub-registry at http://www.iana.org/assignments/smi-numbers according to the procedures set forth in [RFC6615]

Decimal	Name	Description	Reference
	flowSelectorAlgorithm	This Object Identifier identifies the Intermediate Flow Selection Process technique (e.g., Filtering, Sampling) that is applied by the Intermediate Flow Selection Process	TBDx [RFCyyyy]

Table 4: Object Identifiers to be registered

IANA Note: please replace TBDx with the assigned value, throughout the document.

Editor's Note (to be removed prior to publication): the RFC editor is asked to replace "yyyy" in this document by the number of the RFC when the assignment has been made.

9. Security Considerations

Some of the described Intermediate Flow Selection Process techniques (e.g., flow sampling, hash-based flow filtering) aim at the selection of a representative subset of flows in order to estimate parameters of the population. An adversary may have incentives to influence the selection of flows, for example to circumvent accounting or to avoid the detection of packets that are part of an attack.

Security considerations concerning the choice of a Hash Function for Hash-based Packet Selection have been discussed in Section 6.2.3 of [RFC5475] and are also appropriate for Hash-based Flow Selection. [RFC5475] discusses the possibility to craft Packet Streams which are

disproportionately selected or can be used to discover Hash Function parameters. It also describes vulnerabilities of different Hash Functions to these attacks, and practices to minimize these vulnerabilities.

For other sampling approaches a user can gain knowledge about the start and stop triggers in time-based systematic Sampling, e.g., by sending test packets. This knowledge might allow users to modify their send schedule in a way that their packets are disproportionately selected or not selected. For random Sampling, an input to the encryption process, like the Initialization Vector of the CBC (Cipher Block Chaining) mode, should be used to prevent that an advisory can predict the selection decision [Dw01].

Further security threats can occur when Intermediate Flow Selection Process parameters are configured or communicated to other entities. The protocol(s) for the configuration and reporting of Intermediate Flow Selection Process parameters are out of scope of this document. Nevertheless, a set of initial requirements for future configuration and reporting protocols are stated below:

1. Protection against disclosure of configuration information: Intermediate Flow Selection Process configuration information describes the Intermediate Flow Selection Process and its parameters. This information can be useful to attackers. Attackers may craft packets that never fit the selection criteria in order to prevent Flows to be seen by the Intermediate Flow Selection Process. They can also craft a lot of packets that fit the selection criteria and overload or bias subsequent processes. Therefore any transmission of configuration data (e.g., to configure a process or to report its actual status) should be protected by encryption.

2. Protection against modification of configuration information: if wrong configuration information is sent to the Intermediate Flow Selection Process, it can lead to a malfunction of the Intermediate Flow Selection Process. Also if wrong configuration information is reported from the Intermediate Flow Selection Process to other processes it can lead to wrong estimations at subsequent processes. Therefore any protocol that transmits configuration information should prevent that an attacker can modify configuration information. Data integrity can be achieved by authenticating the data.

3. Protection against malicious nodes sending configuration information: The remote configuration of Intermediate Flow Selection Process techniques should be protected against access by unauthorized nodes. This can be achieved by access control

lists at the device that hosts the Intermediate Flow Selection Process (e.g. IPFIX Exporter, IPFIX Mediator or IPFIX Collector) and by source authentication. The reporting of configuration data from an Intermediate Flow Selection Process has to be protected in the same way. That means that also protocols that report configuration data from the Intermediate Flow Selection Process to other processes need to protect against unauthorized nodes reporting configuration information.

The security threats that originate from communicating configuration information to and from Intermediate Flow Selection Processes cannot be assessed solely with the information given in this document. A further more detailed assessment of security threats is necessary when a specific protocol for the configuration or reporting configuration data is proposed.

10. Acknowledgments

We would like to thank the IPFIX group, especially Brian Trammell, Paul Aitken and Benoit Claise for fruitful discussions and for proofreading the document.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, March 2009.

[RFC5476] Claise, B., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[RFC6615] Dietz, T., Kobayashi, A., Claise, B., and G. Muenz, "Definitions of Managed Objects for IP Flow Information

Export", RFC 6615, June 2012.

11.2. Informative References

[Bra75] Brayer, K., "Evaluation of 32 Degree Polynomials in Error Detection on the SATIN IV Autovon Error Patterns", National Technical Information Service p.74, August 1975.

[CoHa08] Cormode, G. and M. Hadjieleftheriou, "Finding frequent items in data streams", Journal, Proceedings of the Very Large DataBase Endowment VLDB Endowment, Volume 1 Issue 2, August 2008, August 2008.

[DuLT01] Duffield, N., Lund, C., and M. Thorup, "Charging from Sampled Network Usage", ACM Internet Measurement Workshop IMW 2001, San Francisco, USA, November 2001.

[Dw01] Dworkin, M., "Recommendation for Block Cipher Modes of Operation - Methods and Techniques", NIST Special Publication NIST Special Publication 800-38A 2001 Edition, December 2001.

[EsVa01] Estan, C. and G,. Varghese, "New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice", ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco (CA), November 2001.

[KaPS03] Karp, R., Papadimitriou, C., and S. S. Shenker, "A simple algorithm for finding frequent elements in sets and bags.", ACM Transactions on Database Systems, Volume 28, 51-55, 2003, March 2003.

[MSZC10] Mai, J., Sridharan, A., Zang, H., and C. Chuah, "Fast Filtered Sampling", Computer Networks Volume 54, Issue 11, Pages 1885-1898, ISSN 1389-1286, January 2010.

[MaMo02] Manku, G. and R. Motwani, "Approximate Frequency Counts over Data Streams", Proceedings of the International Conference on Very large DataBases (VLDB) pages 346--357, 2002, Hong Kong, China, 2002.

[RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.

[RFC6183] Kobayashi, A., Claise, B., Muenz, G., and K. Ishibashi, "IP Flow Information Export (IPFIX) Mediation: Framework", RFC 6183, April 2011.

[iana-ipfix-assignments] "IP Flow Information Export Information Elements", 2007, <http://www.iana.org/assignments/ipfix/ipfix.xml>.

Appendix A. Appendix A. XML Specification of Intermediate Flow Selection Process Information Elements

This appendix contains a machine-readable description of the Intermediate Flow Selection Process Information Elements coded in XML. Note that this appendix is of informational nature, while the text in Section 7 is normative. The format in which this specification is given is described by the XML Schema in Appendix B of [RFC5102].


```
                        <fieldDefinitions xmlns="urn:ietf:params:xml:ns:ipfix-
info"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="urn:ietf:params:xml:ns:ipfix-info
                ipfix-info.xsd">

                 <field name="flowSelectorAlgorithm" dataType="unsigned16"
            dataTypeSemantics="identifier"
            elementId="TBD1" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element identifies the
Intermediate Flow Selection Process technique(e.g., Filtering, Sampling) that
is applied by the Intermediate Flow Selection Process. Most of these techniques
have parameters as decribed in <xref target="config"/>. Further technique
identifiers may be added to the list below. It might be necessary to define new
Information Elements to specify their parameters. The flowSelectorAlgorithm
registry is maintained by IANA. New assignments for the registry will be
administered by IANA and are subject to Expert Review <xref target="RFC5226"/>.
The registry can be updated when specifications of the new technique(s) and any
new Information
                                Elements are provided.
                                </paragraph>
                                <artwork>
                                +----+-----------------------
```

```
+-------------------------+
                                | ID |        Technique         |
Parameters          |
                                +----+-----------------------
+-------------------------+
                                | 1  | Systematic count-based |
flowSamplingInterval     |
                                |    | Sampling               |
flowSamplingSpacing      |
                                +----+-----------------------
+-------------------------+
                                | 2  | Systematic time-based  |
flowSamplingTimeInterval |
                                |    | Sampling               |
flowSamplingTimeSpacing  |
                                +----+-----------------------
+-------------------------+
                                | 3  | Random n-out-of-N      |
samplingSize             |
```

```
                                |    | Sampling              |
samplingPopulation       |
                                +----+-----------------------
+-------------------------+
                                | 4  | Uniform probabilistic |
samplingProbability      |
                                |    | Sampling
|                          |
                                +----+-----------------------
+-------------------------+
                                | 5  | Property Match        | Information
Element      |
                                |    | Filtering             | Value
Range              |
                                +----+-----------------------
+-------------------------+
                                |   Hash-based Filtering     |
hashInitialiserValue     |
                                +----+-----------------------+
hashFlowDomain           |
                                | 6  | using BOB             |
hashSelectedRangeMin     |
                                +----+-----------------------+
hashSelectedRangeMax     |
                                | 7  | using IPSX            |
hashOutputRangeMin       |
                                +----+-----------------------+
hashOutputRangeMax       |
                                | 8  | using CRC
|                          |
                                +----+-----------------------
+-------------------------+
                                |TBDx| Flow-state Dependent  | No agreed
Parameters     |
                                |    | Intermediate Flow
|                          |
                                |    | Selection Process
|                          |
                                +----+-----------------------
+-------------------------+
                                </artwork>
                        </description>
                </field>

                <field name="flowSelectedOctetDeltaCount" dataType="unsigned64"
                        elementId="TBD2" status="proposed">
                        <description>
                                <paragraph>
```

```
                                This Information Element specifies the volume
in octets of all
                                Flows that are selected in the Intermediate
Flow Selection Process
                                since the previous report.
                                </paragraph>
                        </description>
                        <units>octets</units>
                </field>

                <field name="flowSelectedPacketDeltaCount"
dataType="unsigned64"
         elementId="TBD3" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the volume
in packets of all
                                Flows that were selected in the Intermediate
Flow Selection
                                Process since the previous report.
                                </paragraph>
                        </description>
                        <units>packets</units>
                </field>
```

```
                <field name="flowSelectedFlowDeltaCount" dataType="unsigned64"
                        elementId="TBD4" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the number
of Flows that were
                                selected in the Intermediate Flow Selection
Process since the last
                                report.
                                </paragraph>
                        </description>
                        <units>flows</units>
                </field>

                <field name="selectorIDTotalFlowsObserved"
dataType="unsigned64"
                        elementId="TBD5" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the total
number of Flows
                                observed by a Selector, for a specific value of
SelectorId. This
                                Information Element should be used in an
Options Template scoped
                                to the observation to which it refers.
                                </paragraph>
                        </description>
                        <reference>
                                <paragraph>
                                See Section 3.4.2.1 of the IPFIX protocol
document <xref target="RFC5101"/>
                                </paragraph>
                        </reference>
                        <units>flows</units>
                </field>

                <field name="selectorIDTotalFlowsSelected"
dataType="unsigned64"
                        elementId="TBD6" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the total
number of Flows
                                selected by a Selector, for a specific value of
SelectorId. This
                                Information Element should be used in an
Options Template scoped
```

```
                                to the observation to which it refers.
                                </paragraph>
                        </description>
                        <reference>
                                <paragraph>
                                See Section 3.4.2.1 of the IPFIX protocol
document <xref target="RFC5101"/>
                                </paragraph>
                        </reference>
                        <units>flows</units>
                </field>
```

```
                <field name="samplingFlowInterval" dataType="unsigned64"
                        elementId="TBD7" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the number
of Flows that are
                                consecutively sampled.  A value of 100 means
that 100 consecutive
                                Flows are sampled. For example, this
Information Element may be
                                used to describe the configuration of a
systematic count-based
                                Sampling Selector.
                                </paragraph>
                        </description>
                        <units>flows</units>
                </field>

                <field name="samplingFlowSpacing" dataType="unsigned64"
                        elementId="TBD8" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the number
of Flows between two
                                "samplingFlowInterval"s.  A value of 100 means
that the next
                                interval starts 100 Flows (which are not
sampled) after the
                                current "samplingFlowInterval" is over. For
example, this
                                Information Element may be used to describe the
configuration of a
                                systematic count-based Sampling Selector.
                                </paragraph>
                        </description>
                        <units>flows</units>
                </field>

                <field name="flowSamplingTimeInterval" dataType="unsigned64"
                        elementId="TBD9" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the time
interval in
                                microseconds during which all arriving Flows
are sampled. For
                                example, this Information Element may be used
to describe the
```

```
                                configuration of a systematic time-based
Sampling Selector.
                                </paragraph>
                        </description>
                        <units>microseconds</units>
                </field>

                <field name="flowSamplingTimeSpacing" dataType="unsigned64"
                        elementId="TBD10" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the time
interval in
                                microseconds between two
"flowSamplingTimeInterval"s.  A value of
```

```
                                100 means that the next interval starts 100
microseconds (during
                                which no Flows are sampled) after the current
                                "flowsamplingTimeInterval" is over. For
example, this Information
                                Element may used to describe the configuration
of a systematic
                                time-based Sampling Selector.
                                </paragraph>
                        </description>
                        <units>microseconds</units>
                </field>

                <field name="hashFlowDomain" dataType="unsigned16"
                        dataTypeSemantics="identifier"
                        elementId="TBD11" status="proposed">
                        <description>
                                <paragraph>
                                This Information Element specifies the
Information Elements that
                                are used by the Hash-based Flow Selection
Selector as the Hash
                                Domain.
                                </paragraph>
                        </description>
                </field>
```


Authors' Addresses

Salvatore D'Antonio
University of Napoli "Parthenope"
Centro Direzionale di Napoli Is. C4
Naples 80143
Italy

Phone: +39 081 5476766
Email: salvatore.dantonio@uniparthenope.it

Tanja Zseby
CAIDA/FhG FOKUS
San Diego Supercomputer Center (SDSC)
University of California, San Diego (UCSD)
9500 Gilman Drive
La Jolla CA 92093-0505
USA

   Email: tanja@caida.org

Christian Henke
Tektronix Communication Berlin
Wohlrabedamm 32
Berlin 13629
Germany

Phone: +49 17 2323 8717
Email: christian.henke@tektronix.com

Lorenzo Peluso
University of Napoli
Via Claudio 21
Napoli 80125
Italy

Phone: +39 081 7683821
Email: lorenzo.peluso@unina.it