

Network Working Group
Request for Comments: 5476
Category: Standards Track

B. Claise, Ed.
A. Johnson
Cisco Systems, Inc.
J. Quittek
NEC Europe Ltd.
March 2009

Packet Sampling (PSAMP) Protocol Specifications

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Abstract

This document specifies the export of packet information from a Packet SAMPLing (PSAMP) Exporting Process to a PSAMP Collecting Process. For export of packet information, the IP Flow Information eXport (IPFIX) protocol is used, as both the IPFIX and PSAMP architecture match very well, and the means provided by the IPFIX protocol are sufficient. The document specifies in detail how the IPFIX protocol is used for PSAMP export of packet information.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. PSAMP Documents Overview	4
3. Terminology	4
3.1. IPFIX Terminology	4
3.2. PSAMP Terminology	5
3.2.1. Packet Streams and Packet Content	5
3.2.2. Selection Process	6
3.2.3. Reporting	7
3.2.4. Metering Process	8
3.2.5. Exporting Process	8
3.2.6. PSAMP Device	8
3.2.7. Collector	8
3.2.8. Selection Methods	9
3.3. IPFIX and PSAMP Terminology Comparison	11
3.3.1. IPFIX and PSAMP Processes	11
3.3.2. Packet Report, Packet Interpretation, and Data Record	12
4. Differences between PSAMP and IPFIX	12
4.1. Architecture Point of View	12
4.2. Protocol Point of View	14
4.3. Information Model Point of View	14
5. PSAMP Requirements versus the IPFIX Solution	14
5.1. High-Level View of the Integration	15
6. Using the IPFIX Protocol for PSAMP	16
6.1. Selector ID	17
6.2. The Selection Sequence ID	17
6.3. The Exporting Process	17
6.4. Packet Report	17
6.4.1. Basic Packet Report	17
6.4.2. Extended Packet Report	21
6.5. Report Interpretation	22
6.5.1. Selection Sequence Report Interpretation	23
6.5.2. Selector Report Interpretation	25
6.5.2.1. Systematic Count-Based Sampling	25
6.5.2.2. Systematic Time-Based Sampling	27

6.5.2.3. Random n-out-of-N Sampling	28
6.5.2.4. Uniform Probabilistic Sampling	29
6.5.2.5. Property Match Filtering	31
6.5.2.6. Hash-Based Filtering	33
6.5.2.7. Other Selection Methods	36
6.5.3. Selection Sequence Statistics Report Interpretation	37
6.5.4. Accuracy Report Interpretation	39
7. Security Considerations	43
8. IANA Considerations	43
8.1. IPFIX-Related Considerations	43
8.2. PSAMP-Related Considerations	43
9. References	44
9.1. Normative References	44
9.2. Informative References	44
10. Acknowledgments	45

1. Introduction

The name PSAMP is a contraction of the phrase "Packet Sampling". The word "Sampling" captures the idea that only a subset of all packets passing a network element will be selected for reporting. PSAMP selection operations include random selection, deterministic selection, and deterministic approximations to random selection (Hash-based Selection).

The IP Flow Information eXport (IPFIX) protocol specified in [RFC5101] exports IP traffic information [RFC5102] observed at network devices. This matches the general protocol requirements outlined in the PSAMP framework [RFC5474]. However, there are some architectural differences between IPFIX and PSAMP in the requirements for an export protocol. While the IPFIX architecture [RFC5470] is focused on gathering and exporting IP traffic flow information, the focus of the PSAMP framework [RFC5474] is on exporting information on individual packets. This basic difference and a set of derived differences in protocol requirements are outlined in Section 4. Despite these differences, the IPFIX protocol is well suited for the PSAMP protocol. Section 5 specifies how the IPFIX protocol is used for the export of packet samples. Required extensions of the IPFIX information model are specified in the PSAMP information model [RFC5477].

1.1. Conventions Used in This Document

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].

2. PSAMP Documents Overview

This document is one out of a series of documents from the PSAMP group.

[RFC5474]: "A Framework for Packet Selection and Reporting" describes the PSAMP framework for network elements to select subsets of packets by statistical and other methods, and to export a stream of reports on the selected packets to a Collector.

[RFC5475]: "Sampling and Filtering Techniques for IP Packet Selection" describes the set of packet selection techniques supported by PSAMP.

RFC 5476 (this document): "Packet Sampling (PSAMP) Protocol Specifications" specifies the export of packet information from a PSAMP Exporting Process to a PSAMP Collecting Process.

[RFC5477]: "Information Model for Packet Sampling Exports" defines an information and data model for PSAMP.

3. Terminology

As the IPFIX export protocol is used to export the PSAMP information, the relevant IPFIX terminology from [RFC5101] is copied over in this document. All terms defined in this section have their first letter capitalized when used in this document. The terminology summary table in Section 3.1 gives a quick overview of the relationships between the different IPFIX terms. The PSAMP terminology defined here is fully consistent with all terms listed in [RFC5475] and [RFC5474], but only definitions that are relevant to the PSAMP protocol appear here. Section 3.3 applies the PSAMP terminology to the IPFIX protocol terminology.

3.1. IPFIX Terminology

IPFIX-specific terminology used in this document is defined in Section 2 of [RFC5101]. The only exceptions are the Metering Process, Exporting Process, and the Collector terms, which are defined more precisely in the PSAMP terminology section. In this document, as in [RFC5101], the first letter of each IPFIX-specific term is capitalized.

Set	contents	
	Template	record
Data Set	/	Data Record(s)
Template Set	Template Record(s)	/
Options Template Set	Options Template Record(s)	/

Figure A: Terminology Summary Table

3.2. PSAMP Terminology

The PSAMP terminology section has been copied from [RFC5475].

3.2.1. Packet Streams and Packet Content

* Observed Packet Stream

The Observed Packet Stream is the set of all packets observed at the Observation Point.

* Packet Stream

A Packet Stream denotes a set of packets from the Observed Packet Stream that flows past some specified point within the Metering Process. An example of a Packet Stream is the output of the Selection Process. Note that packets selected from a stream, e.g., by Sampling, do not necessarily possess a property by which they can be distinguished from packets that have not been selected. For this reason, the term "stream" is favored over "flow", which is defined as a set of packets with common properties [RFC3917].

* Packet Content

The Packet Content denotes the union of the packet header (which includes link layer, network layer, and other encapsulation headers) and the packet payload. Note that, depending on the Observation Point, the link layer information might not be available.

3.2.2. Selection Process

* Selection Process

A Selection Process takes the Observed Packet Stream as its input and selects a subset of that stream as its output.

* Selection State

A Selection Process may maintain state information for use by the Selection Process. At a given time, the Selection State may depend on packets observed at and before that time, and other variables. Examples include:

- (i) sequence numbers of packets at the input of Selectors;
- (ii) a timestamp of observation of the packet at the Observation Point;
- (iii) iterators for pseudorandom number generators;
- (iv) hash values calculated during selection;
- (v) indicators of whether the packet was selected by a given Selector.

Selection Processes may change portions of the Selection State as a result of processing a packet. Selection state for a packet is to reflect the state after processing the packet.

* Selector

A Selector defines the action of a Selection Process on a single packet of its input. If selected, the packet becomes an element of the output Packet Stream.

The Selector can make use of the following information in determining whether a packet is selected:

- (i) the Packet Content;
- (ii) information derived from the packet's treatment at the Observation Point;
- (iii) any selection state that may be maintained by the Selection Process.

- * Composite Selector

A Composite Selector is an ordered composition of Selectors, in which the output Packet Stream issuing from one Selector forms the input Packet Stream to the succeeding Selector.

- * Primitive Selector

A Selector is primitive if it is not a Composite Selector.

- * Selector ID

The Selector ID is the unique ID identifying a Primitive Selector. The ID is unique within the Observation Domain.

- * Selection Sequence

From all the packets observed at an Observation Point, only a few packets are selected by one or more Selectors. The Selection Sequence is a unique value per Observation Domain describing the Observation Point and the Selector IDs through which the packets are selected.

3.2.3. Reporting

- * Packet Reports

Packet Reports comprise a configurable subset of a packet's input to the Selection Process, including the Packet Content, information relating to its treatment (for example, the output interface), and its associated selection state (for example, a hash of the Packet Content).

- * Report Interpretation

Report Interpretation comprises subsidiary information, relating to one or more packets, that is used for interpretation of their Packet Reports. Examples include configuration parameters of the Selection Process.

- * Report Stream

The Report Stream is the output of a Metering Process, comprising two distinguished types of information: Packet Reports and Report Interpretation.

3.2.4. Metering Process

* Metering Process

A Metering Process selects packets from the Observed Packet Stream using a Selection Process, and produces as output a Report Stream concerning the selected packets.

The PSAMP Metering Process can be viewed as analogous to the IPFIX Metering Process [RFC5101], which produces Flow Records as its output, with the difference that the PSAMP Metering Process always contains a Selection Process. The relationship between PSAMP and IPFIX is further described in [RFC5477] and [RFC5474].

3.2.5. Exporting Process

* Exporting Process

An Exporting Process sends, in the form of Export Packets, the output of one or more Metering Processes to one or more Collectors.

* Export Packet

An Export Packet is a combination of Report Interpretation(s) and/or one or more Packet Reports that are bundled by the Exporting Process into an Export Packet for exporting to a Collector.

3.2.6. PSAMP Device

* PSAMP Device

A PSAMP Device is a device hosting at least an Observation Point, a Selection Process, and an Exporting Process. Typically, corresponding Observation Point(s), Selection Process(es), and Exporting Process(es) are co-located at this device, for example, at a router.

3.2.7. Collector

* Collector

A Collector receives a Report Stream exported by one or more Exporting Processes. In some cases, the host of the Metering and/or Exporting Processes may also serve as the Collector.

3.2.8. Selection Methods

* Filtering

A filter is a Selector that selects a packet deterministically based on the Packet Content, or its treatment, or functions of these occurring in the Selection State. Two examples are:

- (i) Property Match Filtering: A packet is selected if a specific field in the packet equals a predefined value.
- (ii) Hash-based Selection: A Hash Function is applied to the Packet Content, and the packet is selected if the result falls in a specified range.

* Sampling

A Selector that is not a filter is called a Sampling operation. This reflects the intuitive notion that if the selection of a packet cannot be determined from its content alone, there must be some type of Sampling taking place.

* Content-Independent Sampling

A Sampling operation that does not use Packet Content (or quantities derived from it) as the basis for selection is called a Content-independent Sampling operation. Examples include systematic Sampling, and uniform pseudorandom Sampling driven by a pseudorandom number whose generation is independent of Packet Content. Note that in Content-independent Sampling, it is not necessary to access the Packet Content in order to make the selection decision.

* Content-Dependent Sampling

A Sampling operation where selection is dependent on Packet Content is called a Content-dependent Sampling operation. An example is pseudorandom selection according to a probability that depends on the contents of a packet field. Note that this is not a filter, because the selection is not deterministic.

* Hash Domain

A Hash Domain is a subset of the Packet Content and the packet treatment, viewed as an N-bit string for some positive integer N.

*** Hash Range**

A Hash Range is a set of M-bit strings for some positive integer M that define the range of values the result of the hash operation can take.

*** Hash Function**

A Hash Function defines a deterministic map from the Hash Domain into the Hash Range.

*** Hash Selection Range**

A Hash Selection Range is a subset of the Hash Range. The packet is selected if the action of the Hash Function on the Hash Domain for the packet yields a result in the Hash Selection Range.

*** Hash-based Selection**

A Hash-based Selection is Filtering specified by a Hash Domain, a Hash Function, a Hash Range, and a Hash Selection Range.

*** Approximative Selection**

Selectors in any of the above categories may be approximated by operations in the same or another category for the purposes of implementation. For example, uniform pseudorandom Sampling may be approximated by Hash-based Selection, using a suitable Hash Function and Hash Domain. In this case, the closeness of the approximation depends on the choice of Hash Function and Hash Domain.

*** Population**

A Population is a Packet Stream, or a subset of a Packet Stream. A Population can be considered as a base set from which packets are selected. An example is all packets in the Observed Packet Stream that are observed within some specified time interval.

*** Population Size**

The Population Size is the number of all packets in the Population.

* Sample Size

The Sample Size is the number of packets selected from the Population by a Selector.

* Configured Selection Fraction

The Configured Selection Fraction is the expected ratio of the Sample Size to the Population Size, as based on the configured selection parameters.

* Attained Selection Fraction

The Attained Selection Fraction is the ratio of the actual Sample Size to the Population Size. For some Sampling methods, the Attained Selection Fraction can differ from the Configured Selection Fraction due to, for example, the inherent statistical variability in Sampling decisions of probabilistic Sampling and Hash-based Selection. Nevertheless, for large Population Sizes and properly configured Selectors, the Attained Selection Fraction usually approaches the Configured Selection Fraction.

3.3. IPFIX and PSAMP Terminology Comparison

The PSAMP terminology has been specified with an IPFIX background, as PSAMP and IPFIX have similar terms. However, this section clarifies the terms between the IPFIX and PSAMP terminology.

3.3.1. IPFIX and PSAMP Processes

Figure B indicates the sequence of the IPFIX processes (Metering and Exporting) within the PSAMP Device.

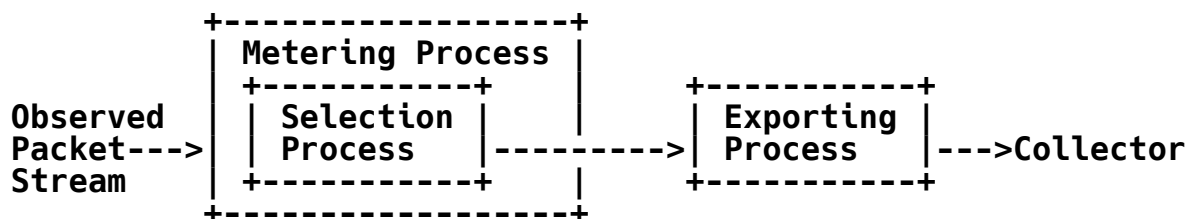


Figure B: PSAMP Processes

The Selection Process, which takes an Observed Packet Stream as its input, is an integral part of the Metering Process. The Selection Process chooses which packets from its input Packet Stream will be

reported on by the rest of the Metering Process. Note that a "Process" is not necessarily implemented as a separate CPU thread.

3.3.2. Packet Report, Packet Interpretation, and Data Record

The PSAMP terminology speaks of Packet Report and Packet Interpretation, while the IPFIX terminology speaks of Data Record and (Options) Template Record. The PSAMP Packet Report, which comprises information about the observed packet, can be viewed as analogous to the IPFIX Data Record defined by a Template Record. The PSAMP Report Interpretation, which comprises subsidiary information used for the interpretation of the Packet Reports, can be viewed as analogous to the IPFIX Data Record defined by an Options Template Record. This Options Template Record contains subsidiary information, applicable to the observed packet sent into the PSAMP Packet Report.

4. Differences between PSAMP and IPFIX

The output of the IPFIX working group relevant for this document is structured into three documents:

- IP Flow information architecture [RFC5470]
- IPFIX protocol specifications [RFC5101]
- IP Flow information export information model [RFC5102]

In the following sections, we investigate the differences between IPFIX and PSAMP for each of those aspects.

4.1. Architecture Point of View

Traffic Flow measurement as described in the IPFIX requirements [RFC3917] and the IPFIX architecture [RFC5470] can be separated into two stages: packet processing and Flow processing. Figure C illustrates these stages.

In stage 1, all processing steps act on packets. Packets are captured, timestamped, selected by one or more selection steps, and finally forwarded to packet classification that maps packets to Flows. The packets' selection steps may include Filtering and Sampling functions.

In stage 2, all processing steps act on Flows. After packets are classified (mapped to Flows), Flows are generated (or updated if they exist already). Flow generation and update steps may be performed repeatedly for aggregating Flows. Finally, Flows are exported.

Packet Sampling as described in the PSAMP framework [RFC5474] covers only stage 1 of the IPFIX architecture with the packet classification replaced by Packet Report export, while IPFIX covers stage 2 also, as it generates Flow Records out of the selected packets.

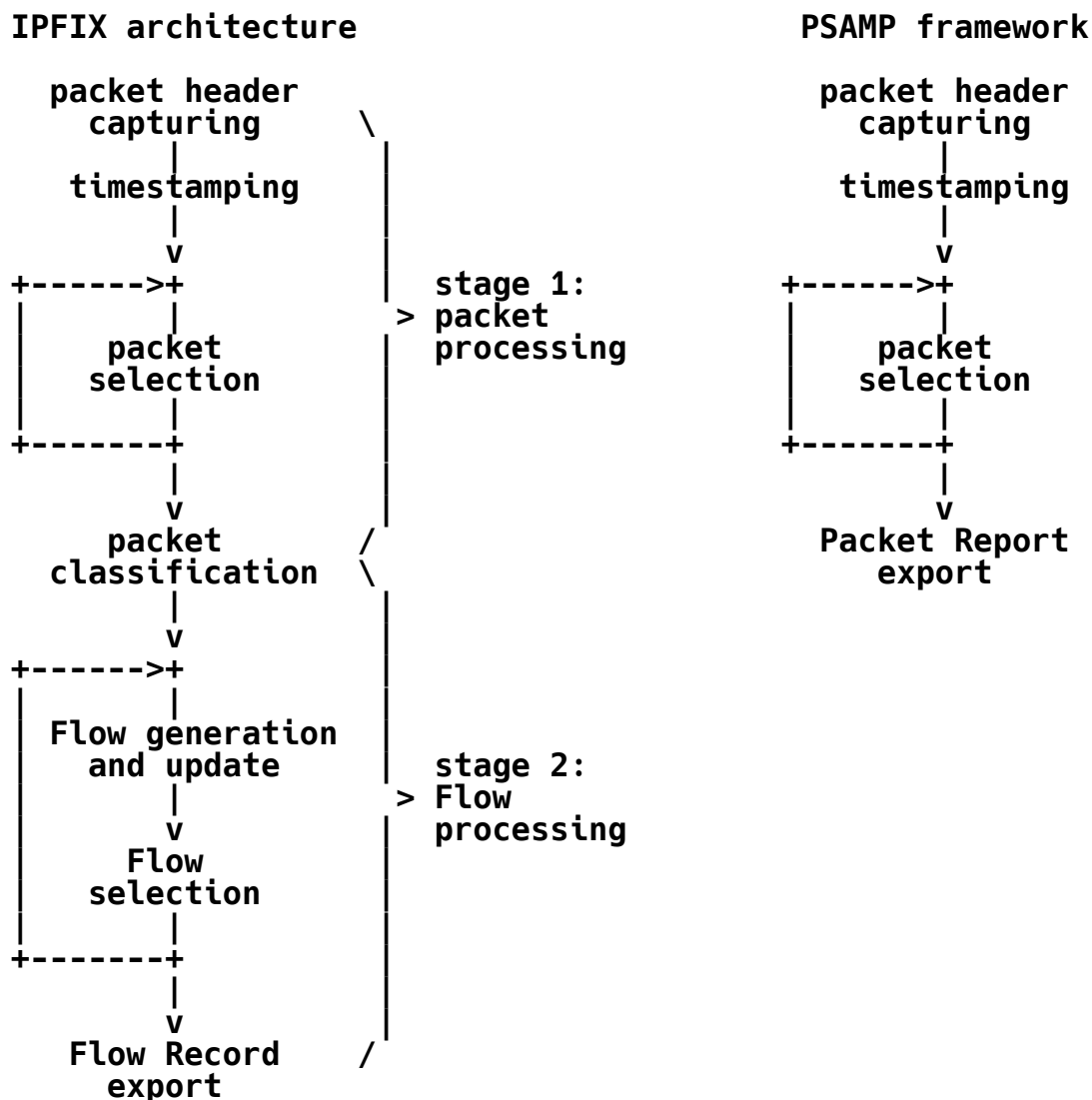


Figure C: Comparison of IPFIX Architecture and PSAMP Framework

4.2. Protocol Point of View

Concerning the protocol, the major difference between IPFIX and PSAMP is that the IPFIX protocol exports Flow Records while the PSAMP protocol exports Packet Reports. From a pure export point of view, IPFIX will not distinguish a Flow Record composed of several packets aggregated together from a Flow Record composed of a single packet.

So the PSAMP export can be seen as a special IPFIX Flow Record containing information about a single packet.

All extensions of the IPFIX protocol that are required to satisfy the PSAMP requirements have already been incorporated in the IPFIX protocol [RFC5101], which was developed in parallel with the PSAMP protocol. An example is the need for a data type for protocol fields that have flexible length, such as an octet array. This was added to the IPFIX protocol specification in order to meet the requirement of the PSAMP protocol to report content of captured packets, for example, the first octets of a packet.

4.3. Information Model Point of View

From the information model point of view, the overlap between both the IPFIX and PSAMP protocols is quite large. Most of the Information Elements in the IPFIX protocol are also relevant for exporting packet information, for example, all fields reporting packet header properties. Only a few Information Elements, such as `observedFlowTotalCount` (whose value will always be 1 for PSAMP), etc., cannot be used in a meaningful way by the PSAMP protocol. Also, IPFIX protocol requirements concerning stage 2 of Figure C do not apply to the PSAMP Metering Process.

Further required extensions apply to the information model. Even if the IPFIX charter speaks of Sampling, no Sampling-related Information Elements are specified in [RFC5102]. The task of specifying them was intentionally left for the PSAMP information model [RFC5477]. A set of several additional fields is required for satisfying the requirements for the PSAMP information model [RFC5475].

Exploiting the extensibility of the IPFIX information model, the required extension is covered by the PSAMP information model specified in [RFC5477].

5. PSAMP Requirements versus the IPFIX Solution

The [RFC5474] contains PSAMP protocol requirements throughout the document, with a special focus in Section 4, "Generic Requirements for PSAMP", and its subsections.

Section 4 of [RFC5474] describes one requirement that, if not directly related to the export protocol, will put some constraints on it. Parallel Measurements: multiple independent Selection Processes at the same entity.

[RFC5474] also describes a series of requirements specifying the different Information Elements that MUST and SHOULD be reported to the Collector. Nevertheless, IPFIX, being a generic export protocol, can export any Information Elements as long as they are described in the information model. So these requirements are mainly targeted for [RFC5477].

The PSAMP protocol specification meets almost all the protocol requirements stated in the PSAMP framework document [RFC5474]:

- * Extensibility
- * Parallel selection processes
- * Encrypted packets
- * Indication of information loss
- * Accuracy
- * Privacy
- * Timeliness
- * Congestion avoidance
- * Secure export
- * Export rate limit
- * Microsecond timestamp resolution

The only requirement that is not met is Export Packet compression. With the choice of IPFIX as the PSAMP export protocol, the Export Packet compression option mentioned in the Section 8.5 of the framework document [RFC5474] is not addressed.

5.1. High-Level View of the Integration

The Template Record in the Template Set is used to describe the different PSAMP Information Elements that will be exported to the Collector. The Collector decodes the Template Record in the Template Set and knows which Information Elements to expect when it receives

the Data Records in the PSAMP Packet Report Data Set. Typically, in the base level of the PSAMP functionality, the Template Set will contain the input sequence number, the packet fragment (some number of contiguous bytes from the start of the packet or from the start of the payload), and the Selection Sequence.

The Options Template Record in the Options Template Set is used to describe the different PSAMP Information Elements that concern the Metering Process itself: Sampling and/or Filtering functions, and the associated parameters. The Collector decodes the Options Template Records in the Options Template Set and knows which Information Elements to expect when it receives the Data Records in the PSAMP Report Interpretation Data Set. Typically, the Options Template would contain the Selection Sequence, the Sampling or Filtering functions, and the Sampling or Filtering associated parameters.

PSAMP requires all the different possibilities of the IPFIX protocol specifications [RFC5101], that is, the three types of Sets (Data Set, Template Set, and Options Templates Set) with the two types of Template Records (Template Record and Options Template Record), as described in Figure A. As a consequence, PSAMP can't rely on a subset of the IPFIX protocol specifications described in [RFC5101]. The entire IPFIX protocol specifications [RFC5101] MUST be implemented for the PSAMP protocol.

6. Using the IPFIX Protocol for PSAMP

In this section, we describe the usage of the IPFIX protocol for PSAMP. We describe the record formats and the additional requirements that must be met. PSAMP uses two different types of messages:

- Packet Reports
- Report Interpretation

The format of Packet Reports is defined in IPFIX Template Records. The PSAMP data is transferred as Information Elements in IPFIX Data Records as described by the Template Record. There are two different types of Packet Reports. Basic Packet Reports contain only the basic Information Elements required for PSAMP reporting. Extended Packet Reports MAY contain other Information Elements, and do not necessarily include Packet Content (See section 6.4.2).

The format of Report Interpretations is defined in the IPFIX Options Template Record. The Information Elements are transferred in IPFIX Data Records as described by the Options Template Record. There are four different types of Report Interpretation messages:

- Selection Sequence Report Interpretation
- Selector Report Interpretation
- Selection Sequence Statistics Report Interpretation
- Accuracy Report Interpretation

A description and examples about the usage of those reports are given below.

6.1. Selector ID

The Selector ID is the unique ID identifying a Primitive Selector. Each Primitive Selector **MUST** have a unique ID within the Observation Domain. The Selector ID is represented by the selectorId Information Element [RFC5477].

6.2. The Selection Sequence ID

From all the packets observed at an Observation Point, a subset of packets is selected by one or more Selectors. The Selection Sequence is the combination of an Observation Point and one or more Selector(s) through which the packets are selected. The Selection Sequence ID is a unique value representing that combination. The Selection Sequence ID is represented by the selectionSequenceId Information Element [RFC5477].

6.3. The Exporting Process

An Exporting Process **MUST** be able to limit the export rate according to a configurable value. The Exporting Process **MAY** limit the export rate on a per Collecting Process basis.

6.4. Packet Report

For each Selection Sequence, for each selected packet, a Packet Report **MUST** be created. The format of the Packet Report is specified in a Template Record contained in a Template Set.

There are two types of Packet Report, as described in [RFC5474]: the basic Packet Report and the extended Packet Report.

6.4.1. Basic Packet Report

For each selected packet, the Packet Report **MUST** contain the following information:

- The selectionSequenceId Information Element
If there is a digest function in the Selection Sequence, the Packet Report MUST contain the hash value (digestHashValue Information Element) generated by the digest Hash Function for each selected packet. If there is more than one digest function, then each hash value MUST be included in the same order as they appear in the Selection Sequence. If there are no digest functions in the Selection Sequence, no element for the digest needs to be sent.
- Some number of contiguous bytes from the start of the packet, including the packet header (which includes link layer, network layer, and other encapsulation headers) and some subsequent bytes of the packet payload. Alternatively, the number of contiguous bytes may start at the beginning of the payload. The dataLinkFrameSection, mplsLabelStackSection, mplsPayloadPacketSection, ipPacketSection, and ipPayloadPacketSection PSAMP Information Elements are available for this use.

For each selected packet, the Packet Report SHOULD contain a time-related Information Element that matches the Metering Process time accuracy. Typically, the observationTimeMicroseconds Information Element. Other possible Information Elements are the observationTimeSeconds, the observationTimeMilliseconds, or the observationTimeNanoseconds.

In the Packet Report, the PSAMP Device MUST be capable of exporting the number of observed packets and the number of packets selected by each instance of its Primitive Selectors (as described by the non-scope Information Elements of the Selection Sequence Statistics Report Interpretation), although it MAY be a configurable option not to include them. If exported, the Attained Selection Fraction may be calculated precisely for the Observed Packet Stream. The Packet Report MAY include only the final selector packetSelected, to act as an index for that Selection Sequence in the Selection Sequence Statistics Report Interpretation, which also allows the calculation of the Attained Selection Fraction.

The contiguous Information Elements (dataLinkFrameSection, mplsLabelStackSection, mplsPayloadPacketSection, ipPacketSection, and ipPayloadPacketSection) MAY be encoded with a fixed-length field or with a variable-sized field. If one of these Information Elements is encoded with a fixed-length field whose length is too long for the number of contiguous bytes in the selected packet, padding MUST NOT be used. In this case, the Exporting Process MUST export the information either in a new Template Record with the correct fixed-length field or in a new Template Record with a variable-length field.

Here is an example of a basic Packet Report, with a SelectionSequenceId value of 9 and dataLinkFrameSection Information Element of 12 bytes, 0x4500 005B A174 0000 FF11 832E, encoded with a fixed-length field.

IPFIX Template Record:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Set ID = 2		Length = 24	
Template ID = 260		Field Count = 4	
selectionSequenceId = 301		Field Length = 4	
digestHashValue = 326		Field Length = 4	
dataLinkFrameSection = 315		Field Length = 12	
observationTimeMicroseconds=324		Field Length = 4	

The associated IPFIX Data Record:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Set ID = 260		Length = 32	
	9		
	0x9123 0613		
	0x4500 005B		
	0xA174 0000		
	0xFF11 832E		
	observation time ...		
	... encoded as dateTimeMicroSeconds [RFC5101]		

Figure D: Example of a Basic Packet Report

Here is an example of a basic Packet Report, with a SelectionSequenceId value of 9 and ipHeaderPacketSection Information Element of 12 bytes, 0x4500 005B A174 0000 FF11 832E, encoded with a variable-sized field.

IPFIX Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 2										Length = 16																													
Template ID = 261										Field Count = 2																													
selectionSequenceId = 301										Field Length = 4																													
ipHeaderPacketSection = 313										Field Length = 65535																													

The associated IPFIX Data Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+-----+																																							

Figure E: Example of a Basic Packet Report
with a Variable-Sized Field

6.4.2. Extended Packet Report

Alternatively to the basic Packet Report, the extended Packet Report MAY contain other Information Elements related to the protocols used in the packet (such as source and destination IP addresses), related to the packet treatment (such as output interface, destination BGP autonomous system [RFC4271]), or related to the Selection State associated with the packet (such as timestamp, hash value).

It is envisaged that selection of fields for extended Packet Reports may be used to reduce reporting bandwidth, in which case the option to report some number of contiguous bytes from the start of the packet, mandatory in the basic Packet Report, may not be exercised. In this case, the Packet Content MAY be omitted. Note this configuration is quite similar to an IPFIX Device for which a Template Record containing information about a single packet is reported.

Example of a detailed Extended Packet Report:

IPFIX Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 2										Length = 32																													
Template ID = 261										Field Count = 6																													
0 selectionSequenceId = 301										Field Length = 4																													
0 sourceIPv4Address = 8										Field Length = 4																													
0 destinationIPv4Address = 12										Field Length = 4																													
0 totalLengthIPv4 = 190										Field Length = 2																													
0 tcpSourcePort = 182										Field Length = 2																													
0 tcpDestinationPort = 183										Field Length = 2																													

The associated IPFIX Data Record:

0										1										2										3												
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Set ID = 261																				Length = 20																						
										9																																
										192.0.2.1																																
										192.0.2.106																																
										72										1372																						
										80																																

Figure F: Example of an Extended Packet Report

6.5. Report Interpretation

To make full sense of the Packet Reports, there are a number of additional pieces of information that must be communicated to the Collector:

- The details about which Selectors and Observation Points are being used within a Selection Sequence **MUST** be provided using the Selection Sequence Report Interpretation.
- The configuration details of each Selector **MUST** be provided using the Selector Report Interpretation.
- The Selector ID statistics **MUST** be provided using the Selection Sequence Statistics Report Interpretation.
- The accuracies of the reported fields **MUST** be provided using the Accuracy Report Interpretation.

6.5.1. Selection Sequence Report Interpretation

Each Packet Report contains a selectionSequenceId Information Element that identifies the particular combination of Observation Point and Selector(s) used for its selection. For every selectionSequenceId Information Element in use, the PSAMP Device MUST export a Selection Sequence Report Interpretation using an Options Template with the following Information Elements:

Scope: selectionSequenceId
Non-Scope: one Information Element mapping the Observation Point selectorId (one or more)

An Information Element representing the Observation Point would typically be taken from the ingressInterface, egressInterface, lineCardId, exporterIPv4Address, or exporterIPv6Address Information Elements (specified in [RFC5102]), but is not limited to those: any Information Element specified in [RFC5102] or [RFC5477] can potentially be used. In case of more complex Observation Points (such as a list of interfaces, a bus, etc.), a new Information Element describing the new type of Observation Point must be specified, along with an Options Template Record describing it in more detail (if necessary).

If the packets are selected by a Composite Selector, the Selection Sequence is composed of several Primitive Selectors. In such a case, the Selection Sequence Report Interpretation MUST contain the list of all the Primitive Selector IDs in the Selection Sequence. If multiple Selectors are contained in the Selection Sequence Report Interpretation, the selectorId's MUST be identified in the order they are used.

Example of two Selection Sequences:

Selection Sequence 7 (Filter->Sampling):

ingressInterface	5
selectorId	5 (Filter, match IPV4SourceAddress 192.0.2.1)
selectorId	10 (Sampler, Random 1 out-of ten)

Selection Sequence 9 (Sampling->Filtering):

ingressInterface	5
selectorId	10 (Sampler, Random 1 out-of ten)
selectorId	5 (Filter, match IPV4SourceAddress 192.0.2.1)

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Set ID = 3										Length = 26																													
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Template ID = 262										Field Count = 4																													
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Scope Field Count = 1										0	selectionSequenceId = 301																												
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Scope 1 Length = 4										0	ingressInterface = 10																												
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Field Length = 4										0	selectorId = 302																												
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Field Length = 4										0	selectorId = 302																												
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								
Field Length = 4																																							
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-								

The associated IPFIX Data Record:

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Set ID = 262										Length = 36																					
										7																					
										5																					
										5																					
										10																					
										9																					
										5																					
										10																					
										5																					

Figure G: Example of a Selection Sequence Report Interpretation

Notes:

- * There are two Records here in the same Data Set. Each record defines a different Selection Sequence.
- * If, for example, a different Selection Sequence is composed of three Selectors, then a different Options Template with three selectorId Information Elements (instead of two) must be used.

6.5.2. Selector Report Interpretation

An IPFIX Data Record, defined by an Options Template Record, **MUST** be used to send the configuration details of every Selector in use. The Options Template Record **MUST** contain the selectorId Information Element as the Scope field and the SelectorAlgorithm Information Element followed by some specific configuration parameters:

Scope: selectorId
Non-scope: selectorAlgorithm
 algorithm-specific Information Elements

The algorithm-specific Information Elements are specified in the following subsections, depending on the selection method represented by the value of the selectorAlgorithm [RFC5477].

6.5.2.1. Systematic Count-Based Sampling

In systematic count-based Sampling, the start and stop triggers for the Sampling interval are defined in accordance with the spatial packet position (packet count) [RFC5475].

The **REQUIRED** algorithm-specific Information Elements in the case of systematic count-based Sampling are:

samplingPacketInterval: number of packets selected in a row
samplingPacketSpace: number of packets between selections

Example of a simple 1 out-of 10 systematic count-based Selector definition, where the samplingPacketInterval is 1 and the samplingPacketSpace is 9.

IPFIX Options Template Record:

1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Set ID = 3										Length = 26																					
Template ID = 263										Field Count = 4																					
Scope Field Count = 1										0	selectorId = 302																				
Scope 1 Length = 4										0	selectorAlgorithm = 304																				
Field Length = 1										0	samplingPacketInterval = 305																				
Field Length = 1										0	samplingPacketSpace = 306																				
Field Length = 1																															

Associated IPFIX Data Record:

Diagram illustrating the structure of the Set ID field (32 bits):

- Segment 0 (bits 0-7): Set ID = 263
- Segment 1 (bits 8-15): 15
- Segment 2 (bits 16-23): 9
- Segment 3 (bits 24-31): 1

Figure H: Example of the Selector Report Interpretation for Systematic Count-Based Sampling

Notes:

- * A selectorAlgorithm value of 1 represents systematic count-based Sampling.
- * samplingPacketInterval and samplingPacketSpace are of type unsigned32 but are compressed down to one octet here, as allowed by the IPFIX protocol specifications [RFC5101].

6.5.2.2. Systematic Time-Based Sampling

In systematic time-based Sampling, the start and stop triggers are used to define the Sampling intervals [RFC5475]. The REQUIRED algorithm-specific Information Elements in the case of systematic time-based Sampling are:

samplingTimeInterval: time (in microseconds) when packets are
selected
samplingTimeSpace: time (in microseconds) between selections

Example of a 100 microsecond out-of 1000 microsecond systematic time-based Selector definition, where the samplingTimeInterval is 100 and the samplingTimeSpace is 900.

IPFIX Options Template Record:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Set ID = 3		Length = 26	
Template ID = 264		Field Count = 4	
Scope Field Count = 1	0	selectorId = 302	
Scope 1 Length = 4	0	selectorAlgorithm = 304	
Field Length = 1	0	samplingTimeInterval = 307	
Field Length = 1	0	samplingTimeSpace = 308	
Field Length = 2			

Associated IPFIX Data Record:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
Set ID = 264		Length = 12	
	16		
2	100	900	

Figure I: Example of the Selector Report Interpretation

for Systematic Time-Based Sampling

Notes:

- * A selectorAlgorithm value of 2 represents systematic time-based Sampling.
- * samplingTimeInterval and samplingTimeSpace are of type unsigned32 but are compressed down here.

6.5.2.3. Random n-out-of-N Sampling

In random n-out-of-N Sampling, n elements are selected out of the parent Population that consists of N elements [RFC5475]. The REQUIRED algorithm-specific Information Elements in case of random n-out-of-N Sampling are:

samplingSize: number of packets selected
 samplingPopulation: number of packets in selection Population

Example of a 1 out-of-10 random n-out-of-N Sampling Selector:

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 3										Length = 26																													
Template ID = 265										Field Count = 4																													
Scope Field Count = 1										0	selectorId = 302																												
Scope 1 Length = 4										0	selectorAlgorithm = 304																												
Field Length = 1										0	samplingSize = 309																												
Field Length = 1										0	samplingPopulation = 310																												
Field Length = 1																																							

Associated IPFIX Data Record:

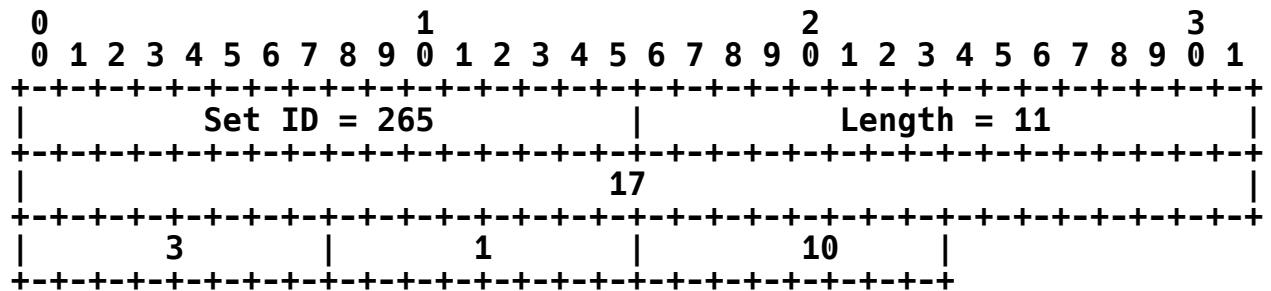


Figure J: Example of the Selector Report Interpretation
for Random n-out-of-N Sampling

Notes:

- * A selectorAlgorithm value of 3 represents Random n-out-of-N Sampling.
- * samplingSize and samplingPopulation are of type unsigned32 but are compressed down to one octet here.

6.5.2.4. Uniform Probabilistic Sampling

In uniform probabilistic Sampling, each element has the same probability p of being selected from the parent Population [RFC5475]. The algorithm-specific Information Element in case of uniform probabilistic Sampling is:

samplingProbability: a floating point number for the Sampling probability.

Example of a 15% uniform probability Sampling Selector:

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+-----+																																							

Associated IPFIX Data Record:

0										1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+																																

Figure K: Example of the Selector Report Interpretation for Uniform Probabilistic Sampling

Notes:

- * A selectorAlgorithm value of 4 represents Uniform Probabilistic Sampling.
- * samplingProbability is of type float64 but is compressed down to a float32 here.

6.5.2.5. Property Match Filtering

This classification includes match(es) on field(s) within a packet and/or on properties of the router state. With this method, a packet is selected if a specific field in the packet equals a predefined value.

The algorithm-specific Information Elements defining configuration parameters for Property Match Filtering are taken from the full range of available Information Elements.

When multiple different Information Elements are defined, the filter acts as a logical AND. Note that the logical OR is not covered by these PSAMP specifications. The Property Match Filtering Options Template Record MUST NOT have multiple identical Information Elements. The result of the filter is independent from the order of the Information Elements in the Options Template Record, but the order may be important for implementation purposes, as the first filter will have to work at a higher rate. In any case, an implementation is not constrained to respect the filter ordering as long as the result is the same, and it may even implement the composite Filtering in one single step.

Since encryption alters the meaning of encrypted fields, when the Property Match Filtering classification is based on the encrypted field(s) in the packet, it MUST be able to recognize that the field(s) are not available and MUST NOT select those packets unless specifically directed by the Information Element description. Even if they are ignored, the encrypted packets MUST be accounted for in the Selector packetsObserved Information Element [RFC5477], part of the Selection Sequence Statistics Report Interpretation.

Example of a match-based filter Selector, whose rules are:

IPv4 Source Address = 192.0.2.1
IPv4 Next-Hop Address = 192.0.2.129

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 3										Length = 26																													
Template ID = 266										Field Count = 4																													
Scope Field Count = 1										0	selectorId = 302																												
Scope 1 Length = 4										0	selectorAlgorithm = 304																												
Field Length = 1										0	sourceIPv4Address = 8																												
Field Length = 4										0	ipNextHopIPv4Address = 15																												
Field Length = 4																																							

Associated IPFIX Data Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 266										Length = 11																													
										21																													
5										192.0.2 ...																													
... .1										192.0.2 ...																													
... .129																																							

Figure L: Example of the Selector Report Interpretation for Match-Based and Router State Filtering

Notes:

- * A selectorAlgorithm value of 5 represents Property Match Filtering.
- * In this filter, there is a mix of information from the packet and information from the router.

6.5.2.6. Hash-Based Filtering

In Hash-based Selection, a Hash Function is run on IPv4 traffic. The following fields **MUST** be used as input to that Hash Function:

- IP identification field
- Flags field
- Fragment offset
- Source IP address
- Destination IP address
- A number of bytes from the IP payload. The number of bytes and starting offset **MUST** be configurable if the Hash Function supports it.

For the bytes taken from the IP payload, IPSX has a fixed offset of 0 bytes and a fixed size of 8 bytes. The number and offset of payload bytes in the BOB function **MUST** be configurable.

The minimum configuration ranges **MUST** be as follows:

Number of bytes:	from 8 to 32
Offset:	from 0 to 64

If the selected payload bytes are not available and the Hash Function can take a variable-sized input, then the Hash Function **MUST** be run with the information that is available and a shorter size. Passing 0 as a substitute for missing payload bytes is only acceptable if the Hash Function takes a fixed size as is the case with IPSX.

If the Hash Function can take an initialization value, then this value **MUST** be configurable.

A Hash-based Selection function **MAY** be configurable as a digest function. Any Selection Process that is configured as a digest function **MUST** have the output value included in the basic Packet Report for any selected packet.

Each Hash Function used as a Hash-based Selection Selector requires its own value for the selectorAlgorithm. Currently, we have BOB (6), IPSX (7), and CRC (8) defined and any **MAY** be used for either Filtering or creating a Packet Digest. Only BOB is recommended though and **SHOULD** be used.

The REQUIRED algorithm-specific Information Elements in case of Hash-based Selection are:

- hashIPPayloadOffset - The payload offset used by a Hash-based Selection Selector
- hashIPPayloadSize - The payload size used by a Hash-based Selection Selector
- hashOutputRangeMin - One or more values for the beginning of each potential output range.
- hashOutputRangeMax - One or more values for the end of each potential output range.
- hashSelectedRangeMin - One or more values for the beginning of each selected range.
- hashSelectedRangeMax - One or more values for the end of each selected range.
- hashDigestOutput - A boolean value, TRUE if the output from this Selector has been configured to be included in the Packet Report as a packet digest.

Note: If more than one selection or output range needs to be sent, then the minimum and maximum elements may be repeated as needed. These MUST make one or more non-overlapping ranges. The elements SHOULD be sent as pairs of minimum and maximum in ascending order; however, if they are sent out of order, then there will only be one way to interpret the ranges to produce a non-overlapping range and the Collecting Process MUST be prepared to accept and decode this.

The following algorithm-specific Information Element MAY be sent, but is optional for security considerations:

- hashInitialiserValue - The initialiser value to the Hash Function.

Since encryption alters the meaning of encrypted fields, when the Hash-based Filtering classification is based on the encrypted field(s) in the packet, it MUST be able to recognize that the field(s) are not available and MUST NOT select those packets. Even if they are ignored, the encrypted packets MUST be accounted for in the Selector packetsObserved Information Element [RFC5477], which is part of the Selection Sequence Statistics Report Interpretation.

Example of a Hash-based Filter Selector, whose configuration is:

```

Hash Function      = B0B
Hash IP Payload Offset = 0
Hash IP Payload Size  = 16
Hash Initialiser Value = 0x9A3F9A3F
Hash Output Range    = 0 to 0xFFFFFFFF
Hash Selected Range   = 100 to 200 and 400 to 500

```

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 3										Length = 50																													
Template ID = 269										Field Count = 8																													
Scope Field Count = 1										0	selectorId = 302																												
Scope 1 Length = 4										0	selectorAlgorithm = 302																												
Field Length = 1										0	hashIPpayloadOffset = 327																												
Field Length = 4										0	hashIPpayloadSize = 328																												
Field Length = 4										0	hashInitialiserValue = 329																												
Field Length = 4										0	hashOutputRangeMin = 330																												
Field Length = 4										0	hashOutputRangeMax = 331																												
Field Length = 4										0	hashSeletionRangeMin = 332																												
Field Length = 4										0	hashSeletionRangeMax = 333																												
Field Length = 4										0	hashSeletionRangeMin = 332																												
Field Length = 4										0	hashSeletionRangeMax = 333																												
Field Length = 4																																							

Associated IPFIX Data Record:

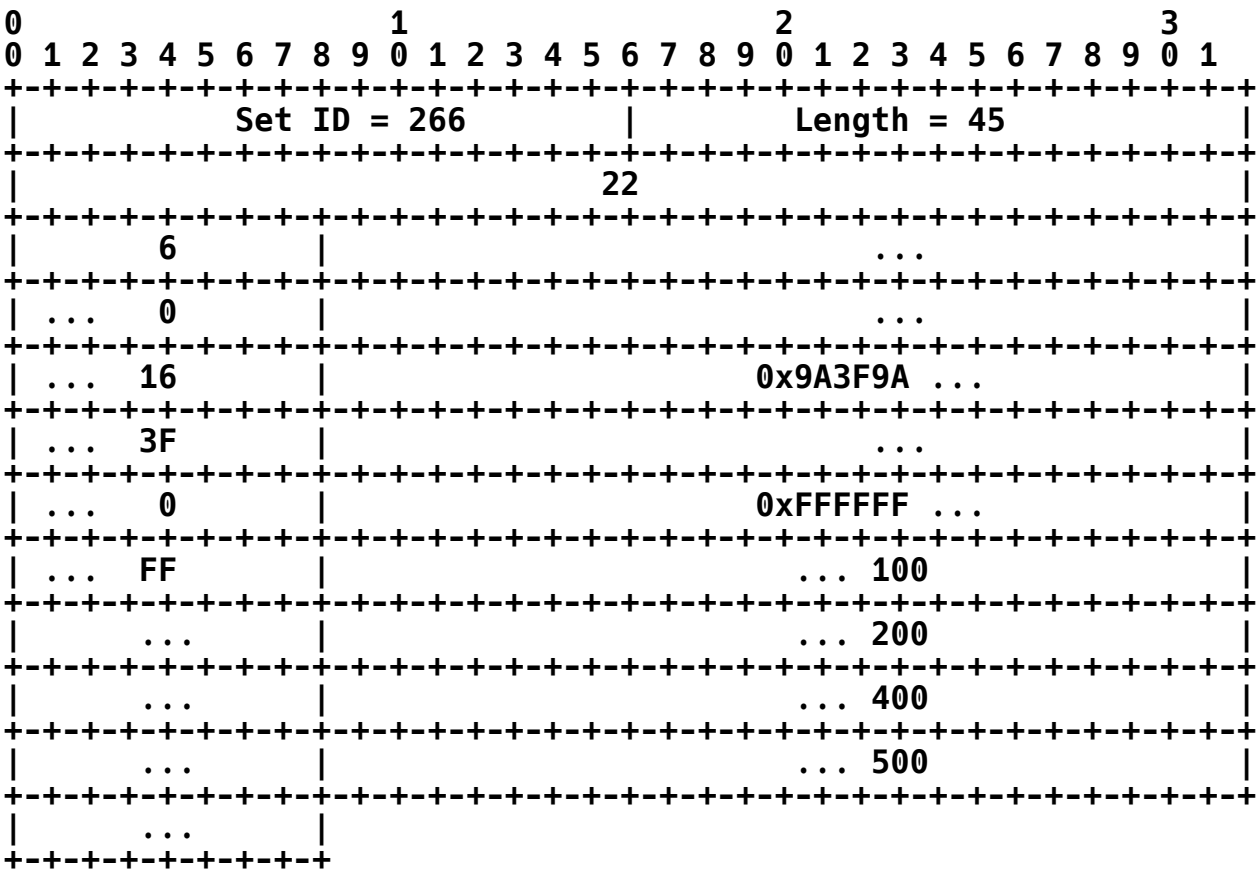


Figure M: Example of the Selector Report Interpretation for Hash-based Filtering

Notes:

* A selectorAlgorithm value of 6 represents Hash-based Filtering using the BOB algorithm.

6.5.2.7. Other Selection Methods

Some potential new selection methods MAY be added. Some of the new selection methods, such as non-uniform probabilistic Sampling and flow-state-dependent Sampling, are described in [RFC5475], with further references.

Each new selection method **MUST** be assigned a unique value for the selectorAlgorithm Information Element. Its configuration parameter(s), along with the way to report it/them with an Options Template, **MUST** be clearly specified.

6.5.3. Selection Sequence Statistics Report Interpretation

A Selector **MAY** be used in multiple Selection Sequences. However, each use of a Selector must be independent, so each separate logical instance of a Selector **MUST** maintain its own individual Selection State and statistics.

The Selection Sequence Statistics Report Interpretation **MUST** include the number of observed packets (Population Size) and the number of packets selected (Sample Size) by each instance of its Primitive Selectors.

Within a Selection Sequence composed of several Primitive Selectors, the number of packets selected for one Selector is equal to the number of packets seen by the next Selector. The order of the Selectors in the Selection Sequence Statistics Report Interpretation **MUST** match the order of the Selectors in the Selection Sequence.

If the full set of statistics is not sent as part of the Basic Packet Reports, the PSAMP Device **MUST** export a Selection Sequence Statistics Report Interpretation for every Selection Sequence, using an Options Template containing the following Information Elements:

Scope:	selectionSequenceId
Non-scope:	packetsObserved
	packetsSelected (first Selector)
	...
	packetsSelected (last Selector)

The packetsObserved Information Element [RFC5477] **MUST** contain the number of packets seen at the Observation Point, and as a consequence passed to the first Selector in the Selection Sequence. The packetsSelected Information Element [RFC5477] contains the number of packets selected by a Selector in the Selection Sequence.

The Attained Selection Fraction for the Selection Sequence is calculated by dividing the number of selected packets (packetsSelected Information Element) for the last Selector by the number of observed packets (packetsObserved Information Element). The Attained Selection Fraction can be calculated for each Selector by dividing the number of packets selected for that Selector by the value for the previous Selector.

The statistics for the whole sequence **SHOULD** be taken at a single logical point in time; the input value for a Selector **MUST** equal the output value of the previous Selector.

The Selection Sequence Statistics Report Interpretation **MUST** be exported periodically.

Example of Selection Sequence Statistics Report Interpretation:

Selection Sequence 7 (Filter->Sampling):

```
Observed    100 (observationPointId 1, Interface 5)
Selected     50 (selectorId 5, match IPV4SourceAddress 192.0.2.1)
Selected      6 (selectorId 10, Sampler: Random one out-of ten)
```

Selection Sequence 9 (Sampling->Filtering):

```
Observed    100 (observationPointId 1, Interface 5)
Selected     10 (selectorId 10, Sampler: Random one out-of ten)
Selected      3 (selectorId 5, match IPV4SourceAddress 192.0.2.1)
```

IPFIX Options Template Record:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Set ID = 3                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Template ID = 267                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Scope Field Count = 1 |0| selectionSequenceId = 301 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Scope 1 Length = 4   |0| packetsObserved = 318 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Field Length = 4     |0| packetsSelected = 319 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Field Length = 4     |0| packetsSelected = 319 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Field Length = 4     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```


to a template, a second scope containing the `templateId` value MUST be added to the Options Template Record. The accuracy SHOULD be reported either with the `absoluteError` Information Element [RFC5477] or with the `relativeError` Information Element [RFC5477].

Accuracy Report Interpretation using the `absoluteError` Information Element:

Scope: `informationElementId`
Non-scope: `absoluteError`

Accuracy Report Interpretation using the `absoluteError` Information Element and a double scope:

Scope: `templateId`
`informationElementId`
Non-scope: `absoluteError`

Accuracy Report Interpretation using the `relativeError` Information Element:

Scope: `informationElementId`
Non-scope: `relativeError`

Accuracy Report Interpretation using the `relativeError` Information Element and a double scope:

Scope: `templateId`
`informationElementId`
Non-scope: `relativeError`

For example, the accuracy of an Information Element whose Abstract Data Type is `dateTimeMilliseconds` [RFC5102], for which the unit is specified as milliseconds, can be specified with the `absoluteError` Information Element with the milliseconds units. In this case, the error interval is the Information Element value +/- the value reported in the `absoluteError`.

For example, the accuracy of an Information Element to estimate the accuracy of a sampled flow, for which the unit would be specified in octets, can be specified with the `relativeError` Information Element with the octet units. In this case, the error interval is the Information Element value +/- the value reported in the `relativeError` times the reported Information Element value.

An alternative to reporting either the `absoluteError` Information Element or the `relativeError` Information Element in the Accuracy Report Interpretation, is to report both. For this case whatever is least accurate for the reported value should be used.

If the accuracy of a reported quantity changes on the Metering Process, a new Accuracy Report Interpretation MUST be generated. The

Collecting Process **MUST** keep the accuracy of the latest Accuracy Report Interpretation.

Example of an Accuracy Report Interpretation using the `absoluteError` Information Element and a double scope: the `timeMicroseconds` contained in the Template 5 has an accuracy of +/- 2 ms, represented by the `absoluteError` Information Element.

Scope: `templateId = 6`
 `informationElementId = timeMicroseconds`
 Non-scope: `absoluteError = 2 ms`

IPFIX Options Template Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 3										Length = 22																													
Template ID = 267										Field Count = 3																													
Scope Field Count = 2										0										templateId = 145																			
Scope 1 Length = 2										0										InformationElementId = 303																			
Scope 2 Length = 2										0										absoluteError = 320																			
Field Length = 4																																							

The associated IPFIX Data Record:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 267										Length = 12																													
5										324																													
2 (encoded as a float32)																																							

Figure 0: Example of the Selection Sequence Statistics Report Interpretation

Notes:

* `absoluteError` is of type `float64` but is compressed down to a `float32` here.

The second example displays an Accuracy Report Interpretation using the `relativeError` Information Element and a single scope: the `timeMicroseconds` has an error of 5%, represented by the `proportionalAccuracy` Information Element.

Scope: `informationElementId = timeMicroseconds`
 Non-scope: `relativeError = 0.05`

IPFIX Options Template Record:

```

      0      1      2      3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Set ID = 3                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Template ID = 268                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Scope Field Count = 1      |0| InformationElementId = 303         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Scope 1 Length = 2         |0|          relativeError= 321       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          Field Length = 4           |                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The associated IPFIX Data Record:

```

      0      1      2      3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Set ID = 267                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     324                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     0.05 ...                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... (encoded as a float32)                                                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

**Figure P: Example of the Selection Sequence
Statistics Report Interpretation**

Notes:

* `relativeError` is of type `float64` but is compressed down to a `float32` here.

7. Security Considerations

As IPFIX has been selected as the PSAMP export protocol and as the PSAMP security requirements are not stricter than the IPFIX security requirements, refer to the IPFIX export protocol [RFC5101] for the security considerations.

In the basic Packet Report, a PSAMP Device exports some number of contiguous bytes from the start of the packet, including the packet header (which includes link layer, network layer, and other encapsulation headers) and some subsequent bytes of the packet payload. The PSAMP Device **SHOULD NOT** export the full payload of conversations, as this would mean wiretapping [RFC2804]. The PSAMP Device **MUST** respect local privacy laws.

8. IANA Considerations

The PSAMP protocol, as set out in this document, has two sets of assigned numbers. Considerations for assigning them are discussed in this section, using the example policies as set out in [RFC5226], "Guidelines for IANA Considerations".

8.1. IPFIX-Related Considerations

As the PSAMP protocol uses the IPFIX protocol, refer to the IANA considerations section in [RFC5101] for the assignments of numbers used in the protocol and for the numbers used in the information model.

8.2. PSAMP-Related Considerations

Each new selection method **MUST** be assigned a unique value for the selectorAlgorithm Information Element [RFC5477]. Initial contents of this registry are found in Section 8.2.1 in [RFC5477]. Its configuration parameter(s), along with the way to report them with an Options Template, **MUST** be clearly specified.

New assignments for the PSAMP selection method will be administered by IANA, on a First Come First Served basis [RFC5226], subject to Expert Review [RFC5226]. The group of experts must double check the Information Elements definitions with already defined Information Elements for completeness, accuracy, and redundancy. These experts will initially be drawn from the Working Group Chairs and document editors of the IPFIX and PSAMP Working Groups.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5101] Claise, B., Ed., "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.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 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.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.

9.2. Informative References

- [RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804, May 2000.
- [RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export" RFC 5470, March 2009.
- [RFC5474] Duffield, N., Ed., "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

10. Acknowledgments

The authors would like to thank the PSAMP group, especially Paul Aitken for fruitful discussions and for proofreading the document several times.

Authors' Addresses

Benoit Claise
Cisco Systems
De Kleetlaan 6a b1
1831 Diegem
Belgium

Phone: +32 2 704 5622
EMail: bclaise@cisco.com

Juergen Quittek
NEC Europe Ltd.
Network Laboratories
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 90511-15
EMail: quittek@nw.neclab.eu

Andrew Johnson
Cisco Systems
96 Commercial Quay
Edinburgh EH6 6LX
Scotland

Phone: +44 131 561 3641
EMail: andrjohn@cisco.com