IPFIX Working Group
Internet-Draft
Intended status: Informational
Expires: November 22, 2007

E. Boschi
Hitachi Europe
L. Mark
Fraunhofer FOKUS
B. Claise
Cisco Systems, Inc.
May 21, 2007

Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports
draft-ietf-ipfix-reducing-redundancy-04.txt

Abstract

This document describes a bandwidth saving method for exporting flow or packet information using the IP Flow Information Export (IPFIX) protocol. As the Packet Sampling (PSAMP) protocol is based on IPFIX, these considerations are valid for PSAMP exports as well.

This method works by separating information common to several flow records from information specific to an individual flow record. Common flow information is exported only once in a data record defined by an option template, while the rest of the specific flow information is associated with the common information via a unique identifier.

Table of Contents

1. Introduction

The IPFIX working group has specified a protocol to export IP Flow information [I-D.ietf-ipfix-protocol]. This protocol is designed to export information about IP traffic flows and related measurement data, where a flow is defined by a set of key attributes (e.g. source and destination IP address, source and destination port, etc.). However, thanks to its template mechanism, the IPFIX protocol can export any type of information, as long as the information element is specified in the IPFIX Information Model [I-D.ietf-ipfix-protocol] or registered with IANA.

Regardless of the fields content, flow records with common properties export the same fields in every single data record. These common properties may represent values common to a collection of flows or packets, or values that are invariant over time. Note that the common properties don't represent the list of flow keys, which are used to define a flow definition: however, the common properties may contain some of the flow keys. The reduction of redundant data from the export stream can result in a significant reduction of the transferred data.

This draft specifies a way to export these invariant or common properties only once, while the rest of the flow specific properties are exported in regular data records. Unique common properties identifiers are used to link data records and the common attributes.

The proposed method is applicable to IPFIX flow and to PSAMP per packet information, without any changes to both the IPFIX and PSAMP protocol specifications.

1.1. IPFIX Documents Overview

The IPFIX Protocol [I-D.ietf-ipfix-protocol] provides network administrators with access to IP flow information. The architecture for the export of measured IP flow information out of an IPFIX exporting process to a collecting process is defined in the IPFIX Architecture [I-D.ietf-ipfix-architecture], per the requirements defined in RFC 3917 [RFC3917]. The IPFIX Architecture [I-D.ietf-ipfix-architecture] specifies how IPFIX data record and templates are carried via a congestion-aware transport protocol from IPFIX exporting processes to IPFIX collecting process. IPFIX has a formal description of IPFIX information elements, their name, type and additional semantic information, as specified in the IPFIX Information Model [I-D.ietf-ipfix-info]. Finally the IPFIX Applicability Statement [I-D.ietf-ipfix-as] describes what type of applications can use the IPFIX protocol and how they can use the information provided. It furthermore shows how the IPFIX framework

relates to other architectures and frameworks.

1.2. PSAMP Documents Overview

The document "A Framework for Packet Selection and Reporting" [I-D.ietf-psamp-framework], 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. The set of packet selection techniques (sampling, filtering, and hashing) supported by PSAMP are described in "Sampling and Filtering Techniques for IP Packet Selection" [I-D.ietf-psamp-sample-tech]. The PSAMP protocol [I-D.ietf-psamp-protocol] specifies the export of packet information from a PSAMP exporting process to a PSAMP collecting process. Like IPFIX, PSAMP has a formal description of its information elements, their name, type and additional semantic information. The PSAMP information model is defined in [I-D.ietf-psamp-info]. Finally [I-D.ietf-psamp-mib] describes the PSAMP Management Information Base.

2. Terminology

The terms in this section are in line with the IPFIX terminology section in the IPFIX [I-D.ietf-ipfix-protocol], and PSAMP [I-D.ietf-psamp-protocol] protocol specifications. Note that this document selected the IPFIX definition of the term Exporting Process [I-D.ietf-ipfix-protocol], as this definition is more generic than the PSAMP definition [I-D.ietf-psamp-protocol].

Observation Point: An Observation Point is a location in the network where IP packets can be observed. Examples include: a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router. Note that every Observation Point is associated with an Observation Domain (defined below), and that one Observation Point may be a superset of several other Observation Points. For example one Observation Point can be an entire line card. That would be the superset of the individual Observation Points at the line card's interfaces.

Observation Domain: An Observation Domain is the largest set of Observation Points for which Flow information can be aggregated by a Metering Process. For example, a router line card may be an Observation Domain if it is composed of several interfaces, each of which is an Observation Point. In the IPFIX Message it generates, the Observation Domain includes its Observation Domain ID, which is unique per Exporting Process. That way, the Collecting Process can identify the specific Observation Domain

from the Exporter that sends the IPFIX Messages. Every Observation Point is associated with an Observation Domain. It is RECOMMENDED that Observation Domain IDs are also unique per IPFIX Device.

IP Traffic Flow or Flow: There are several definitions of the term 'flow' being used by the Internet community. Within the context of IPFIX we use the following definition:

A Flow is defined as a set of IP packets passing an Observation Point in the network during a certain time interval. All packets belonging to a particular Flow have a set of common properties. Each property is defined as the result of applying a function to the values of:

1. one or more packet header field (e.g. destination IP address), transport header field (e.g. destination port number), or application header field (e.g. RTP header fields [RFC3550])

2. one or more characteristics of the packet itself (e.g. number of MPLS labels, etc...)

3. one or more of fields derived from packet treatment (e.g. next hop IP address, the output interface, etc...)

A packet is defined to belong to a Flow if it completely satisfies all the defined properties of the Flow.

This definition covers the range from a Flow containing all packets observed at a network interface to a Flow consisting of just a single packet between two applications. It includes packets selected by a sampling mechanism.

Flow Record: A Flow Record contains information about a specific Flow that was observed at an Observation Point. A Flow Record contains measured properties of the Flow (e.g. the total number of bytes for all the Flow's packets) and usually characteristic properties of the Flow (e.g. source IP address).

Metering Process: The Metering Process generates Flow Records. Inputs to the process are packet headers and characteristics observed at an Observation Point, and packet treatment at the Observation Point (for example the selected output interface).

The Metering Process consists of a set of functions that includes packet header capturing, timestamping, sampling, classifying, and maintaining Flow Records.

The maintenance of Flow Records may include creating new records, updating existing ones, computing Flow statistics, deriving further Flow properties, detecting Flow expiration, passing Flow Records to the Exporting Process, and deleting Flow Records.

Exporting Process: The Exporting Process sends Flow Records to one or more Collecting Processes. The Flow Records are generated by one or more Metering Processes.

Exporter: A device which hosts one or more Exporting Processes is termed an Exporter.

IPFIX Device: An IPFIX Device hosts at least one Exporting Process. It may host further Exporting processes and arbitrary numbers of Observation Points and Metering Process.

Collecting Process: A Collecting Process receives Flow Records from one or more Exporting Processes. The Collecting Process might process or store received Flow Records, but such actions are out of scope for this document.

Template: Template is an ordered sequence of (type, length) pairs, used to completely specify the structure and semantics of a particular set of information that needs to be communicated from an IPFIX Device to a Collector. Each Template is uniquely identifiable by means of a Template ID.

Template Record: A Template Record defines the structure and interpretation of fields in a Data Record.

Data Record: A Data Record is a record that contains values of the parameters corresponding to a Template Record.

Options Template Record: An Options Template Record is a Template Record that defines the structure and interpretation of fields in a Data Record, including defining how to scope the applicability of the Data Record.

Set: Set is a generic term for a collection of records that have a similar structure. In an IPFIX Message, one or more Sets follow the Message Header. There are three different types of Sets: Template Set, Options Template Set, and Data Set.

Template Set: A Template Set is a collection of one or more Template Records that have been grouped together in an IPFIX Message.

Options Template Set: An Options Template Set is a collection of one or more Options Template Records that have been grouped together in an IPFIX Message.

Data Set: A Data Set is one or more Data Records, of the same type, that are grouped together in an IPFIX Message. Each Data Record is previously defined by a Template Record or an Options Template Record.

Information Element: An Information Element is a protocol and encoding independent description of an attribute which may appear in an IPFIX Record. The IPFIX information model [I-D.ietf-ipfix-info] defines the base set of Information Elements for IPFIX. The type associated with an Information Element indicates constraints on what it may contain and also determines the valid encoding mechanisms for use in IPFIX.

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

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.

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

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:

1. the Packet Content;

2. information derived from the packet's treatment at the Observation Point;

3. any selection state that may be maintained by the Selection Process.

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.

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. Examples include field match Filtering, and Hash-based Selection.

Transport Session: In SCTP, the transport session is known as the SCTP association, which is uniquely identified by the SCTP endpoints [RFC2960]; in TCP, the transport session is known as the TCP connection, which is uniquely identified by the combination of IP addresses and TCP ports used; In UDP, the transport session is known as the UDP session, which is uniquely identified by the combination of IP addresses and UDP ports used.

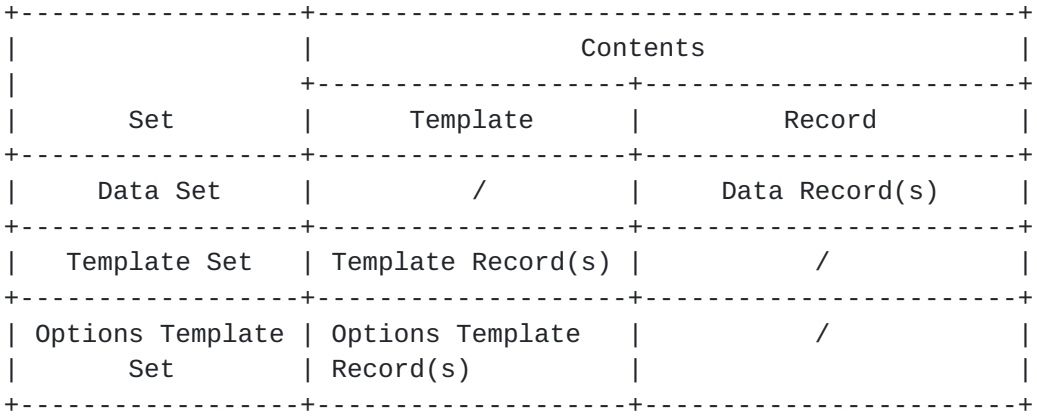
commonPropertiesID: The commonPropertiesID is an identifier of a set of common properties that is locally unique per Observation Domain and Transport Session. Typically, this Information Element is used to link to information reported in separate Data Records. See the IPFIX information model [I-D.ietf-ipfix-info] for the Information Element definition.

Common Properties: Common Properties are a collection of one or more attributes shared by a set of different Flow Records. Each set of Common Properties is uniquely identifiable by means of a commonPropertiesID.

Specific Properties: Specific Properties are a collection of one or more attributes reported in a Flow Record that are not included in the Common Properties defined for that Flow Record.

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].

2.1. Terminology Summary Table



Terminology Summary Table

A Data Set is composed of Data Record(s). No Template Record is included. A Template Record or an Options Template Record defines the Data Record.

A Template Set contains only Template Record(s).

An Options Template Set contains only Options Template Record(s).

2.2. IPFIX Flows versus PSAMP Packets

As described in the PSAMP protocol specification [I-D.ietf-psamp-protocol], the major difference between IPFIX and PSAMP is that the IPFIX protocol exports Flow Records while the PSAMP protocol exports Packet Records. 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 special IPFIX Flow Record containing information about a single packet.

For this document clarity, the term Flow Record represents a generic term expressing an IPFIX Flow Record or a PSAMP packet record, as foreseen by its definition. However, when appropriate, a clear distinction between Flow Record or packet Record will be made.

3. Specifications for bandwidth saving information export

Several Flow Records often share a set of Common Properties. Repeating the information about these Common Properties for every Flow Record introduces a huge amount of redundancy. This document proposes a method to reduce this redundancy.

The PSAMP specifications are used for the export of per-packet information, exporting the specific observed packet in an IPFIX Flow Record. This can be considered as a special Flow Record case, composed of a single packet. Therefore, the method described in this document is also applicable to per packet data reduction, e.g. for export of One Way Delay (OWD) measurements (see Appendix), trajectory sampling, etc.

3.1. Problem Statement and High Level Solution

Consider a set of properties "A", e.g. common sourceAddressA and sourcePortA, equivalent for each Flow Records exported. Figure 2 shows how this information is repeated with classical IPFIX Flow Records, expressing the waste of bandwidth to export redundant

information.

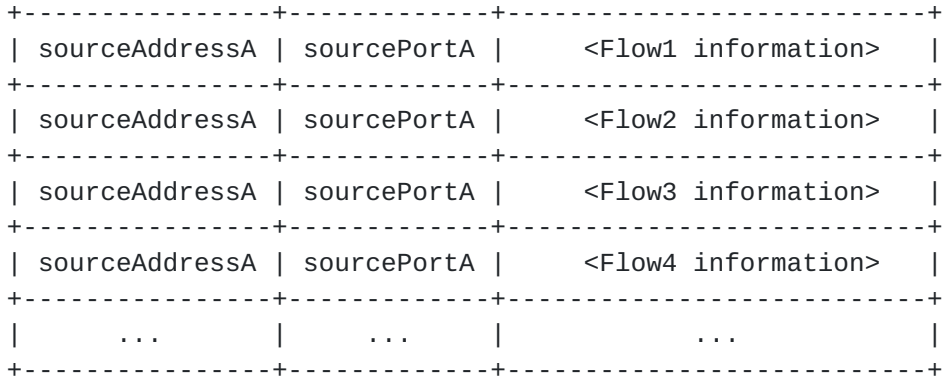
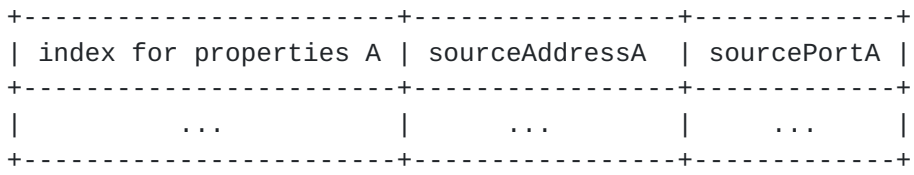


Figure 2: Common and Specific Properties exported together

Figure 3 shows how this information is exported when applying the specifications of this document. The Common Properties are separated from the Specific Properties for each Flow Record. The Common Properties would be exported only once in a specific Data Record (defined by an Option Template), while each Flow Record contains a pointer to the Common Properties A, along with its Flow specific information. In order to maintain the relationship between these sets of properties, we introduce indices (in this case: index for properties A) for the Common Properties that are unique for all Common Properties entries within an Observation Domain. The purpose of the indices is to serve as a "key" identifying "rows" of the Common Properties table. The rows are then referenced by the Specific Properties by using the appropriate value for the Common Properties identifier.



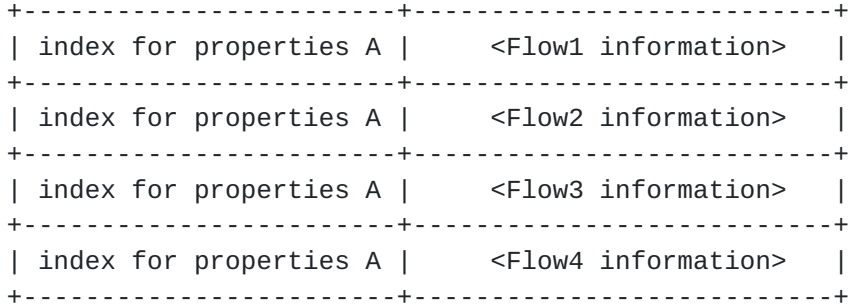


Figure 3: Common and Specific Properties exported separately

This unique export of the Common Properties results in a decrease of the bandwidth requirements for the path between the Exporter and the Collector.

3.2. Data Reduction technique

The IPFIX protocol [I-D.ietf-ipfix-protocol] is Template based. Templates define how data should be exported, describing data fields together with their type and meaning. IPFIX specifies two types of Templates: the Template Record and the Options Template Record. The difference between the two is that the Options Template Record includes the notion of scope, defining how to scope the applicability of the Data Record. The scope, which is only available in the Options Template Record, gives the context of the reported Information Elements in the Data Records. The Template Records and Options Template Records are necessary to decode the Data Records. Indeed, by only looking at the Data Records themselves, this is impossible to distinguish a Data Record defined by Template Record from a Data Record defined by an Option Template Record. To export information more efficiently, this specification proposes to group Flow Records by their common properties. We define Common Properties as a collection of attributes shared by a set of different Flow Records.

An implementation using the proposed specification MUST follow the IPFIX transport protocol specifications defined in the IPFIX protocol [I-D.ietf-ipfix-protocol].

As explained in Figure 4, the information is split into two parts, using two different Data Records. Common Properties MUST be exported via Data Records defined by an Option Template Record. Like Template Records, they MUST be sent only once per SCTP association or TCP connection, and MUST be sent reliably via SCTP if SCTP is the transport protocol. These properties represent values common to several Flow Records (e.g. IP source and destination address). The Common Properties Data Records MUST be sent prior to the corresponding Specific Properties Data Records. The Data Records reporting Specific Properties MUST be associated with the Data Records reporting the Common Properties using a unique identifier for the Common Properties, the commonPropertiesID Information Element [I-D.ietf-ipfix-info]. The commonPropertiesID MUST be exported as the scope in the Options Template Record, and also exported in the associated Template Record.

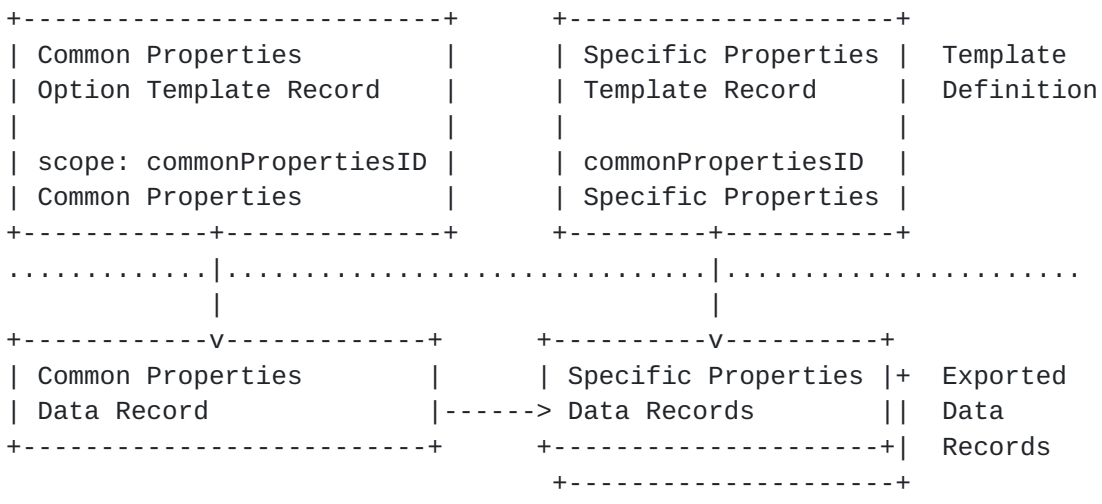


Figure 4: Template Record and Data Record dependencies

From the IPFIX protocol, there are no differences between the per Flow or per packet data reduction, except maybe the terminology where the Specific Properties could be called packet Specific Properties in the previous figure.

4. Transport Protocol Choice

This document follows the IPFIX transport protocol specifications defined in the IPFIX protocol [I-D.ietf-ipfix-protocol]. However, depending on the transport protocol choice, this document imposes some more constraints. If PR-SCTP is selected as the IPFIX protocol, the SCTP sub-section specifications MUST be respected. If UDP is selected as the IPFIX protocol, the UDP sub-section specifications MUST be respected. If TCP is selected as the IPFIX protocol, the TCP sub-section specifications MUST be respected.

4.1. PR-SCTP

The active Common Properties MUST be sent after the SCTP association establishment before the corresponding Specific Properties Data Records. In case of SCTP association re-establishment, all active Common Properties MUST be re-sent before the corresponding Specific Properties Data Records.

The Common Properties Data Records MUST be sent reliably.

4.2. UDP

Common Properties Data Records MUST be re-sent at regular intervals, whose frequency MUST be configurable. The default value for the frequency of Common Properties transmission (refresh timeout) is 10 minutes.

The Exporting Process SHOULD transmit the Common Properties definition in advance of any Data Record that use these Common Properties, to help ensure that the Collector has the Common Properties definition before receiving the first associated Data Record.

If a commonPropertiesID is not used anymore the Exporting Process stops re-sending the related Common Properties Data Record. The old commonPropertiesID MUST NOT be used until its lifetime (see Section 6.1) has expired.

4.3. TCP

Common Properties MUST be sent after the TCP connection establishment before the corresponding Specific Properties Data Records. In case of TCP connection re-establishment, all active Common Properties MUST be re-sent before the corresponding Specific Properties Data Records.

5. commonPropertiesID Management

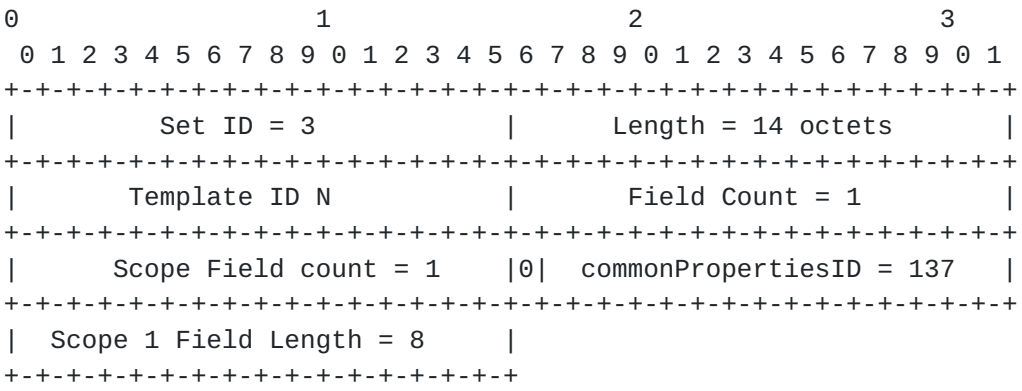
The commonPropertiesId is an identifier of a set of common properties that is locally unique per Observation Domain and Transport Session. The Exporting Process MUST manage the commonPropertiesIDs allocations for its Observation Domains and Transport Session. Different Observation Domains from the same Exporter MAY use the same commonPropertiesID value to refer to different sets of Common Properties.

The commonPropertiesID values MAY be assigned sequentially, but it is NOT REQUIRED. Particular commonPropertiesID ranges or values MAY have explicit meanings for the IPFIX Device. For example, commonPropertiesID values may be assigned based on the result of a hash function, etc...

Using a 64 bit commonPropertiesID Information Element allows the export of 2**64 active sets of Common Properties, per Observation Domain and per Transport Session.

commonPropertiesIDs that are not used anymore SHOULD be withdrawn. The Common Properties ID withdrawal message is a Data Record defined

by an Option Template consisting of only one scope field namely the commonPropertiesID (with a type of 137 [I-D.ietf-ipfix-info]) and no non-scope fields.



commonPropertiesID Withdrawal Message

If UDP is selected as the transport protocol, the commonPropertiesID Template Withdraw Messages MUST not be used, as this method is inefficient due to the unreliable nature of UDP.

6. The Collecting Process Side

This section describes the Collecting Process when using SCTP and PR-SCTP as the transport protocol. Any necessary changes to the Collecting Process specifically related to TCP or UDP transport protocols are specified in the subsections.

The Collecting Process MUST store the commonPropertiesId information for the duration of the association so that it can interpret the corresponding Data Records that are received in subsequent Data Sets. The Collecting Process can either store the Data Records as they arrive, without reconstructing the initial Flow Record, or reconstruct the initial Flow Record. In the former case, there might be less storage capacity required at the Collector side. In the latter case, the collector job is more complex and time-consuming due to the higher resource demand for record processing in real time.

If the Collecting Process has received the Specific Properties Data Record before the associated Common Properties Data Record, the Collecting Process SHOULD store the Specific Properties Data Record and await the retransmission or out-of-order arrival of the Common Properties Data Record.

Common Properties IDs are unique per SCTP association and per Observation Domain. If the Collecting Process receives a Common Properties ID which has already been received but which has not previously been withdrawn (i.e. a commonPropertiesID from the same Exporter Observation Domain received on the SCTP association), then the Collecting Process MUST shutdown the association.

When an SCTP association is closed, the Collecting Process MUST discard all Common Properties IDs received over that association and stop decoding IPFIX Messages that use those Common Properties IDs.

If a Collecting Process receives a Common Properties Withdrawal message, the Collecting Process MUST delete the corresponding Common Properties associated with the specific SCTP association and specific Observation Domain, and stop interpreting Data Records referring to those Common Properties. The receipt of Data Records referring to Common Properties that have been withdrawn MUST be ignored and SHOULD be logged by the Collecting Process.

If the Collecting Process receives a Common Properties Withdrawal message for a Common Properties that it has not received before on this SCTP assocation, it MUST reset the SCTP association, discard the IPFIX Message, and SHOULD log the error as it does for malformed IPFIX Messages.

6.1. UDP

The Collecting Process MUST associate a lifetime with each Common Property received via UDP. Common Properties not refreshed by the Exporting Process within the lifetime are expired at the Collecting Process.

If the Common Properties are not refreshed before that lifetime has expired, the Collecting Process MUST discard the corresponding definition of the commonPropertiesID and any current and future associated Data Records. In which case, an alarm MUST be logged.

The Collecting Process MUST NOT decode any further Data Records which are associated with the expired Common Properties. If a Common Property is refreshed with a definition that differs from the previous definition, the Collecting Process SHOULD log a warning and replace the previous received Common Property with the new one. The Common Property lifetime at the Collecting Process MUST be at least 3 times higher than the refresh timeout of the Template used to export the Common Property definition, configured on the Exporting Process.

The Collecting Process SHOULD accept Data Records without the associated Common Properties required to decode the Data Record. If

the Common Properties have not been received at the time Data Records are received, the Collecting Process SHOULD store the Data Records for a short period of time and decode them after the Common Properties definitions are received. The short period of time MUST be lower than the lifetime of definitions associated with identifiers considered unique within the UDP session.

6.2. TCP

When the TCP connection is reset, either gracefully or abnormally, the Collecting Processes MUST delete all commonPropertiesID values and associated Common Properties data corresponding to that connection.

If a Collection Process receives a commonPropertiesID Withdraw message, the Collection Process MUST expire the related Common Properties data.

7. Advanced Techniques

7.1. Multiple Data Reduction

A Flow Record can refer to one or more Common Properties sets; the use of multiple Common Properties can lead to more efficient exports. When sets of Common Properties are identified in the data, it may be found that there is more than one set of non-overlapping properties.

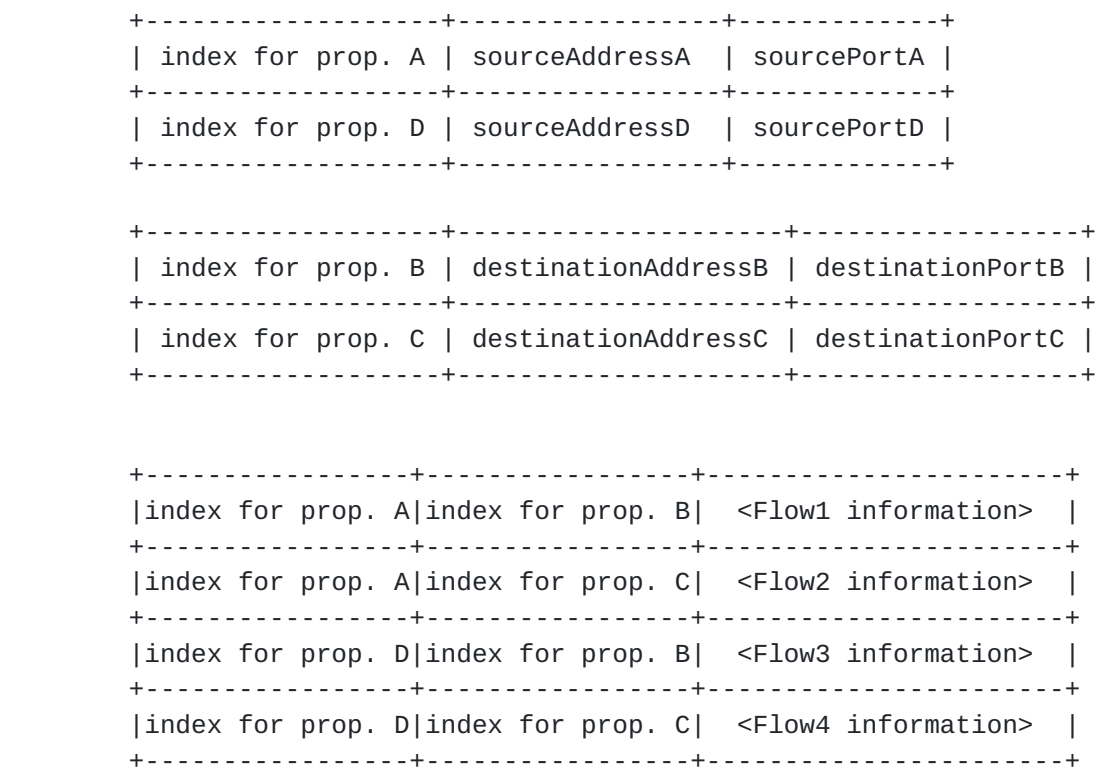
Note that in the case of multiple Common Properties in one Data Record, the different sets of Common Properties MUST be disjoint (i.e. MUST NOT have Information Elements in common), to avoid potential collisions.

Consider a set of properties "A", e.g. common sourceAddressA and sourcePortA, and another set of properties "B", e.g. destinationAddressB and destinationPortB. Figure 6 shows how this information is repeated with classical IPFIX export in several Flow Records.

Figure 6: Common and Specific Properties exported together

Besides that other sets of Properties might be repeated as well (e.g. properties C and D in the figure above).

We can separate the Common Properties into the properties A composed of sourceAddressA and sourcePortA, properties D composed of sourceAddressD and sourcePortD, and into the properties B composed of destinationAddressB and destinationPortB and properties C composed of destinationAddressC and destinationPortC,. These four records can be expanded to four combinations of Data Records to reduce redundancy without the need to define four complete sets of Common Properties (see the figure below). The more Common Properties sets are defined, the more combinations are available.



Multiple Common (above) and Specific Properties (below) exported separately

The advantage of the multiple Common Properties is that the objective of reducing the bandwidth is met while the number of indexes is kept to a minimum. Defining an extra index for all records would not bring to save bandwidth in the case of Figure 6 and is generally a less efficient solution.

If a set of Flow Records share multiple sets of Common Properties, multiple commonPropertiesID instances MAY be used to increase export efficiency even further, as displayed in Figure 8.

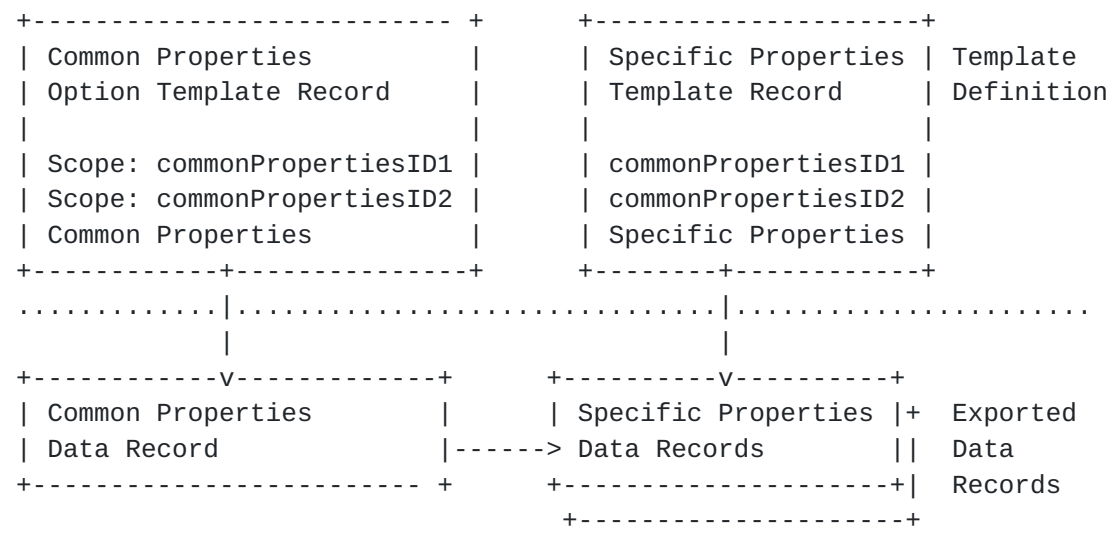


Figure 8: Multiple Data Reduction

7.2. Cascading Common Properties

An Exporting Process MUST NOT export any set of Common Properties which contains, either directly or via other cascaded Common Properties, references to itself in its own definition (i.e., a circular definition). When the Collecting Process receives Common Properties that reference other Common Properties, it MUST resolve the references to Common Properties. If the Common Properties aren't available at the time Data Records are received, the Collecting Process SHOULD store the Data Records for a short period of time and decode them after the Common Properties are received.

If the Collecting Process could not decode a cascading Common Properties definition because the referenced Common Properties are not available before the short period of time, then the Collecting Process SHOULD log the error.

If the Collecting Process could not decode a cascading Common Properties definition because it detects a circular definition, then the Collecting Process SHOULD log the error.

Information Element ordering MUST be preserved when creating and expanding Common Properties.

8. Export and Evaluation Considerations

The objective of the method specified in this document is the

reduction in the amount of measurement data that has to be transferred from the Exporter to the Collector. Note that the efficiency of this method may vary, as discussed in this section. In addition there might be less storage capacity required at the Collector side if the Collector decides to store the Data Records as they arrive, without reconstructing the initial Flow Record.

On the other hand, this method requires additional resources on both the Exporter and the Collector. The Exporter has to manage Common Properties information and to assign commonPropertiesId values. The Collector has to process records described by two templates instead of just one. Additional effort is also required when post processing the measurement data, in order to correlate Flow Records with Common Properties information.

8.1. Transport Protocol Choice

The proposed method is most effective using a reliable transport protocol for the transfer of the Common Properties. Therefore the use of PR-SCTP with the reliable mode or TCP is recommended. However, if the path from the Exporting Process to the Collecting Process is not fully reliable, the SCTP or TCP retransmission might reduce the benefits of this specification. If the path from the Exporting Process to the Collecting Process is full reliable, the use of UDP is less effective because the Common Properties have to be re-sent regularly.

8.2. Reduced Size Encoding

The transfer of the commonPropertiesIDs originates some overhead and might even increase the amount of exported data if the length of the commonPropertiesID field is not shorter than the length of the replaced fields.

In cases where the range of the commonPropertiesID can be restricted, it is RECOMMENDED to apply reduced-size encoding to the commonPropertiesID, to achieve a further bandwidth efficiency gain.

8.3. Efficiency Gain

While the goal of this specification is to reduce the bandwidth, the efficiency might be limited. Indeed, the efficiency gain is based on the numerous redundant information in Flows and would be directly proportional to the re-use of the defined commonPropertiesID values (In other words, the more we re-use a commonPropertiesID value, the better the efficiency gain), with a theoretical limit where all the Data Records would use a single commonPropertiesID. While the Exporting Process can evaluate the direct gain for the Flow Records

to be exported, it cannot predict whether future Flow Records would contain the information specified by active commonPropertiesID values. This implies that the efficiency factor of this specification is higher for specific applications where filtering is involved, such as one-way delay or trajectory sampling.

Note that this technique might even lead to an increase in bandwidth usage under certain conditions. Taking into account the overhead of exporting the commonPropertiesID values, if the commonPropertiesID values are not used in future Data Records, this technique would actually increase the export bandwidth. A typical case would be the assignments of Common Properties based on past observed traffic, hoping that future Flows would contain the same characteristics.

The efficiency gain depends also on the difference between the length of the replaced fields and the length of the commonPropertiesID. The shorter is the length of commonPropertiesID with respect to the total length of the Common Properties fields, the bigger is the gain.

The example in section Appendix A.2 below uses IPFIX to export measurement data for each received packet. In that case, for a Flow of 1000 packets the amount of data can be decreased more than 26 percent.

9. IANA Considerations

This document has no actions for IANA.

10. Security Considerations

The same security considerations as for the IPFIX Protocol [I-D.ietf-ipfix-protocol] apply.

11. Acknowledgments

The authors would like to thank Guido Pohl for initiating this work and for his contribution to early versions of this document. Thanks also to Andrew Johnson, Gehrard Muenz, Brian Trammell and Paul Aitken for their comments and feedback.

12. References

12.1. Normative References

[I-D.ietf-ipfix-protocol]
Claise, B., "Specification of the IPFIX Protocol for the Exchange", draft-ietf-ipfix-protocol-24 (work in progress), November 2006.

[I-D.ietf-ipfix-info]
Quittek, J., "Information Model for IP Flow Information Export", draft-ietf-ipfix-info-15 (work in progress), February 2007.

[I-D.ietf-psamp-protocol]
Claise, B., "Packet Sampling (PSAMP) Protocol Specifications", draft-ietf-psamp-protocol-07 (work in progress), October 2006.

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

12.2. Informative References

[I-D.ietf-ipfix-as]
Zseby, T., "IPFIX Applicability", draft-ietf-ipfix-as-11 (work in progress), February 2007.

[I-D.ietf-ipfix-architecture]
Sadasivan, G., "Architecture for IP Flow Information Export", draft-ietf-ipfix-architecture-12 (work in progress), September 2006.

[I-D.ietf-psamp-info]
Dietz, T., "Information Model for Packet Sampling Exports", draft-ietf-psamp-info-05 (work in progress), October 2006.

[I-D.ietf-psamp-sample-tech]
Zseby, T., "Sampling and Filtering Techniques for IP Packet Selection", draft-ietf-psamp-sample-tech-07 (work in progress), July 2005.

[I-D.ietf-psamp-mib]
Dietz, T. and B. Claise, "Definitions of Managed Objects for Packet Sampling", draft-ietf-psamp-mib-06 (work in progress), June 2006.

[I-D.ietf-psamp-framework]
Duffield, N., "A Framework for Packet Selection and

Reporting", draft-ietf-psamp-framework-11 (work in progress), May 2007.

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

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

Appendix A. Examples

A.1. Per Flow Data Reduction

In this section we show how Flow information can be exported efficiently using the method described in this draft. Let's suppose we have to periodically export data about two IPv6 Flows.

In this example we report the following information:

Flow	dstIPv6Address	dst-Port	nPkts	nBytes
A	2001:DB8:80AD:5800:0058:0800:2023:1D71	80	30	6000
A	2001:DB8:80AD:5800:0058:0800:2023:1D71	80	50	9500
B	2001:DB8:80AD:5800:0058:00AA:00B7:AF2B	1932	60	8000
A	2001:DB8:80AD:5800:0058:0800:2023:1D71	80	40	6500
A	2001:DB8:80AD:5800:0058:0800:2023:1D71	80	60	9500
B	2001:DB8:80AD:5800:0058:00AA:00B7:AF2B	1932	54	7600

The Common Properties in this case are the destination IPv6 address and the destination port. We first define an Option Template that

contains the following Information Elements:

- Scope: commonPropertiesID in [I-D.ietf-ipfix-info], with a type of 137 and a length of 8 octets.

- The destination IPv6 address: destinationIPv6Address in [I-D.ietf-ipfix-info], with a type of 28 and a length of 16 octets.

- The destination port: destinationTransportPort in [I-D.ietf-ipfix-info], with a type of 11, and a length of 2 octets

Figure 10 shows the Option template defining the Common Properties with commonPropertiesID as scope:

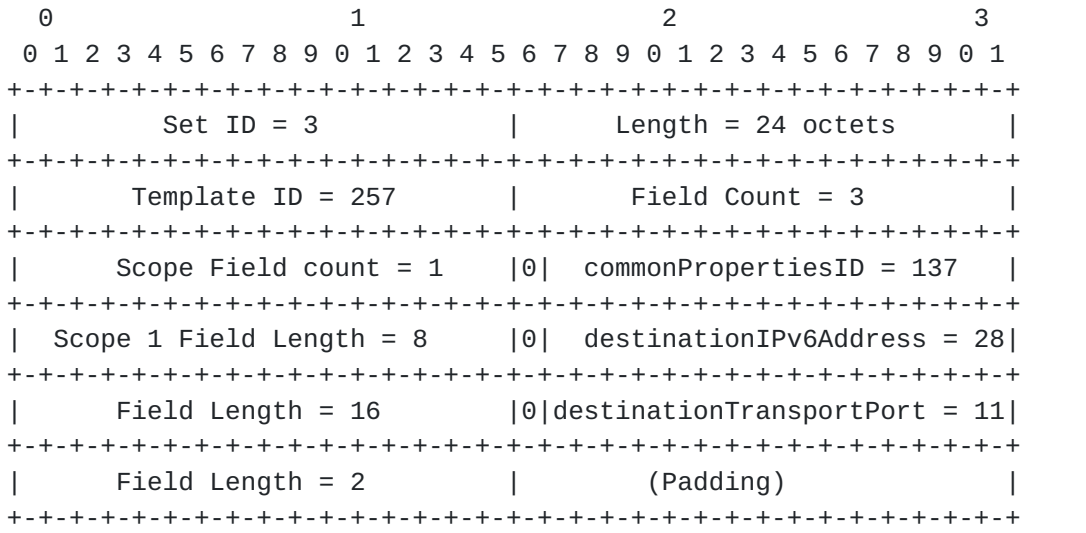


Figure 10: Common Properties Option Template

The Specific Properties Template consists of the information not contained in the Option Templates, i.e. Flow specific information, n this case the number of packets and the number of bytes to be reported. Additionally, this Template contains the commonPropertiesID. In Data Records, the value of this field will contain one of the unique indices of the Option Records exported before. It contains the following Information Elements (see also Figure 11):

- commonPropertiesID with a length of 8 octets

- The number of packets of the Flow: inPacketDeltaCount in [I-D.ietf-ipfix-info], with a length of 4 octets.

o The number of octets of the Flow: inOctetDeltaCount in [I-D.ietf-ipfix-info], with a length of 4 octets

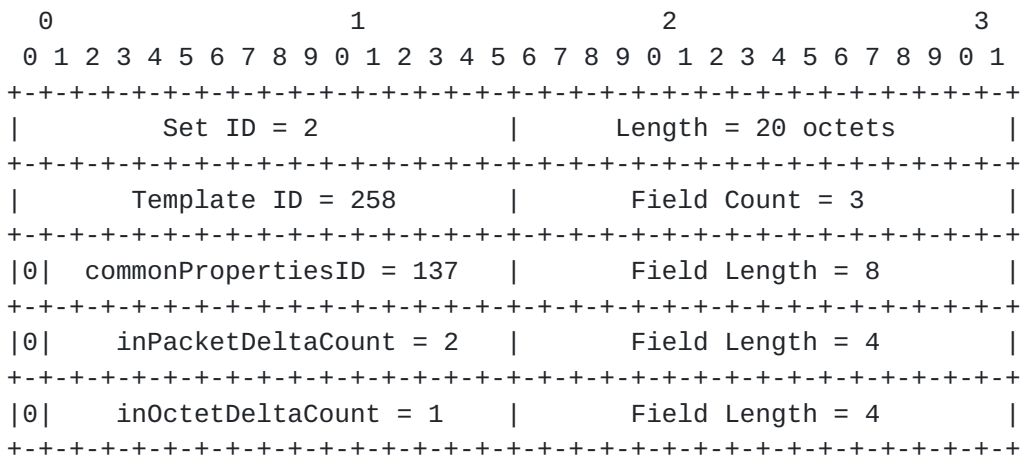


Figure 11: Specific Properties Template

Considering the data shown at the beginning of this example, the following two Data Records will be exported:

Common-PropertiesID	dstAddress	dst-Port
101	2001:DB8:80AD:5800:0058:0800:2023:1D71	80
102	2001:DB8:80AD:5800:0058:00AA:00B7:AF2B	1932

The Data Records reporting the Common Properties will look like:

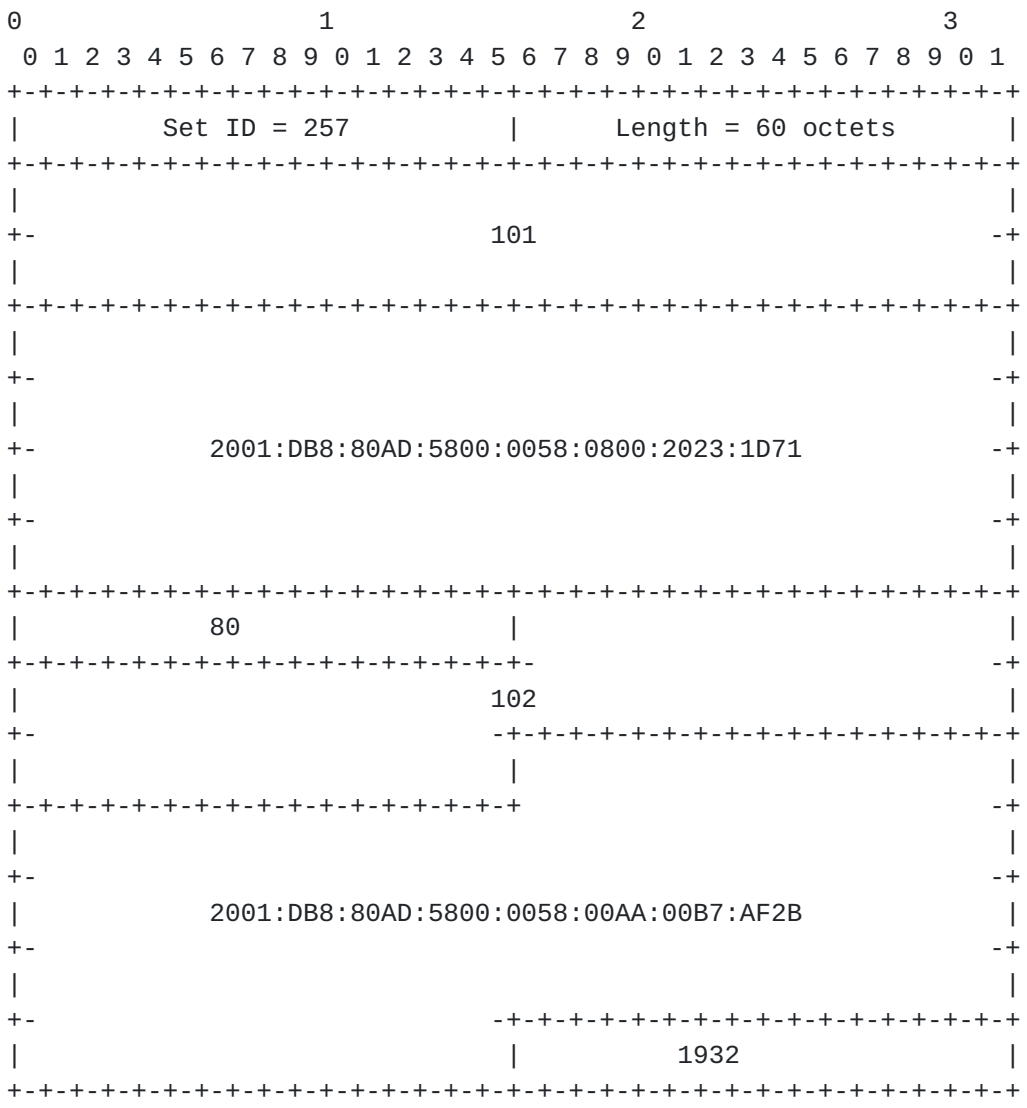


Figure 13: Data Records reporting Common Properties

The Data Records will in turn be:

commonPropertiesID	inPacketDeltaCount	inOctetDeltaCount
101	30	6000
101	50	9500
102	60	8000
101	40	6500
101	60	9500
102	54	7600

Figure 15 shows the first Data Record listed in the table:

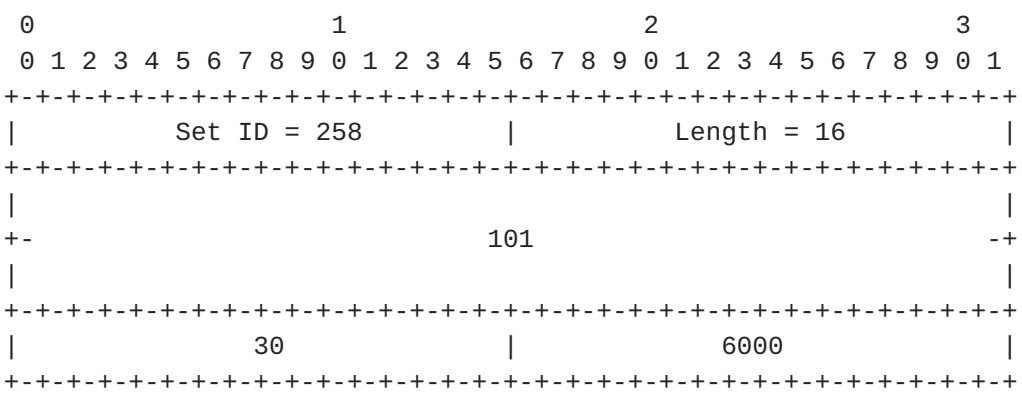


Figure 15: Data Records reporting Common Properties

A.2. Per Packet Data Reduction

An example of the per packet data reduction is the measurement of One-Way Delay (OWD), where the exact same specific packet must be observed at the source and destination of the path to be measured. By subtracting the time of observation of the same packet at the two end points with synchronized clocks, the OWD is computed. As the OWD is measured for a specific application on which a Service Level Agreement (SLA) is bound, this translates into the observation of multiple packets with Specific Properties, results of filtering. In order to match the identical packet at both Observation Points, a series of packets with a set of properties (For example, all the packets of a specific source and destination IP addresses, of a specific DSCP value, and of a specific destination transport port) must be observed at both ends of the measurements. This implies that the source and destination must export of a series of Flow Records composed of two types of information: some common information for all packets, and some unique information about packets in order to generate a unique identifier for each packet passing this Observation Point (for example, a hash value on the invariant fields of the packet). So, the source and destination composing the measurements end points can individually and independently apply the redundancy technique described in this draft in order to save some bandwidth for their respective Flow Records export.

The Templates required for exporting measurement data of this kind are illustrated in the figures below. Figure 16 shows the Option Template containing the information concerning Flows using the commonPropertiesID as scope. In the Common Properties Template we export the following Information Elements:

- The source IPv4 Address: sourceIPv4Address in [I-D.ietf-ipfix-info], with a type of 8 and a length of 4 octets.

- The destination IPv4 Address: destinationIPv4Address in [I-D.ietf-ipfix-info], with a type of 12 and a length of 4 octets.

- The Class of Service field: ClassOfServiceIPv4 in [I-D.ietf-ipfix-info], with a type of 5 and a length of 1 octet

- The Protocol Identifier: protocolIdentifier in [I-D.ietf-ipfix-info], with a type of 4 and a length of 1 octet

- The source port: sourceTransportPort in [I-D.ietf-ipfix-info], with a type of 7 and a length of 2 octets.

- The destination port: destinationTransportPort in [I-D.ietf-ipfix-info], with a type of 11 and a length of 2 octets.

The commonPropertiesID Information Element is used as the Scope Field.

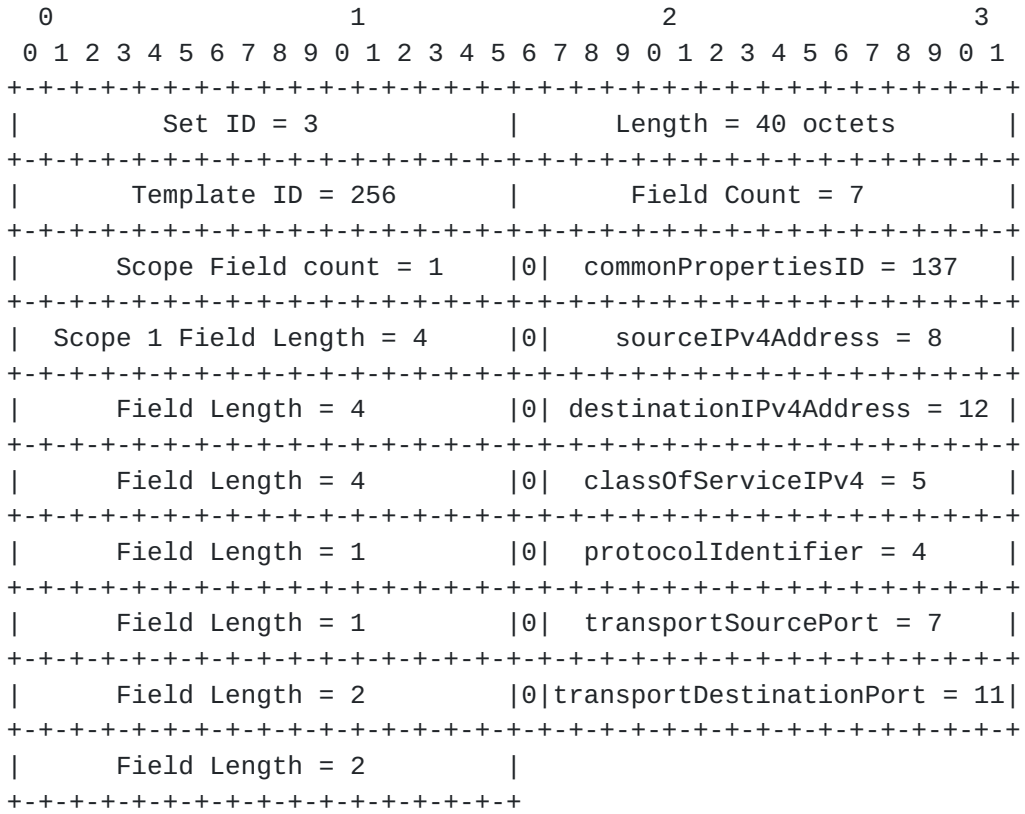


Figure 16: Example Flow Properties Template

For passive One Way Delay measurement, the Packet Properties Template, or Specific Properties Template, consists of at least Timestamp and Packet ID. Additionally, this template contains a commonPropertiesId field to associate the packet with a Flow.

Figure 17 displays the template with the packet properties. In this example we export the following Information Elements:

- commonPropertiesID. In this case reduced size encoding is used, and the Information Element is declared with a length of 4 octets instead of 8.

- The packet timestamp: observationTimeMilliseconds in the PSAMP Information Model [I-D.ietf-psamp-info], with a type of 323 and a length of 8 octets.

- digestHashValue in the PSAMP Information Model [I-D.ietf-psamp-info], with a type of 326 and a length of 8 octets

- The packet length: ipTotalLength in the IPFIX Information Model [I-D.ietf-ipfix-info], with a type of 224 and a length of 8 octets

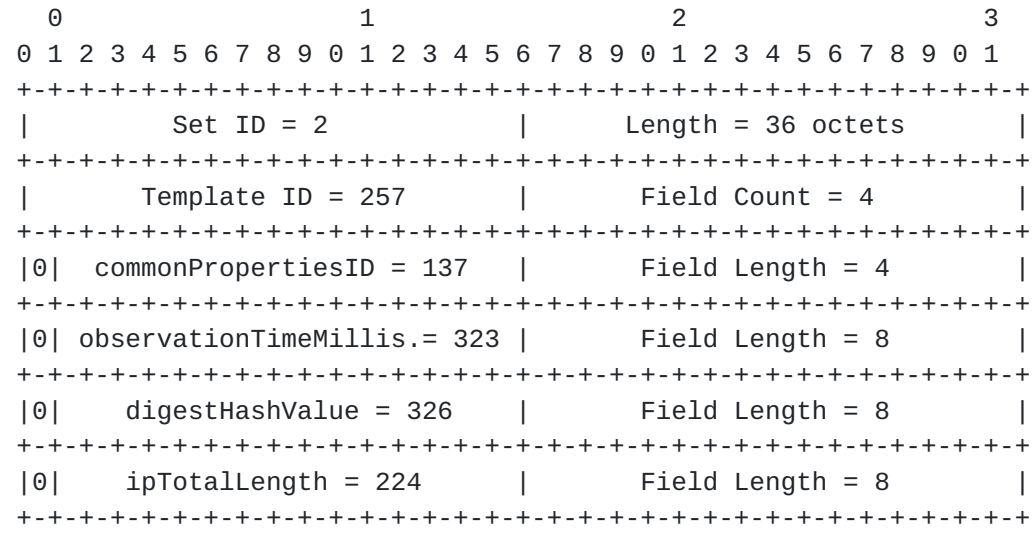


Figure 17: Example Packet Properties Template

At the collection point, packet records from the two measurement points are gathered and correlated by means of the packet ID. The resulting delay data records are exported in a similar manner as the packet data. One-way delay data is associated with Flow information by the commonPropertiesId field. The OWD properties contain the

Packet Pair ID (which is the packet ID of the two contributing packet records), the timestamp of the packet passing the reference monitor point in order to reconstruct a time series, the calculated delay value, and the commonPropertiesID.

In this example using IPFIX to export the measurement data for each received packet 38 bytes have to be transferred (sourceAddressV4=4, destinationAddressV4=4, classOfServiceV4=1, protocolIdentifier=1, sourceTransportPort=2, destionationTransportPort=2, observationTimeMilliseconds=8, digestHashValue=8, ipTotalLength=8). Without considering the IPFIX protocol overhead a Flow of 1000 packets produces 38000 bytes of measurement data. Using the proposed optimization each packet produces an export of only 28 bytes (observationTimeMilliseconds=8, digestHashValue=8, ipTotalLength=8, commonPropertiesID=4). The export of the Flow information produces 18 bytes (sourceAddressV4=4, destinationAddressV4=4, classOfServiceV4=1, protocolIdentifier=1, sourceTransportPort=2, destionationTransportPort=2, commonPropertiesID=4). For a Flow of 1000 packets this sums up to 28018 bytes. This is a decrease of more than 26 percent.

A.3. commonPropertiesID Template Withdrawal Message

This section shows an example commonPropertiesID Withdrawal message. Figure 18 depicts the Option Template Record with the commonPropertiesID as unique scope field, and no non-scope fields.

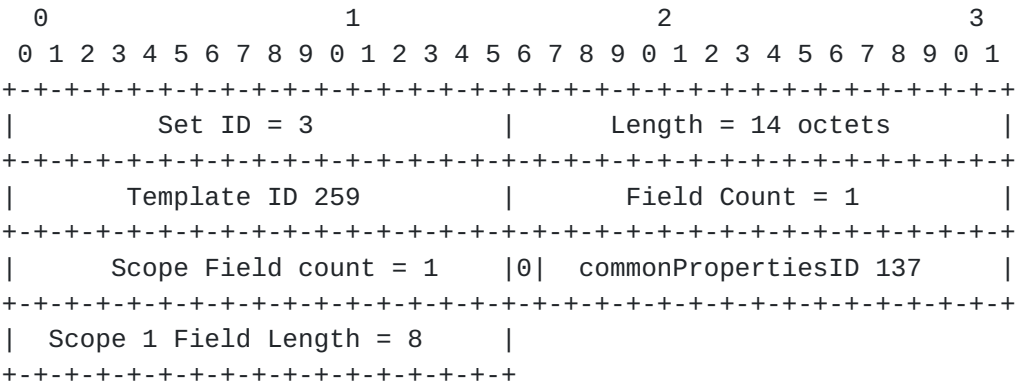


Figure 18: example commonPropertiesID withdrawal template

Figure 19 shows the Option Data Record withdrawing commonPropertiesID N:

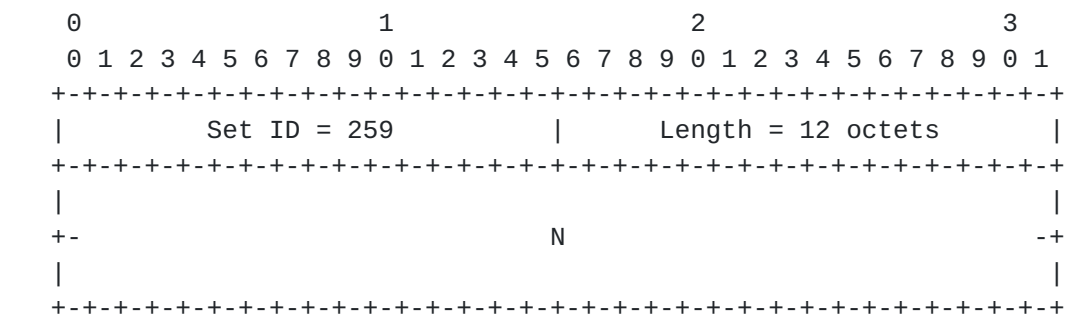


Figure 19: commonPropertiesID withdrawal record, withdrawing commonPropertiesID N

Authors' Addresses

Elisa Boschi
Hitachi Europe SAS
Immeuble Le Theleme
1503 Route les Dolines
06560 Valbonne
France

Phone: +33 4 89874100
Email: elisa.boschi@hitachi-eu.com

Lutz Mark
Fraunhofer FOKUS
Kaiserin Augusta Allee 31
10589 Berlin
Germany

Phone: +49 30 34637306
Email: mark@fokus.fraunhofer.de

Benoit Claise
Cisco Systems
De Kleetlaan 6a b1
Diegem 1813
Belgium

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

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).