        **Reducing Redundancy in IP Flow Information Export (IPFIX)
                 and Packet Sampling (PSAMP) Reports**

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Copyright Notice

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 Options 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 [RFC5101]. 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 [RFC5101] or registered with IANA.

   Regardless of the fields' contents, 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 document 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 [RFC5101] 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 [RFC5470],
   per the requirements defined in RFC 3917 [RFC3917]. The IPFIX
   Architecture [RFC5470] specifies how IPFIX Data Records and templates
   are carried via a congestion-aware transport protocol from IPFIX
   exporting processes to IPFIX collecting processes. IPFIX has a
   formal description of IPFIX Information Elements, their names, types,
   and additional semantic information, as specified in the IPFIX
   information model [RFC5102]. Finally, the IPFIX applicability
   statement [RFC5472] 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"
   [RFC5474] 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 is described in "Sampling and Filtering
   Techniques for IP Packet Selection" [RFC5475]. The PSAMP protocol
   [RFC5476] 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 names,
   types, and additional semantic information. The PSAMP information
   model is defined in [RFC5477]. Finally, [PSAMP-MIB] describes the
   PSAMP Management Information Base.

2. Terminology

   IPFIX-specific terminology used in this document is defined in
   Section 2 of the IPFIX protocol specification [RFC5101] and Section 3
   of the PSAMP protocol specification [RFC5476]. As in [RFC5101] and
   [RFC5476], these IPFIX-specific terms have the first letter of a word
   capitalized when used in this document.

   In addition, the following new terms are defined in this document:

   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 [RFC5102] 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

```
+-----------------+-----------------------------------------------+
|                 |                    Contents                   |
|                 +-------------------+---------------------------+
|      Set        |     Template      |          Record           |
+-----------------+-------------------+---------------------------+
|    Data Set     |         /         |      Data Record(s)       |
+-----------------+-------------------+---------------------------+
|  Template Set   | Template Record(s)|             /             |
+-----------------+-------------------+---------------------------+
| Options Template| Options Template  |             /             |
|      Set        | Record(s)         |                           |
+-----------------+-------------------+---------------------------+
```

                     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 [RFC5476], 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 a
   special IPFIX Flow Record containing information about a single
   packet.

   For this document's 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 Record exported. Figure 1
shows how this information is repeated with classical IPFIX Flow
Records, expressing the waste of bandwidth to export redundant
information.

```
+----------------+-------------+--------------------------+
| sourceAddressA | sourcePortA |     <Flow1 information>   |
+----------------+-------------+--------------------------+
| sourceAddressA | sourcePortA |     <Flow2 information>   |
+----------------+-------------+--------------------------+
| sourceAddressA | sourcePortA |     <Flow3 information>   |
+----------------+-------------+--------------------------+
| sourceAddressA | sourcePortA |     <Flow4 information>   |
+----------------+-------------+--------------------------+
|      ...       |     ...     |           ...            |
+----------------+-------------+--------------------------+
```

        Figure 1: Common and Specific Properties Exported Together

Figure 2 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 Options 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, the 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.

```
+------------------------+----------------+------------+
| index for properties A | sourceAddressA | sourcePortA |
+------------------------+----------------+------------+
|           ...          |      ...       |    ...     |
+------------------------+----------------+------------+
```

```
+------------------------+--------------------------+
| index for properties A |    <Flow1 information>   |
+------------------------+--------------------------+
| index for properties A |    <Flow2 information>   |
+------------------------+--------------------------+
| index for properties A |    <Flow3 information>   |
+------------------------+--------------------------+
| index for properties A |    <Flow4 information>   |
+------------------------+--------------------------+
```

     Figure 2: 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 [RFC5101] 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, it is impossible to distinguish a Data
   Record defined by Template Record from a Data Record defined by an
   Options 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
   [RFC5101].

As explained in Figure 3, the information is split into two parts,
using two different Data Records. Common Properties MUST be exported
via Data Records defined by an Options 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
[RFC5102]. The commonPropertiesID Information Element MUST be
included in the scope of the Options Template Record, and also
included in the associated Template Record.

```
+----------------------------+      +----------------------+
| Common Properties          |      | Specific Properties  |  Template
| Options Template Record    |      | Template Record      |  Definition
|                            |      |                      |
| scope: commonPropertiesID  |      | commonPropertiesID   |
| Common Properties          |      | Specific Properties  |
+------------+---------------+      +----------+-----------+
.............|.................................|.........................
            |                                  |
+------------v-------------+      +----------v----------+
| Common Properties        |      | Specific Properties |+  Exported
| Data Record              |------> Data Records        ||  Data
+--------------------------+      +--------------------+|  Records
                                   +--------------------+
```
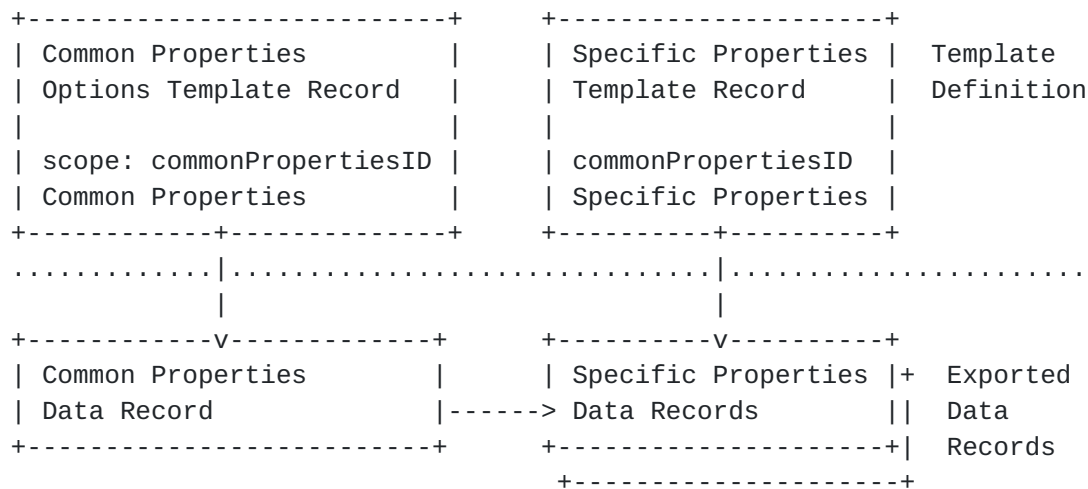
               Figure 3: Template Record and Data Record Dependencies

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 [RFC5101]. However, depending on the
transport protocol choice, this document imposes some additional
constraints. If Partial Reliable Stream Control Transmission
Protocol (PR-SCTP) [RFC3758] is selected as the IPFIX protocol, the
following PR-SCTP subsection specifications MUST be respected. If
UDP is selected as the IPFIX protocol, the following UDP subsection

specifications MUST be respected. If TCP is selected as the IPFIX
protocol, the following TCP subsection specifications MUST be
respected.

4.1. PR-SCTP

The active Common Properties MUST be sent after the SCTP association
establishment and before the corresponding Specific Properties Data
Records. In the case of SCTP association re-establishment, all
active Common Properties MUST be resent 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 resent on a regular basis.
The periodicity 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 uses 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 resending 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, and before the corresponding Specific Properties Data
Records. In the case of TCP connection re-establishment, all active
Common Properties MUST be resent 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 Withdrawal message is a Data Record defined by
an Options Template consisting of only one scope field -- namely, the
commonPropertiesID (with a type of 137 [RFC5102]) and no non-scope
fields.

```
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 = 14 octets     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Template ID N          |        Field Count = 1        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Scope Field count = 1    |0|  commonPropertiesID = 137   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Scope 1 Field Length = 8     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
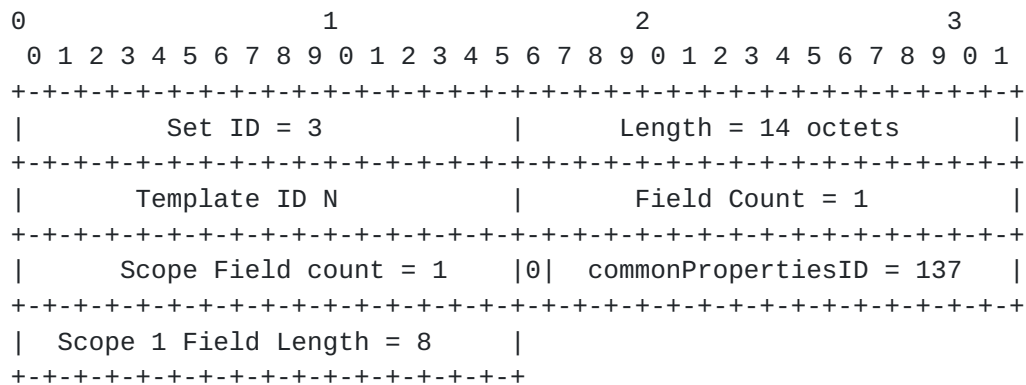
                Figure 4: Common Properties Withdrawal Message

If UDP is selected as the transport protocol, the Common Properties
Withdrawal 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.

commonPropertiesIDs are unique per SCTP association and per
Observation Domain. If the Collecting Process receives an Options
Template Record with a scope containing a commonPropertiesID that has
already been received but that 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
shut down the association.

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

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 Common Properties that it has not received before on this
SCTP association, it MUST reset the SCTP association and discard the
IPFIX Message, and it 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 this case, an alarm MUST be logged.

The Collecting Process MUST NOT decode any further Data Records that
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 previously 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 Common Properties Withdrawal
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 5 shows how this
information is repeated with classical IPFIX export in several Flow
Records.

```
+--------+--------+---------+---------+--------------------+
|srcAddrA|srcPortA|destAddrB|destPortB| <Flow1 information> |
+--------+--------+---------+---------+--------------------+
|srcAddrA|srcPortA|destAddrC|destPortC| <Flow2 information> |
+--------+--------+---------+---------+--------------------+
|srcAddrD|srcPortD|destAddrB|destPortB| <Flow3 information> |
+--------+--------+---------+---------+--------------------+
|srcAddrD|srcPortD|destAddrC|destPortC| <Flow4 information> |
+--------+--------+---------+---------+--------------------+
|   ...  |  ...   |   ...   |   ...   |       ...          |
+--------+--------+---------+---------+--------------------+
```

        Figure 5: Common and Specific Properties Exported Together

Besides A and B, 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 properties A composed of
sourceAddressA and sourcePortA, properties D composed of
sourceAddressD and sourcePortD, 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 that are
defined, the more combinations that are available.

```
+------------------+----------------+-------------+
| index for prop. A | sourceAddressA  | sourcePortA |
+------------------+----------------+-------------+
| index for prop. D | sourceAddressD  | sourcePortD |
+------------------+----------------+-------------+


+------------------+--------------------+------------------+
| index for prop. B | destinationAddressB | destinationPortB |
+------------------+--------------------+------------------+
| index for prop. C | destinationAddressC | destinationPortC |
+------------------+--------------------+------------------+


+-----------------+-----------------+---------------------+
|index for prop. A |index for prop. B |  <Flow1 information>  |
+-----------------+-----------------+---------------------+
|index for prop. A |index for prop. C |  <Flow2 information>  |
+-----------------+-----------------+---------------------+
|index for prop. D |index for prop. B |  <Flow3 information>  |
+-----------------+-----------------+---------------------+
|index for prop. D |index for prop. C |  <Flow4 information>  |
+-----------------+-----------------+---------------------+
```

        Figure 6: 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 indices is kept
to a minimum. Defining an extra index for all records would not save
bandwidth in the case of Figure 5 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 7.

```
+-------------------------- +     +-------------------+
| Common Properties       |     | Specific Properties | Template
| Options Template Record |     | Template Record     | Definition
|                         |     |                     |
| Scope: commonPropertiesID1 |     | commonPropertiesID1 |
| Scope: commonPropertiesID2 |     | commonPropertiesID2 |
| Common Properties       |     | Specific Properties |
+-----------+--------------+    +---------+----------+
............|.................................|.........................
           |                               |
+-----------v-------------+     +----------v----------+
| Common Properties       |     | Specific Properties |+  Exported
| Data Record             |------> Data Records        ||  Data
+----------------------- +     +-------------------+|  Records
                               +-------------------+
```
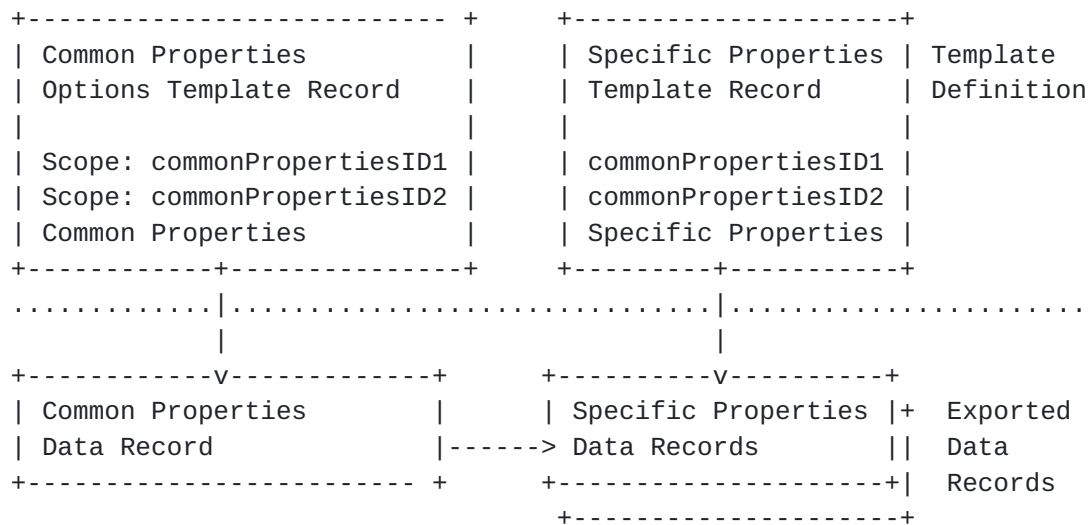
                  Figure 7: Multiple Data Reduction

7.2. Cascading Common Properties

   An Exporting Process MUST NOT export any set of Common Properties
   that 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 full reliability or TCP is recommended for the
transmission of IPFIX Messages containing Common Properties. Note
that use of UDP is less efficient for the transmission of Common
Properties, as they have to be resent 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 gain in bandwidth efficiency.

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 abundance of redundant information in Flows and would be directly
proportional to the reuse of the defined commonPropertiesID values,
with a theoretical limit where all the Data Records would use a
single commonPropertiesID. In other words, the more we reuse a
commonPropertiesID value, the better the efficiency gain. 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 the length of the commonPropertiesID is (with respect to the
total length of the Common Properties fields), the bigger the gain
is.

The example in 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. Security Considerations

The same security considerations as for the IPFIX protocol [RFC5101]
apply.

10. 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.

11. References

11.1. Normative References

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

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

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

[RFC5476]   Claise, B., Ed., "Packet Sampling (PSAMP) Protocol
              Specifications", RFC 5476, March 2009.

11.2. Informative References

   [PSAMP-MIB] Dietz, T., Ed. and B. Claise, "Definitions of Managed
                Objects for Packet Sampling", Work in Progress,
                June 2006.

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

   [RFC3758]   Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P.
                Conrad, "Stream Control Transmission Protocol (SCTP)
                Partial Reliability Extension", RFC 3758, May 2004.

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

   [RFC4960]   Stewart, R., "Stream Control Transmission Protocol",
                RFC 4960, September 2007.

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

   [RFC5472]   Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP
                Flow Information Export (IPFIX) Applicability",
                RFC 5472, March 2009.

   [RFC5474]   Duffield, N., Ed., "A Framework for Packet Selection and
                Reporting", RFC 5474, March 2009.

   [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.

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 document. Let's
   suppose we have to periodically export data about two IPv6 Flows.

   In this example, we report the following information:

```
Flow|        dstIPv6Address                 | dst- |nPkts|nBytes
    |                                        | Port |     |
    ----------------------------------------------------------------
 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
```

                   Figure 8: Flow Information Example

   The Common Properties in this case are the destination IPv6 address
   and the destination port. We first define an Options Template that
   contains the following Information Elements:

   o Scope: commonPropertiesID in [RFC5102], with a type of 137 and a
      length of 8 octets.

   o The destination IPv6 address: destinationIPv6Address in [RFC5102],
      with a type of 28 and a length of 16 octets.

   o The destination port: destinationTransportPort in [RFC5102], with
      a type of 11, and a length of 2 octets.

   Figure 9 shows the Options Template defining the Common Properties
   with commonPropertiesID as scope:

```
  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 = 24 octets    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Template ID = 257      |          Field Count = 3      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Scope Field count = 1   |0|  commonPropertiesID = 137   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Scope 1 Field Length = 8     |0|  destinationIPv6Address = 28|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Field Length = 16       |0|destinationTransportPort = 11|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Field Length = 2        |           (Padding)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Common Properties Options Template

The Specific Properties Template consists of the information not
contained in the Options Templates, i.e., Flow-specific information;
in 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 10):

o   commonPropertiesID with a length of 8 octets.

o   The number of packets of the Flow: inPacketDeltaCount in
    [RFC5102], with a length of 4 octets.

o   The number of octets of the Flow: inOctetDeltaCount in [RFC5102],
    with a length of 4 octets.

```
 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 = 20 octets     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Template ID = 258      |         Field Count = 3       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|   commonPropertiesID = 137  |        Field Length = 8       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|    inPacketDeltaCount = 2   |        Field Length = 4       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|    inOctetDeltaCount = 1    |        Field Length = 4       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
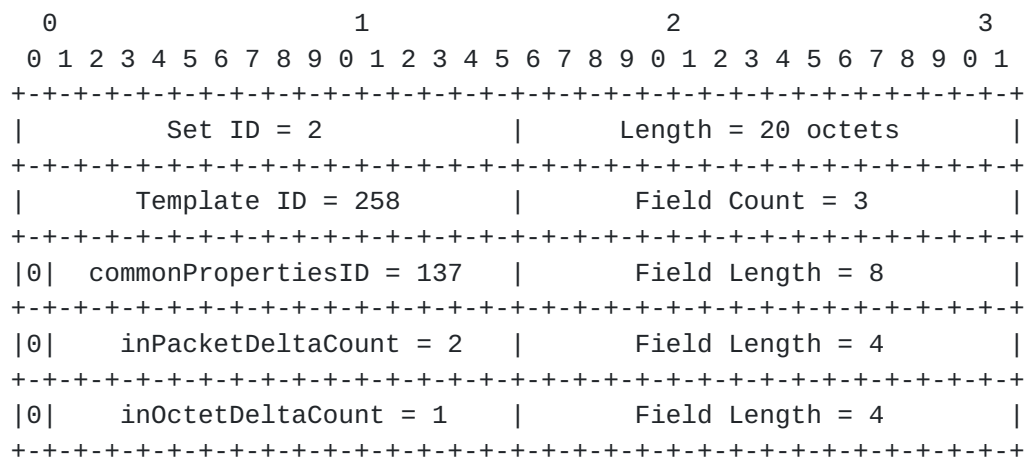
                  Figure 10: Specific Properties Template

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

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

                              Figure 11

The Data Records reporting the Common Properties will look like:

```
 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 = 257          |        Length = 60 octets     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                               101                           -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                             -+
|                                                               |
+-           2001:DB8:80AD:5800:0058:0800:2023:1D71           -+
|                                                               |
+-                                                             -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             80                |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-                              -+
|                                 102                           |
+-                               -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                 |                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                             -+
|                                                               |
+-                                                             -+
|           2001:DB8:80AD:5800:0058:00AA:00B7:AF2B              |
+-                                                             -+
|                                                               |
+-                               -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                 |             1932            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

              Figure 12: 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 13

Figure 14 shows the first Data Record listed in the table:

```
 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 = 258        |            Length = 16        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                              101                            -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             30                |             6000              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Data Record 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.
The OWD is computed by subtracting the time of observation of the
same packet at the two end-points with synchronized clocks. 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. 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 Diffserv codepoint
(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 a series of Flow Records composed
of two types of information: some common information for all packets,
and some unique information about each packet 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 measurement's end-points can
individually and independently apply the redundancy technique
described in this document in order to save some bandwidth for their
respective Flow Records exports.

The Templates required for exporting measurement data of this kind
are illustrated in the figures below. Figure 15 shows the Options
Template containing the information concerning Flows using the
commonPropertiesID as scope. In the Common Properties Template, we
export the following Information Elements:

o The source IPv4 Address: sourceIPv4Address in [RFC5102], with a
   type of 8 and a length of 4 octets.

o The destination IPv4 Address: destinationIPv4Address in [RFC5102],
   with a type of 12 and a length of 4 octets.

o The Class of Service field: ClassOfServiceIPv4 in [RFC5102], with
   a type of 5 and a length of 1 octet.

o The Protocol Identifier: protocolIdentifier in [RFC5102], with a
   type of 4 and a length of 1 octet.

o The source port: sourceTransportPort in [RFC5102], with a type of
   7 and a length of 2 octets.

o The destination port: destinationTransportPort in [RFC5102], with
   a type of 11 and a length of 2 octets.

The commonPropertiesID Information Element is used as the Scope
Field.

```
  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 = 40 octets      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Template ID = 256        |       Field Count = 7         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Scope Field count = 1    |0|  commonPropertiesID = 137   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Scope 1 Field Length = 4      |0|    sourceIPv4Address = 8    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 4         |0| destinationIPv4Address = 12 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 4         |0|  classOfServiceIPv4 = 5     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 1         |0|  protocolIdentifier = 4     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 1         |0|  transportSourcePort = 7    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 2         |0|transportDestinationPort = 11|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Field Length = 2         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Example Flow Properties Template

For passive OWD measurement, the Packet Properties Template or
Specific Properties Template consists of at least the timestamp and
packet ID. Additionally, this template contains a commonPropertiesID
field to associate the packet with a Flow.

Figure 16 displays the template with the packet properties. In this
example, we export the following Information Elements:

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

o The packet timestamp: observationTimeMilliseconds in the PSAMP
   information model [RFC5477], with a type of 323 and a length of 8
   octets.

o digestHashValue in the PSAMP information model [RFC5477], with a
   type of 326 and a length of 8 octets.

o The packet length: ipTotalLength in the IPFIX information model
   [RFC5102], with a type of 224 and a length of 8 octets.

```
  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 = 36 octets      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |       Template ID = 257      |        Field Count = 4        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0|  commonPropertiesID = 137  |        Field Length = 4       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0| observationTimeMillis.= 323 |       Field Length = 8        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0|    digestHashValue = 326    |        Field Length = 8       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0|    ipTotalLength = 224      |        Field Length = 8       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                Figure 16: 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, destinationTransportPort=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,
destinationTransportPort=2, commonPropertiesID=4). For a Flow of
1000 packets, this sums to 28018 bytes. This is a decrease of more
than 26 percent.

A.3. Common Properties Withdrawal Message

This section shows an example commonPropertiesID Withdrawal message.
Figure 17 depicts the Options Template Record with the
commonPropertiesID as unique scope field, and no non-scope fields.

```
     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 = 14 octets       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Template ID 259        |        Field Count = 1        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Scope Field count = 1    |0|  commonPropertiesID 137     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Scope 1 Field Length = 8     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

           Figure 17: Example Common Properties Withdrawal Template

Figure 18 shows the Option Data Record withdrawing commonPropertiesID
N:

```
 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 = 259         |       Length = 12 octets      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                              N                              -+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
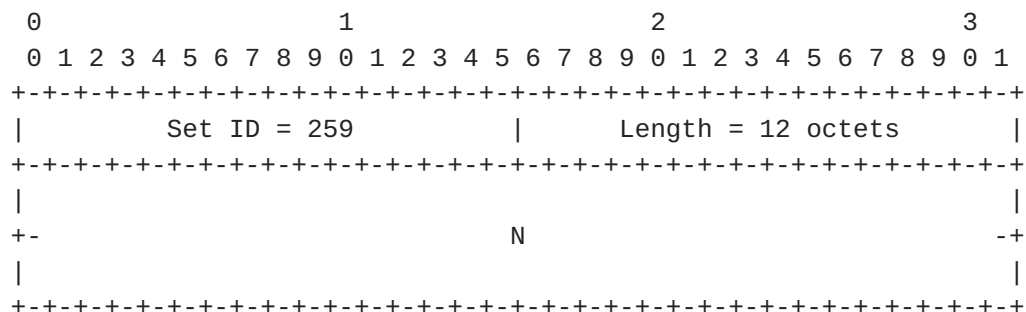
Figure 18: Record Withdrawing commonPropertiesID N

Authors' Addresses

Elisa Boschi
Hitachi Europe
c/o ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Phone: +41 44 6327057
EMail: elisa.boschi@hitachi-eu.com

Lutz Mark
Fraunhofer IFAM
Wiener Str. 12
28359 Bremen
Germany

Phone: +49 421 2246206
EMail: lutz.mark@ifam.fraunhofer.de

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

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