Network Working Group
Internet-Draft
Expires: May 22, 2008

F. Dressler
C. Sommer
Univ. Erlangen
G. Muenz
Univ. Tuebingen
A. Kobayashi
NTT PF Lab.
November 19, 2007

IPFIX Flow Aggregation
<draft-dressler-ipfix-aggregation-04.txt>

Status of this Memo

Copyright Notice

Abstract

IPFIX Flow Aggregation describes a methodology for reducing the amount of measurement data exchanged between monitoring devices (IPFIX Exporters) and analyzers (IPFIX Collectors). Aggregation techniques represent a necessary enhancement in order to cope with

increasing amounts of monitoring data that accrue with the ever-growing network capacities. Using aggregation techniques, measurement information of multiple Flows that are sharing some common criteria is merged to be exported in one Compound Flow. Subsets of Flows eligible for aggregation, as well as the desired degree of similarity, can be customized using a set of Aggregation Rules.

Table of Contents

1. Introduction

Flow measurement in high-speed large-scale networks easily produces an amount of data that can no longer be handled by a single IPFIX Collector. Also, many applications processing Flow measurement data do not require detailed Flow-level information, but require only generic Flow information, with the scope of this information being very application-specific. Examples for applications benefiting of IPFIX Flow aggregation are charging and intrusion detection. In the former application, detailed information about individual Flows is not required. Similarly, intrusion detection applications may be satisfied with Flow information for specific subnets. Flow aggregation is also a viable solution for the anonymization of Flow information.

This document presents a flexible Flow aggregation scheme that helps reduce the number and the size of exported Flow Records, as well as helps adapt the transmitted measurement information to the requirements of deployed applications. Measurement data reduction is achieved by discarding unneeded measurement information and merging multiple individual Flows into a smaller number of Compound Flows, which are then exported to the analyzer.

Flow aggregation can take place either directly in IPFIX-enabled devices or externally, in an IPFIX concentrator. Monitoring networks can thus be deployed in a logical tree topology, using multiple levels of intermediate concentrators, rather than in a logical star topology, i.e. with each exporter connecting to a central analyzer.

The following sections illustrate the design and implementation of Flow aggregation. 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 [RFC2119].

2. Terminology

Apart from the basic terms described in [RFC3917], [I-D.ietf-ipfix-protocol], and [I-D.ietf-ipfix-architecture], the following terms are used within this document:

Compound Flow:
A Compound Flow is the result of an aggregation of one or more individual input Flows that matched an Aggregation Rule. It might, for example, contain the total count of all packets addressed to a common subnet that were observed within a given time interval.

Aggregation Rule:
An Aggregation Rule defines the properties of a Compound Flow and the contents of the corresponding Flow Records. Optionally, a set of selection criteria MAY be specified in order to restrict the applicability of the Aggregation Rule to those Flows that show certain patterns.

Preceding Rule:
The Preceding Rule represents a mechanism to inform the collector about the position of the matching Aggregation Rule in the entire chain or tree of Aggregation Rules. This parameter MAY be used to dynamically transmit parts of the rule chain or tree.

3. Architecture

Aggregation of measurement data may take place at two possible stages:

An internal aggregator is implemented as part of a metering process running in an IPFIX-enabled device. The aggregated Flow data is exported in form of Compound Flows.

An IPFIX concentrator, as introduced in [RFC3917], may be used if the deployed monitoring devices cannot be modified to incorporate an internal aggregator and, hence, an external aggregator needs to be deployed. Additionally, a concentrator MAY be employed to save processing resources of distributed monitoring devices. Furthermore, concentrators enable cascaded, multi-level aggregation of Flow information.

4. Aggregation

In order to efficiently customize both the contents and the size of exported Compound Flows, a two-step approach is proposed.

1. Incoming Flows are selected by comparing contained information with configured selection criteria, enabling the aggregator to discard unwanted Flows.

2. The selected Flows are aggregated by discarding fields or parts of fields, enabling the aggregator to create Compound Flows by merging Flows according to a reduced Flow Key.

For the configuration of the selection and aggregation processes, a rule-based approach is proposed, where each aggregator is supplied a list of user-defined Aggregation Rules.

4.1. Aggregation Rule

An Aggregation Rule (see Section 5 for an example) consists of multiple instructions, one for each data field that is to be considered. Each of the Aggregation Rules' instructions comprises the following elements:

- The data field the instruction refers to (e.g. destinationIPv4Address). Only Flows that contain the field mentioned will be eligible for aggregation according to this Aggregation Rule.

- An optional selection pattern (e.g. 192.0.2.0/24) for this field that further restricts eligibility of incoming Flows to those that match the given value(s). Only Flows that match all patterns of the Aggregation Rule will be aggregated. Selection is explained in more detail in Section 4.3

- A field modifier (e.g. mask to 28 bits) that either configures how much information in this field should be retained unmodified by the aggregator or that the field's values should be aggregated instead. Compound Flow creation is explained in more detail in Section 4.4.

Fields that do not appear in any of an Aggregation Rule's instructions are not part of its associated Compound Flow Records. Incoming Flows that are not covered by any Aggregation Rule are discarded.

4.2. Multiple Aggregation Rules

By default, incoming Flow Records are checked against all of the configured Aggregation Rules. If an Aggregation Rule matches, i.e. if the Flow Record comprises all required fields and matches all given patterns, the field modifiers are applied and corresponding Compound Flow Records generated or updated. Therefore, incoming Flow Records that match multiple Aggregation Rules contribute to multiple Compound Flows.

In some cases, it is preferred that an incoming Flow Record that matched a certain Aggregation Rule is not checked against further Aggregation Rules in order to avoid that this Flow contributes to multiple Compound Flows. Therefore, it is possible to indicate a Preceding Rule for each Aggregation Rule. If a Preceding Rule is set for an Aggregation Rule, an aggregator first tries to aggregate an incoming Flow according to the Preceding Rule. Only if the Preceding Rule is not applicable, e.g. because the incoming Flow does not match the given pattern, the current Aggregation Rule is applied. Using

the Preceding Rule relationship, Aggregation Rules can thus be organized in rule chains and rule trees where only the first matching Aggregation Rule is applied in every chain or branch. Consequently, each incoming Flow Record is aggregated at most once per chain or tree. Rules that do not define a Preceding Rule constitute the beginning of a rule chain or the root of a rule tree and are used to check all incoming Flow Records.

If a Preceding Rule is set for an Aggregation Rule, a suitable mechanism SHOULD be employed by an aggregator to communicate to receivers of a Compound Flow not only its common inclusion criteria, i.e. having matched the Aggregation Rule's selection patterns, but also its common exclusion criteria, i.e. not having matched any Preceding Rule's selection patterns. IPFIX extensions that enable efficient transport of such information are introduced in [I-D.sommer-ipfix-mediator-ext].

4.3. Selection

As introduced in Section 4.1, the applicability of an Aggregation Rule MAY be restricted to Flows that match certain patterns. Thus, patterns act as criteria that enable the selection of Flows eligible for aggregation and subsequent export to the analyzer. For example, the pattern "80" can configured for the sourceTransportPort field in order to export only Compound Flows sent by an HTTP server.

Selecting Flows means that all of the source Flows that make up a certain Compound Flow will share a specific set of field values (e.g. destination address 192.0.2.1 and destination port 80). This common set of field values MUST be transmitted to receivers along with Compound Flows' specific field values, so as not to lose information.

In order to conserve traffic volume it SHOULD, however, not be directly included in all exported Compound Flow Records, but rather communicated by more bandwidth-conserving means which still guarantee a stable association from specific properties to Common Properties of a Flow.

One such means is the transmission as a set of Common Properties. This method is outlined in [I-D.ietf-ipfix-reducing-redundancy] and illustrated in Figure 1. In order to unambiguously communicate to receivers which Common Properties of a Flow stem from aggregation, when multiple Common Properties are transmitted in one Flow, an aggregator SHOULD make sure that the first commonPropertiesID transmitted in Flows directly corresponds to the set of selection criteria used.

Extensions to the IPFIX protocol and the IPFIX information model,

which allow for a much more compact data format, are introduced in [I-D.sommer-ipfix-mediator-ext].

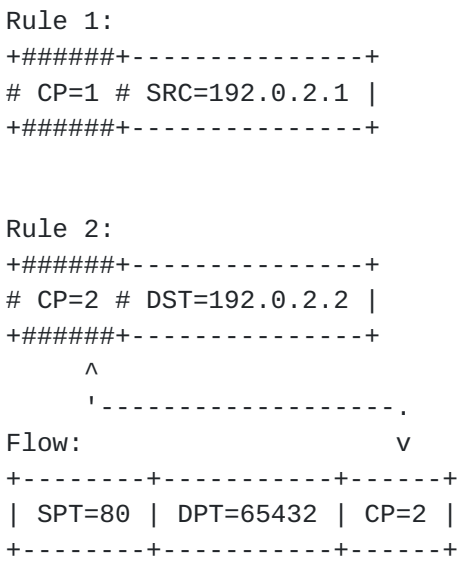

Figure 1: Using Common Properties to transmit Rules

4.4. Compound Flow Creation

As introduced in Section 4.1, a different field modifier can be assigned to each field of an Aggregation Rule. The following types of field modifiers can be used:

discard:
The field is not included in Compound Flow Records, i.e. Compound Flows are not distinguishable with respect to this field. Incoming Flows with different values for this field can be merged and thus contribute to the same Compound Flow.

keep:
The field is included unmodified in Compound Flow Records, i.e. Compound Flows are distinguishable with respect to this field. Incoming Flows with different values for this field are not merged, but contribute to different Compound Flows instead.

mask to n bits:
The field is included in Compound Flow Records, but the number of significant bits is reduced (applicable to IP addresses only). Incoming Flows with IP addresses belonging to the same subnet are merged, so Compound Flows are distinguishable with respect to resulting subnet addresses only. In accordance with [I-D.ietf-ipfix-info], the resulting subnet address MAY be encoded

with an IP prefix field and an IP mask field. For performance reasons it SHOULD, however, be encoded with a single field of the abstract data type ipv4Network introduced in [I-D.sommer-ipfix-mediator-ext].

aggregate:
The field is included in Compound Flow Records, but field values derived from multiple incoming Flows' values.

The value of fields with a field modifier of "aggregate" is computed from the corresponding values of the original Flows by taking the most appropriate of the following actions (listed in ascending order of priority):

1. As also specified in [I-D.ietf-ipfix-info] the value of variable fields, which may change from packet to packet within a single Flow, is determined by the first packet observed for the corresponding Compound Flow. As a consequence, if no other action is more appropriate, the default behavior requires using the value of the incoming Flow Record with the earliest start timestamp.

2. For some Information Elements, [I-D.ietf-ipfix-info] explicitly specifies a different semantic, which is to then take precedence over the aforementioned default behavior.

3. For some Information Elements, Table 1 specifies an aggregation function that has to be used in order to obtain a correct, aggregated value. If such an aggregation function is listed, it takes precedence over all other available functions. For example, the start timestamp of the Compound Flow is always set to the minimum of the original start timestamps, while packet and octet counts of aggregated Flows are always summed up.

Information Element	Aggregation Function
minimumPacketLength	min
minimumTtl	min
flowStartSeconds	min
flowStartMilliSeconds	min
maximumPacketLength	max
maximumTtl	max
flowEndSeconds	max
flowEndMilliSeconds	max
octetDeltaCount	sum
packetDeltaCount	sum

Table 1: Treatment of Fields Carrying Metering Information

4.5. Deriving Templates from Aggregation Rules

With the definition of an Aggregation Rule as being comprised of one instruction per Compound Flow field, each Aggregation Rule unambiguously defines the structure of these Compound Flows. As illustrated in Table 2, all selection patterns of an Aggregation Rule become Common Properties of associated Compound Flows. With the exception of fields that are discarded, all fields of an Aggregation Rule transmit specific properties of Compound Flows and will thus need to be included in each exported Flow Record. Of those fields, all fields, except those that were aggregated, form the Flow Key of Compound Flows, as defined in [I-D.ietf-ipfix-architecture].

Selection	Aggregation	Common Property	Specific Property	Flow Key
any	discard			
any	keep		x	x
any	mask		x	x
any	aggregate		x	
pattern	discard	x		
pattern	keep	x	x	x
pattern	mask	x	x	x
pattern	aggregate	x	x	

Table 2: Mapping Rules to Templates

It should be noted that certain combinations of selection and aggregation instructions can cause undesirable side effects and

SHOULD NOT be used. These combinations are:

atomic pattern, do not discard:
Selecting Flows to only allow a specific field value (as opposed to a range of values) and retaining this field as a discriminating property will lead to the transmission of the selection pattern as both a Common Property of all Compound Flows and a discriminating property. If only an atomic pattern is used, the field can be discarded with no loss of information.

any field value, discard:
If neither a pattern nor a field modifier should apply to a field, it is sufficient for an Aggregation Rule to not include an instruction for this field. Specifying for a field neither a pattern nor a modifier will mandate presence of the field for an incoming Flow to be eligible for aggregation, but not accomplish any real selection or aggregation.

pattern, aggregate:
Selecting Flows to only allow certain field values in non-discriminating properties, such as packet counters, then modifying these properties, can lead to semantic conflicts when interpreting the received Compound Flows.

5. Example

An Aggregation Rule shown in Table 3 will set up a stream of Compound Flows, creation of which is performed as follows.

Selection:
Only Flows containing at least fields for the source address, destination address, destination port, and the packet count will be considered for aggregation. In addition, the destination address must be in the subnet 192.0.2.0/28 and the destination port must be equal to 80.

Compound Flow creation:
Destination addresses are merged according to subnets in 192.0.2.0/30 and all packet counters of one Compound Flow are added up.

Export:
The resulting Compound Flow Records comprise the source address, the destination subnet address, and the packet counter as their specific properties, as well as a destination subnet of 192.0.2.0/28 and a destination port of 80 as their Common Property.

IPFIX Field	Selection	Aggregation
sourceIPv4Address		keep
destinationIPv4Address	192.0.2.0/28	mask to 30 bit
destinationTransportPort	80	discard
packetDeltaCount		aggregate

Table 3: Example Aggregation Rule

Adding the Aggregation Rule shown in Table 4 and configuring it with a Preceding Rule, the Aggregation Rule of Table 3, will set up a second stream of Compound Flows, creation of which is performed as follows.

Selection:

As introduced in Section 4.2, Flows that were already aggregated according to the Preceding Rule will be skipped. In addition, only Flows containing at least fields for the source address, destination address, destination port, and the packet count will be considered for aggregation. Furthermore, the destination port of incoming Flows must be equal to 80.

Compound Flow creation:

Source and destination addresses are merged according to subnets in 192.0.2.0/30 and all packet counters of one Compound Flow are added up.

Export:

The resulting Compound Flow Records comprise the source subnet address, the destination subnet address, and the packet counter as their specific properties, as well as a destination port of 80 as their Common Property. Furthermore, a Common Property of all Compound Flow Records is that they did not match the Preceding Rule, i.e. the combination of their destination subnet and destination port was not 192.0.2.0/28 and 80.

IPFIX Field	Selection	Aggregation
sourceIPv4Address		mask to 30 bit
destinationIPv4Address		mask to 30 bit
destinationTransportPort	80	discard
packetDeltaCount		aggregate

Table 4: Second Aggregation Rule (Chained)

The following example Table 5 illustrates the application of the described chain of Aggregation Rules to selected Flows.

Source IP	Source Port	Destination IP	Destination Port	Packets
192.0.2.1	64235	192.0.2.101	80	10
192.0.2.2	64236	192.0.2.102	110	10
192.0.2.3	64237	192.0.2.103	80	10
192.0.2.101	64238	192.0.2.1	80	10
192.0.2.102	64239	192.0.2.2	80	10

Table 5: Incoming Flows

Two sets of Compound Flows, as depicted in Table 6 and Table 7, will be exported in our example according to the Aggregation Rules shown in Table 3 and Table 4, respectively.

Source IP	Destination IP	Destination Port	Packets
192.0.2.101	192.0.2.0	80	10
192.0.2.102	192.0.2.0	80	10

Table 6: Compound Flows according to the first Aggregation Rule

Source IP	Destination IP	Destination Port	Packets
192.0.2.0	192.0.2.100	80	20

Table 7: Compound Flows according to the second Aggregation Rule

6. Open Issues

While the aggregation methodology introduced in Section 4 solves most problems that could arise when doing Flow aggregation, some issues are left unresolved. These issues require special attention or need to be addressed at higher layers and are presented in this section.

One problem arises when received Option Data Records are forwarded by an aggregator. Option Data Records that refer to an Observation Domain, e.g. Data Records based on the Metering Process

(Reliability) Statistics Option Template, only include an observationDomainId. However, [I-D.ietf-ipfix-info] only mandates that the observationDomainId be locally unique to an Exporting Process, so in order to unambiguously refer to an Observation Domain, an additional identifier of the Exporting Process would need to be transmitted. While the selection of an Observation Domain ID that is unique to the aggregation domain would alleviate this issue, a more generic solution seems to be preferable.

Another problem arises when different sources transmit Flows containing merely pseudonyms instead of IP addresses, and these Flows are to be aggregated e.g. by an IPFIX concentrator. If an aggregator is not explicitly informed of the anonymous nature of received Flows, it assumes that identical IP address values refer to identical hosts, which might not be the case if Flow sources employ different algorithms to generate pseudonyms.

7. Security Considerations

As all methods described in this document are merely variations on methods already introduced in [I-D.ietf-ipfix-protocol], the same security considerations regarding exchange of Flow information apply.

8. IANA Considerations

This document has no actions for IANA.

9. References

9.1. Normative References

[I-D.ietf-ipfix-protocol]
Claise, B., "Specification of the IPFIX Protocol for the Exchange of IP Traffic Flow Information", draft-ietf-ipfix-protocol-26 (work in progress), September 2007.

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

9.2. Informative References

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

[I-D.ietf-ipfix-reducing-redundancy]
Boschi, E., "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", draft-ietf-ipfix-reducing-redundancy-04 (work in progress), May 2007.

[I-D.sommer-ipfix-mediator-ext]
Sommer, C., Dressler, F., and G. Muenz, "Mediator-Specific Extensions to IPFIX Protocol and Information Model", draft-sommer-ipfix-mediator-ext-00 (work in progress), November 2007.

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

Authors' Addresses

Falko Dressler
University of Erlangen-Nuremberg
Department of Computer Science 7
Martensstr. 3
Erlangen 91058
Germany

Phone: +49 9131 85-27914
Email: dressler@informatik.uni-erlangen.de
URI: http://www7.informatik.uni-erlangen.de/

Christoph Sommer
University of Erlangen-Nuremberg
Department of Computer Science 7
Martensstr. 3
Erlangen 91058
Germany

Phone: +49 9131 85-27993
Email: christoph.sommer@informatik.uni-erlangen.de
URI: http://www7.informatik.uni-erlangen.de/~sommer/

Gerhard Muenz
University of Tuebingen
Computer Networks and Internet
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70534
Email: muenz@informatik.uni-tuebingen.de
URI: http://net.informatik.uni-tuebingen.de/

Atsushi Kobayashi
NTT Information Sharing Platform Laboratories
3-9-11 Midori-cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81-422-59-3978
Email: akoba@nttv6.net

Full Copyright Statement

Intellectual Property

Acknowledgment

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