IPFIX Working Group
Internet-Draft
Intended status: Informational
Expires: August 25, 2008

A. Kobayashi
K. Ishibashi
T. Kondoh
NTT PF Lab.
D. Matsubara
Hitachi

February 22, 2008

Reference Model for IPFIX Mediators
draft-kobayashi-ipfix-mediator-model-02.txt

Status of this Memo

Copyright Notice

Abstract

An IPFIX Mediator is an intermediate device between IPFIX Exporting Processes and IPFIX Collecting Processes. IPFIX Mediators act as an IPFIX Proxy, and IPFIX Concentrator. IPFIX Mediators mediate IPFIX protocol using several functions. That enables the flow-based measurement system to become a high-capacity system and accommodate a variety of monitoring methods. This document describes each function that is provided by IPFIX Mediators and the method of handling the Flow Records of each function. In addition, this document describes a model of an applicable scenario using IPFIX Mediators.

Table of Contents

1. Introduction

An IPFIX Mediator is located between one or more Exporting Processes and one or more Collecting Processes. An IPFIX Mediator acts as a Collector by receiving Flow Records, and it acts as an Exporter by sending Flow Records. This dual-role architecture enables cascading IPFIX Mediators and building a combination of several solutions.

By defining IPFIX Mediators, network operators can take increasing advantage of an extensive Template format, and handle Flow Records in accordance with their preference. This document describes a model of applicable scenarios by using IPFIX Mediator and its key component.

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

The definitions of basic IPFIX and PSAMP terms are identical with those in [I-D.ietf-psamp-framework], [RFC3917], [RFC5101], [RFC5102], and [I-D.ietf-ipfix-architecture]. The terminology related to IPFIX Mediation is described in [I-D.kobayashi-ipfix-large-ps]. Other than the above terminology, the following terminology is used in this document. Therefore, terms defined in the IPFIX terminology are capitalized in this document.

Metering Process

The Metering Process in IPFIX Mediators can be considered as a partial Metering Process separated from the Metering Process in the Original Exporter. The Metering Process in IPFIX Mediators consists of a set of subprocesses that include the Selecting Process, Aggregating Process, and Modifying Process. The Metering Process generates the final Flow Records that should be exported.

Selecting Process

The Selecting Process in an IPFIX Mediator is similar to that of PSAMP Devices, which is described in [I-D.ietf-psamp-framework]. However, the Selecting Process in an IPFIX Mediator only has field-match filtering functions. This filtering function blocks Flow Records based on the values of specified Information Elements to forward them to the next process. The Selecting Process is one of the subprocesses in the Metering Process.

Aggregating Process

The Aggregating Process creates aggregated Flow Records from input Flow Records in accordance with aggregation rules that are described in [I-D.dressler-ipfix-aggregation]. The Aggregating Process is one of the subprocesses in the Metering Process.

Modifying Process

The Modifying Process carries out two different kinds of modification, as follows.

* The Modifying Process changes the Template and record structure by adding/deleting specific Information Elements. For example, the Modifying Process adds Information Elements like derived packet properties that cannot be extracted in the Original Exporters. Information Elements related to derived packet properties are described in [RFC5102].

* The Modifying Process changes the value of the specific Information Elements. For example, the values of specific Information Elements are anonymized to avoid violating privacy.

The Modifying Process is a key part of a IPFIX Masquerading Proxy. The Modifying Process is one of the subprocesses in the Metering Process.

Storing Process

The Storing Process stores the input Flow Records from any process in a storage system such as a database or flat-file system. Stored data may be retrieved from the Storing Process in different ways, which are outside the scope of this document.

Distributing Function

The Distributing Function classifies input Flow Records based on the value of specified Information Elements. The classified Flow Records are exported to the specified Collectors. The Distributing Function is carried out by the Exporting Process.

Observation Domain ID

An IPFIX Mediator does not host the Observation Points and Observation Domain. The Observation Domain ID in the IPFIX header sent by IPFIX Mediator also indicates the largest set of Observation Points in the Original Exporter, but this value does not indicate the physical entity of the Original Exporter. If an IPFIX Mediator handles one Collecting session and one Exporting session, the IPFIX Mediator does not need to change the value of the Observation Domain ID. If an IPFIX Mediator handles multiple sessions on the collecting and exporting side, the IPFIX Mediator needs to assign a new value.

Transport Session Information

In SCTP, the Transport Session Information is the SCTP association. In TCP and UDP, the Transport Session Information corresponds to a 5-tuple {exporter IP address, collector IP address, Exporter transport port, Collector transport port, transport protocol}. In IPFIX Mediator, the Collecting Process manages this information.

3. Framework for IPFIX Mediators

The internal model of an IPFIX Mediator is composed of Collecting Processes, Metering Processes, Storing Processes, and Exporting Processes. An IPFIX Mediator may optionally have Metering Processes and Storing Processes. Basically, the allocation of these processes depends on the role of devices, such as IPFIX Proxy, IPFIX Masquerading Proxy, IPFIX Distributor, and IPFIX Concentrator.

3.1. Internal Components Model

The following figure indicates four components (Collecting Process, Metering Process, Exporting Process, and Storing Process) within the IPFIX Mediator that are referred to in [RFC3917]. The Metering Process can have one or multiple subprocesses. These subprocesses are the Selecting Process, Aggregating Process, and Modifying Process that can be connected to each other in any sequence defined by the user. The Metering Process and Storing Process are options.

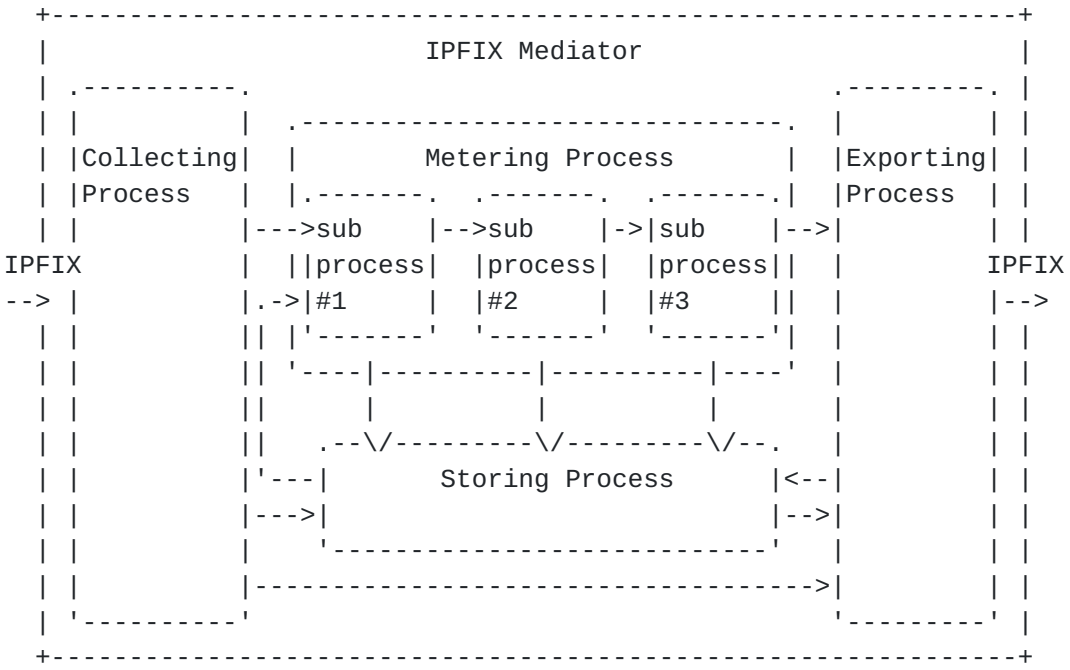

Figure A: Key components within IPFIX Mediator.

Each process is associated with a common identifier in the IPFIX Mediator. This method is similar to PSAMP associations in [I-D.ietf-psamp-sample-tech].

3.1.1. Collecting Process

The Collecting Process receives Flow Records from the previous Exporter. An instance of the process is created according to the IPFIX session. Functions of the process are described in [RFC5101]. The process also forwards received Flow Records with IPFIX header information and Transport Session Information to multiple Metering Processes or Storing Processes. In other words, Flow Records can be duplicated by forwarding multiple Metering Processes or Exporting Processes. In addition, the process can directly forward Flow Records to the Exporting Process.

3.1.2. Metering Process

The Metering Process generates a new set of Flow Records from input Flow Records with received IPFIX header information, such as "Export Time" and "Observation Domain ID". The process hosts several subprocesses. The processing order of these functions, which could be configured by user definitions would create a different set of Flow Records.

3.1.2.1. Selecting Process

The Selecting Process decides whether a Flow Record passes through to the next process. The process has a filtering function and selects Flow Records that are matched under given conditions. Prior to receiving Flow Records, the user configures a filter pattern in Selecting Process, which specifies how the Flow Records are treated by the process. If the values of some Information Elements in the Flow Record match the filter pattern, this process selects Flow Records with all fields and forwards these Flow Records to the next process. For example, the process selects Flow Records that are included in the specified destination IP address.

3.1.2.2. Aggregating Process

The Aggregating Process gathers Flow Records within a given time interval and then distinguishes Flow Records that have common properties. If values of a given key field are the same, that means those Flow Records have common properties. This process merges Flow Records that have a common property and creates an aggregated Flow Record. Therefore, for example, aggregated Flow Records have an aggregation counter that indicates the number of packets. These functions are defined in accordance with the IPFIX aggregation rule in [I-D.dressler-ipfix-aggregation].

The process has aggregation rule defined by the user prior to receiving Flow Records. The process indicates Information Elements

that should become aggregated Flow Keys and other Information Elements that should be kept or discarded. In addition, these instruction rules include Information Elements that should be added to aggregated Flow Records. Aggregated Flow Records may need to complement information that is discarded during the Aggregating Process. They help the Collector to analyze aggregated Flow Records. For example, these Information Elements correspond to "averageActiveTime", "synCount", and "flowCount" elements, as follows.

- averageActiveTime

  This Information Element indicates the average active time of an input Flow Record in the aggregated Flow Records. This Information Element is created from flow time stamp Information Elements. There are "flowStartSeconds", "flowEndSeconds", "flowStartMilliSeconds", "flowEndMilliSeconds", "flowStartSysUpTime", and "flowEndSysUpTime". Moreover, "minimumActiveTime" and "maxmumActiveTime" might be considered in addition to the element.

- synCount

  This Information Element indicates the number of input Flow Records that have "tcpControlBits", which the SYN bit sets to 1 in an aggregated Flow Record. Using this element, Collector can determine the number of SYN packets throughout the network. Moreover, "ackCount", "finCount", "pshCount", "urgCount", and "rstCount" might be considered in addition to this element.

- flowCount

  This Information Element is the number of input Flow Records included in an aggregated Flow Record.

3.1.2.3. Modifying Process

The Modifying Process modifies the input Flow Records. The process can add new Information Elements, delete included Information Elements, or modify the value of included Information Elements, as follows. If this process modifies the original Template, it SHOULD revise the received "flowKeyIndicator".

Adding new Information Elements

This function adds specified Information Elements into the input Flow Records. The values of Information Elements are extracted by searching some database based on the input value of other

specified Information Elements. The added Information Elements and used Information Elements are configured according to instructions by the user to obtain the value. The method to obtain a value from some Information Elements is outside the scope of this document.

This function, instead of the Original Exporter, adds a derived packet property parameter, which is useful for the traffic-monitoring technique. Doing that can compensate for the inability of some Exporters to add a derived packet property parameter. Therefore, the Collector does not need to recognize the difference between implementations of routers from several vendors. For example, the addition of "bgpNextHop{IPv4|IPv6}Address" and "bgpCommunity" Information Elements is useful for making a traffic matrix that covers the whole network domain. "bgpNextHop{IPv4|IPv6}Address" can indicate the egress router of some network domain. In addition, "bgpCommunity" can indicate the same group of destination or source IP addresses. This value can be given by looking for the BGP route database based on the destination or source IP address. In addition, "mplsVpnRouteDistinguisher", which cannot be extracted from the core router in MPLS networks, indicates the customer's identification. Network Operators can monitor the traffic behavior of each customer by adding "mplsVpnRouteDistinguisher" to the Flow Records. This value can be given by looking for the BGP route database based on the "mplsTopLabelStackSection" and "mplsTopLabel{IPv4|IPv6}Address".

Deleting Information Elements

This function deletes specified Information Elements according to instructions that are configured by the user, which indicate whether an Information Element should be removed. Hiding network topology information and private information by using this function is possible.

In the case of exchanging Flow Records with different network domains or customers, this function can avoid making a vulnerability by deleting unnecessary Information Elements. By deleting unnecessary Information Elements, this function can hide the network topology and another customer's information. In particular, "ipNextHopIP{v4|v6}Address", "bgpNextHopIP{v4|v6}Address", and "bgp{Next|Prev}AdjacentAsNumber" correspond to network topology information. In addition, MPLS-related Information Elements, such as "mplsLabelStackSection", that are useless for customers might be removed in the case of feeding Flow Records to VPN customers.

Modifying the value of Information Elements

This function modifies the value of specified Information Elements according to instructions configured by the user.

For example, this function enables IPFIX Mediator to overwrite private information with zeros or the maximum value. In particular, IP address and port number is sensitive private information. In the case of monitoring traffic trends and traffic engineering, these Information Elements are not essential factors for those purposes. In that case, this function anonymizes the relevant Information Elements to prevent a violation of privacy. If modification can anonymize some Information Elements, the function might need to report which Information Elements are anonymized. For example, "anonymizationIndicator" indicates which Information Elements have been anonymized as a bitmap, just like the "flowKeyIndicator". The anonymization method is outside the scope of this document.

3.1.3. Exporting Process

The Exporting Process forwards Flow Records to the next Collector. The process manages the reporting Template and makes an IPFIX datagram.

In addition, the process carries out the Distributing Function as an option. If this function is enabled, the process classifies Flow Records based on the value of specified Information Elements and then exports each classified Flow Record to the individual Collector.

For example, the Exporting Process classifies Flow Records on the basis of the peering AS, as shown in the following figure. The set of classified Flow Records is exported to a dedicated Collector on the basis of the Peering AS.

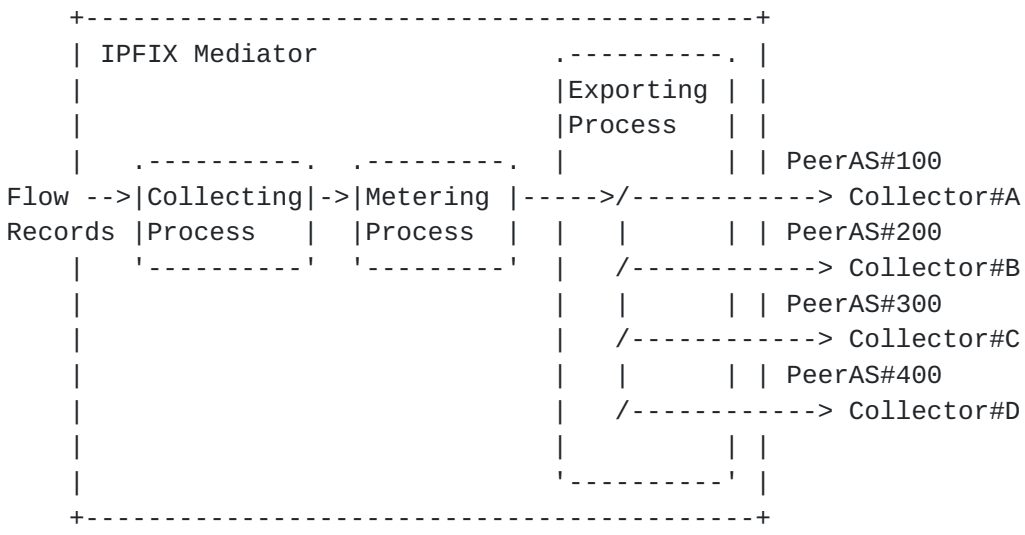

Figure B: Exporting each classified Flow Record to dedicated Collector.

3.1.4. Storing Process

The Storing Process stores the input Flow Records from any Processes in a storage system, such as a database or flat-file system. If the storing record structure that user wants is different from the template of input Flow Records, this process selects specified Information Elements from the input Flow Records using the instruction rules. Instruction rules configured by user indicate whether the Information Elements should be stored by using a field modifier. The field modifier that indicates "keep" or "discard" is applied to Information Elements within Flow Record and IPFIX header, such as "Observation Domain ID" and "Export time". This header information MAY be used when IPFIX datagrams are made of past Flow Records. That procedure is similar to the instruction of the Aggregating Process.

When another device retrieves past Flow Records on the basis of a time period given by a user, the retrieving method can be considered in many ways. One solution is that another device gets a specified flat-file from a Mediator and decodes that flat-file by itself. Other solutions are that another device sends out a query command to the IPFIX Mediator through XML-RPC, SNMP, or NETCONF, and then, the IPFIX Mediator exports the specified past Flow Records. This method is outside the scope of this document.

3.2. IPFIX Protocol Considerations

This section describes IPFIX Protocol considerations with regard to IPFIX Mediator.

3.2.1. Export Time Issue

If the Exporting Process writes the "Export Time" of the IPFIX message when an IPFIX message leaves, an IPFIX Mediator needs to compensate for any delta time, which is the difference from "Export Time" of Information Elements contained in each Flow Record by performing calculations. An IPFIX Proxy MUST reuse the "Export Time" of received IPFIX messages from the Original Exporter.

3.2.2. Observation Domain ID Management

The Observation Domain ID is locally unique to the Exporting Process in IPFIX Mediator. To comply with the IPFIX Protocol, the Observation Domain ID value is RECOMMENDED to be assigned a unique value per IPFIX Mediator. If an IPFIX Mediator relays an IPFIX datagram from a transport session to a transport session, IPFIX Mediator does not need to overwrite the Observation Domain ID with another value. If an IPFIX Mediator relays an IPFIX datagram from multiple transport sessions to a single transport session, IPFIX Mediator needs to overwrite the Observation Domain ID. In that case, IPFIX Mediator assigns the Observation Domain ID based on received Transport Session Information and the original Observation Domain ID. The renewed Observation Domain ID SHOULD be managed using the received Transport Session Information and original Observation Domain ID. This linkage information is available for overwriting the scope field of the Option Template.

Note
If the Metering Process aggregates input Flow Records, the value of the Observation Domain ID should be 0 to comply with the description in [RFC5101].

3.2.3. Template Management

The Template ID of a generated Template SHOULD be unique on the basis of the Observation Domain ID assigned by an IPFIX Mediator. The Template ID needs to be unique on the basis of IPFIX Mediator when the Observation Domain ID is 0. If the IPFIX Mediator overwrites the received Template ID to relay a received Template or modified Template, the renewed Template ID SHOULD be managed using received Transport Session Information and the received Observation Domain ID. This linkage information is available for overwriting the scope field of an Option Template and Template handling. If IPFIX Mediator

receives a "Template Withdraw Message", it SHOULD modify this message to indicate relevant Templates, and send a "Template Withdraw Message".

3.2.4. Transport Session Management

Each session of the Collecting Process and Exporting Process should operate independently. Even if one session is reset, the status of the other session is kept current. However, Templates for resetting the Collecting Session SHOULD be withdrawn for the Exporting Session.

3.2.5. Option Template Management

IPFIX Mediator MUST check whether the scope field is applicable, if Data Records associated with Option Templates are exported. If an IPFIX Mediator rewrites the Observation Domain ID or Template ID, these values included in scope fields SHOULD be rewritten before exporting. Instead of exporting Option Template Records and associated Data Records, Information Elements, such as sampling rate or sampling method, exported using the Option template Record from the Original Exporter could be merged in a Flow Record in an IPFIX Mediator. In that case, IPFIX Mediator MUST modify the relevant Template Record. Several sorts of received Statistics Option Template Records and associated Data Records could be exported in different ways as other Templates. In IPFIX Mediator, the Data Record associated by Statistics Option Template Records can be exported after merging its counter. In addition, Statistics Option Template Records and associated Data Records can be exported by indicating the source of the statistics data as a scope field instead of merging the counter. This method is described in Section 5. The user policy determines whether IPFIX Mediator and the above methods should export Option Templates Records and associated Data Records.

3.2.6. Reporting of Exporter Information

If IPFIX Mediator acts as an IPFIX Masquerading Proxy or IPFIX Proxy, reporting the Original Exporter IP address increases the vulnerability. On the other hand, if IPFIX Mediator acts as other devices, such as IPFIX Concentrator or IPFIX Distributor, the Exporter IP address is important information for traffic analysis such as traffic engineering. In the case of making a traffic matrix, the Exporter IP address can indicate the ingress router of a network domain. Therefore, reporting of Exporter Information, such as Exporter IP address, is useful to identify the Original Exporter. There are various methods as follows.

An IPFIX Mediator can directly merge Exporter Information into Flow Records or use Option Templates described in Section 5. If an IPFIX

Mediator receives Information Elements related to the Exporter information, IPFIX Mediator SHOULD NOT rewrite its own previous Exporter information. The IPFIX Mediator can append its own previous Exporter Information instead of rewriting. In the Collecting Process, the order of the Exporter information indicates the Original Exporter and the route of IPFIX Mediator. These methods defined by user policy determine whether IPFIX Mediator should report Exporter Information.

4. Solution Scenarios with IPFIX Mediators

4.1. Flexible Aggregation

An IPFIX Mediator can aggregate Flow Records in the same manner as that of IPFIX Concentrator and reduce the number of Flow Records received by a Collector.

The following figure indicates a cascade connection of IPFIX Mediators. If a Collector measures a traffic matrix to obtain traffic demand, the Collector needs Flow Records of the whole network domain, but does not need detailed Flow Records. In the first step, a first level Mediator receives Flow Records from IPFIX Devices and then creates aggregated low-level Flow Records. For example, this step is prefix mask aggregation. Next, a second level Mediator receives aggregated Flow Records and aggregates them further. For example, the second step is the aggregation of the BGP next-hop address and Exporter address. After this, the Collector receives high-level aggregated Flow Records and then stores them. This method enables step-by-step aggregation of Flow Records without overloading a single node.

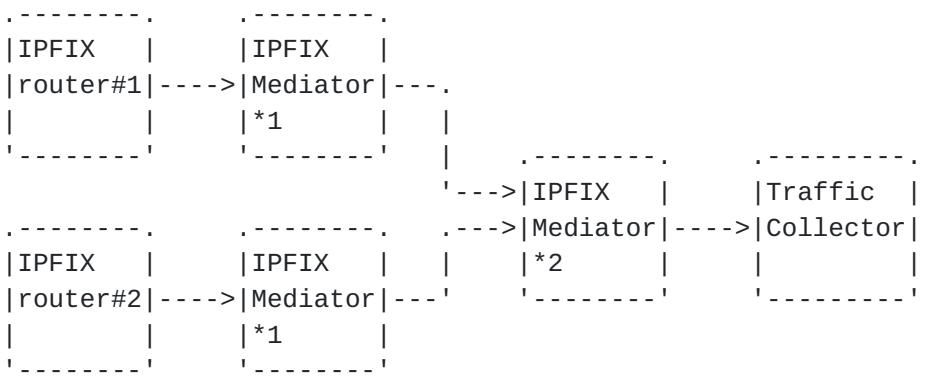

Figure C: Flexible Aggregation with cascading IPFIX Mediators.

4.2. Distributed Aggregation

When the network is used globally, the distances between PoPs become longer, and the maintenance of a dedicated management network is very expensive. Therefore, the huge number of Flow Records has burdened management networks of global ISPs. If network operators place Mediators at each PoP, the number of Flow Records exported from each PoP can be reduced. Mediators can minimize the number of Flow Records exported to the Collector. If the Collector needs detailed information, it can retrieve Flow Records from Mediators that store original Flow Records.

A management network of a global ISP is shown in the following figure. The Mediators are located at each PoP of the network, and they collect Flow Records from routers in each PoP domain. The Mediator reduces the number of Flow Records by aggregating or filtering, so this system reduces the load of a management network.

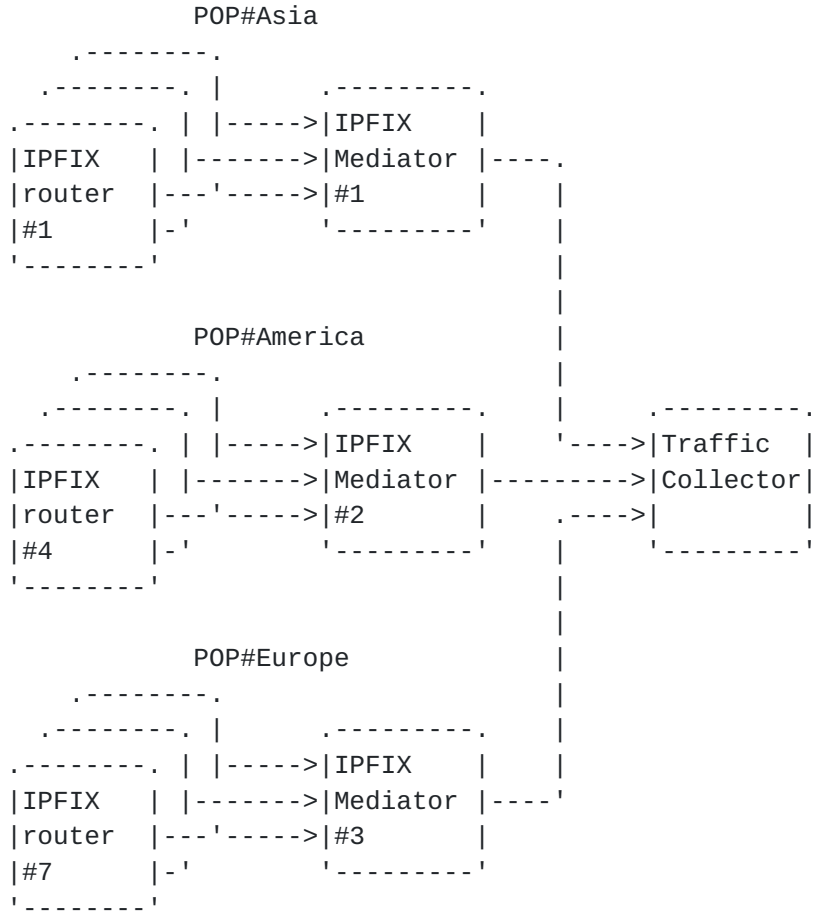

Figure D: Traffic monitoring architecture in global network.

4.3. Duplication of Flow Records

An IPFIX Mediator duplicates Flow Records to achieve redundant storage or utilizes them for several purposes. The pair of Collecting Process and Metering Process is similar to the pair of the Observation Point and Metering Process. The Collecting Process duplicates Flow Records by forwarding them to the multi-Metering Process.

Several departments in an ISP want to use the same traffic information for each intended purpose. The network design department measures the traffic matrix to obtain traffic demand. The customer service division uses traffic information for performing accounting services for each customer while network operators use traffic information for trouble shooting analysis. That situation is shown in the following figure. An IPFIX Mediator distributes Flow Records to several Collectors that have the appropriate aggregated granularity. In addition, when network operators conduct troubleshooting, past Flow Records from Mediators can be retrieved.

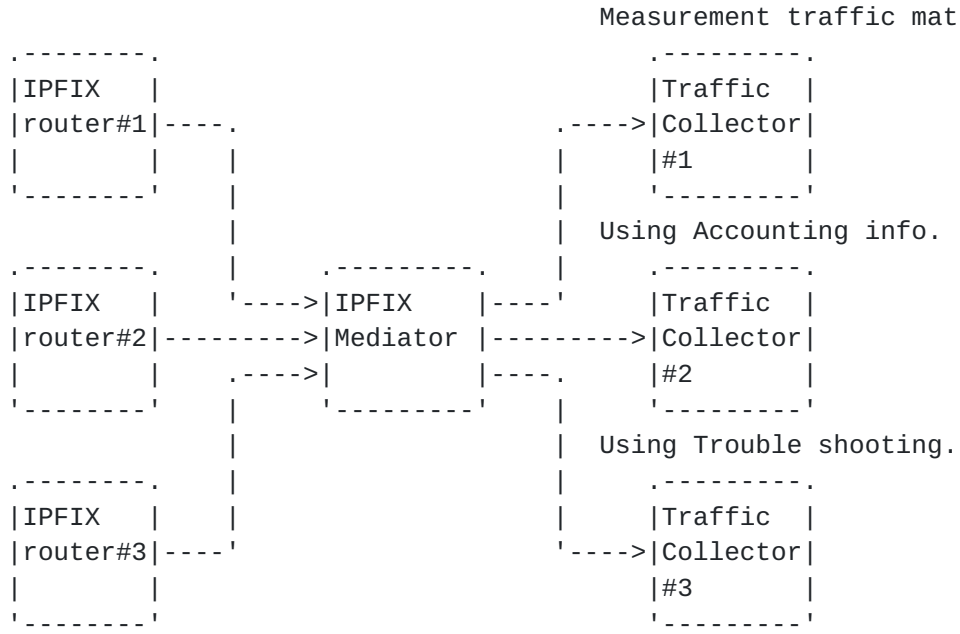

Figure E: Duplication of Flow Records for several purposes.

4.4. Distribution of Flow Records

An IPFIX Mediator MAY distribute Flow Records based on the value of specified Information Elements. This function enables load balancing of Collector and sorting Flow Records without extra Collector functions. If Flow Records are used as accounting information, Mediator can distribute Flow Records to the dedicated Collector of each customer.

When network operators disclose traffic information to each customer, security or the privacy policy should be considered. In that case, the IPFIX Mediator hides private information about each customer. In addition, Mediator distributes traffic information based on RD (Route Distinguisher), ingress IF, peering AS number, or BGP next hop, which identify the customer. In the following figure, the IPFIX Mediator distributes Flow Records based on RD. The system securely allows each customer to access only their own records.

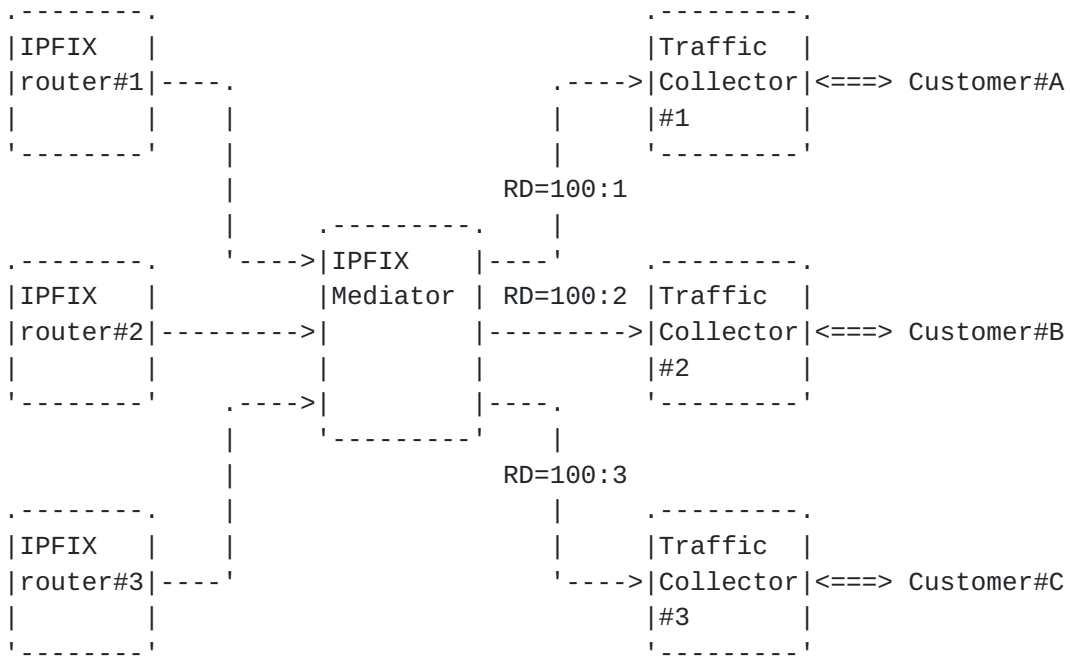

Figure F: Distribution of Flow Records for each customer.

4.5. Extraction of Suspicious Flow

An IPFIX Mediator performs filtering based on the value of specified Information Elements. Filter conditions are set depending on a suspicious flow as follows. The Collector receives the specified suspicious flow and detects an anomalous flow by simply monitoring the traffic volume of each suspicious flow.

- TCP Flow Records whose "tcpControlBits" value is set to "null"
- TCP Flow Records whose "tcpControlBits" value is set to the SYN bit only and the packet counter is only 1.
- ICMP Flow Records whose length is too long.

5. Mediator Option Template Presentation

This section describes Option Templates that are used by IPFIX Mediators.

5.1. Exporter Information Option Template

Each IPFIX Mediator and final destination Collector needs to know the Original Exporter and route of IPFIX Mediators. Therefore, each IPFIX Mediator informs the next Collector about previous Exporter information, which is the Exporter Information Option Template that specified the Original Exporter and the route of the IPFIX Mediator. The final destination Collector can recognize them by receiving this template. This template is composed of the following Information Elements.

- exporter{IPv4|IPv6}Address
- collector{IPv4|IPv6}Address
- exporterTransportPort
- collectorTransportPort
- collectorTransportProtocol
- observationDomainId

The Observation Domain ID of the Original Exporter or IPFIX Mediator is identified by specifying Exporter/Collector Information Elements, such as "collector{IPv4|IPv6}Address", "collectorTransportPort", "collectorTransportProtocol", "exporter{IPv4|IPv6}Address", "exporterTransportPort", and "observationDomainId". The set of "observationDomainId" and "templateId" or "observationDomainId" might be used as a scope field. Not all Information Elements are necessary. For example, the exporter{IPv4|IPv6}Address is necessary to inform the next Collector about the Original Exporter that created Flow Records. If the IPFIX Mediator receives this Template, it SHOULD not overwrite each field. The IPFIX Mediator appends its own previous Exporter information onto received Data Records specified by the Exporter Option Template and sends that information to the Collector. In this manner, the route is maintained until the final destination Collector.

The following example describes the cascade connection of IPFIX Mediators. Each Mediator informs the next Collector about previous Exporter information.

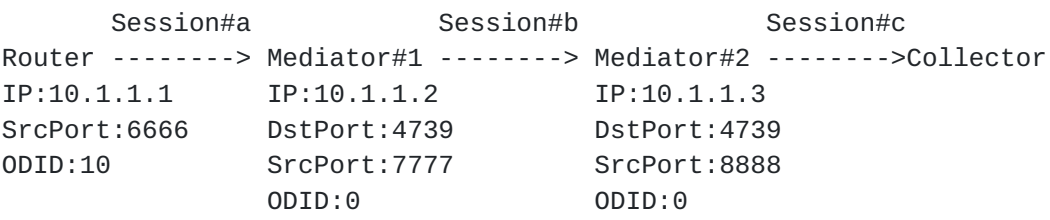

Figure G: Cascade connection of IPFIX Mediators.

Mediator#1 or Mediator#2 sends a Data Record specified by the Exporter Option Template. The Data records are shown in Session#b or Session#c, as follows.

Session#b Data Record:
- Field Count = 7
- Scope Count = 1
- templateId = XXX
- exporterIPv4Address = 10.1.1.1
- collectorIPv4Address = 10.1.1.2
- collectorTransportProtocol = 132
- exporterTransportPort = 6666
- collectorTransportPort = 4739
- observationDomainId = 10

Session#c Data Record:
- Field Count = 13
- Scope Count = 1
- templateId = XXX
- exporterIPv4Address = 10.1.1.1
- collectorIPv4Address = 10.1.1.2
- collectorTransportProtocol = 132
- exporterTransportPort = 6666
- collectorTransportPort = 4739
- observationDomainId = 10
- exporterIPv4Address = 10.1.1.2
- collectorIPv4Address = 10.1.1.3
- collectorTransportProtocol = 132
- exporterTransportPort = 7777
- collectorTransportPort = 4739
- observationDomainId = 0

5.2. Usage of Scope Field

An IPFIX Mediator needs to send Options Template Records and associated Data Records from the Original Exporter. However, IPFIX Mediator cannot export original Option Template Records and associated Data Records without modification because changing a session from an Exporting Process to a Collecting Process causes the scope fields to have a useless values. When an IPFIX Mediator relays the Option Template Records that included Observation Domain ID as a scope field and associated Data Records, an IPFIX Mediator uses the Exporter Information Option Template. Option Template Records that were created from an Original Exporter can use all fields of the Exporter Information Option template as multiple scope fields. Option Template Records that were created from an IPFIX Mediator can use some fields of the Exporter Information Option Template as multiple scope fields. An IPFIX Mediator needs to modify associated Data Records according to the modified Options Template Record. However, if each node uses another field, except for the Observation Domain ID, as the scope, the scope field should be considered on a case-by-case basis.

The following example describes the cascade connection of IPFIX Mediators. Router#1 and Mediator#1 export the Metering Process Statistics Option Template.

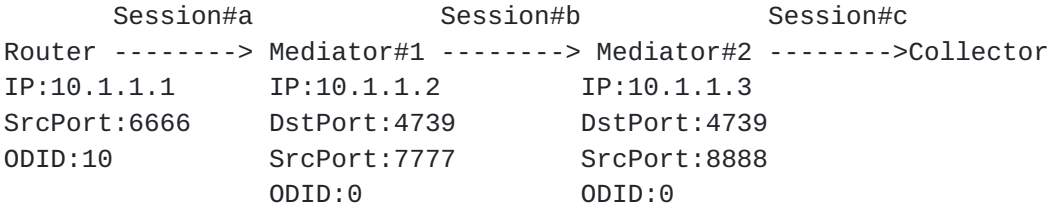

Figure H: Cascade connection of IPFIX Mediators.

Mediator#2 exports each Option Template and its Data Record with a suitable scope.

Session#c Metering Process Statistics Data Records from the Original Exporter:

- Field Count = 15
- Scope Count = 12
- exporterIPv4Address = 10.1.1.1
- collectorIPv4Address = 10.1.1.2
- collectorTransportProtocol = 132
- exporterTransportPort = 6666
- collectorTransportPort = 4739
- observationDomainId = 10

```
      exporterIPv4Address = 10.1.1.2
      collectorIPv4Address = 10.1.1.3
      collectorTransportProtocol = 132
      exporterTransportPort = 7777
      collectorTransportPort = 4739
      observationDomainId = 0
      exportedMessageTotalCount
      exportedFlowTotalCount
      exportedOctetTotalCount
```

6. Security Considerations

The IPFIX concentrator uses the IPFIX protocol. Security considerations about flow information are described in [RFC5101].

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. 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", January 2008.

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

8.2. Informative References

[I-D.dressler-ipfix-aggregation]
Dressler, F., Sommer, C., Munz, G., and A. Kobayashi, "IPFIX Aggregation", draft-dressler-ipfix-aggregation-04.txt (work in progress) , November 2007.

[I-D.ietf-ipfix-architecture]
Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", draft-ietf-ipfix-architecture-12.txt(work in progress) , September 2006.

[I-D.ietf-psamp-framework]
Duffield, N., "A Framework for Packet Selection and Reporting", draft-ietf-psamp-framework-12.txt , June 2007.

[I-D.ietf-psamp-sample-tech]
Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", draft-ietf-psamp-sample-tech-10.txt , June 2007.

[I-D.kobayashi-ipfix-large-ps]
Kobayashi, A., Nishida, H., Sommer, C., Dressler, F., and E. Stephan, "Problems with Flow Collection in Large-Scale Networks", draft-kobayashi-ipfix-large-ps-01.txt(work in progress) , February 2008.

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

Authors' Addresses

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

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

Phone: +81-422-59-3407
Email: ishibashi.keisuke@lab.ntt.co.jp

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

Phone: +81-422-59-2419
Email: kondoh.tsuyoshi@lab.ntt.co.jp

Daisuke Matsubara
Hitachi, Ltd., Central Reseach Laboratory
1-280 Higashi-koigakubo
Kokubunji-shi, Tokyo 185-8601
Japan

Phone: +81-42-323-1111
Email: daisuke.matsubara.pj@hitachi.com

Full Copyright Statement

Intellectual Property

Acknowledgment

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