Network Working Group
Request for Comments: 5472
Category: Informational

T. Zseby
Fraunhofer FOKUS
E. Boschi
Hitachi Europe
N. Brownlee
CAIDA
B. Claise
Cisco Systems, Inc.

March 2009

IP Flow Information Export (IPFIX) Applicability

Status of This Memo

Copyright Notice

Abstract

In this document, we describe the applicability of the IP Flow Information eXport (IPFIX) protocol for a variety of applications. We show how applications can use IPFIX, describe the relevant Information Elements (IEs) for those applications, and present opportunities and limitations of the protocol. Furthermore, we describe relations of the IPFIX framework to other architectures and frameworks.

Table of Contents

1. Introduction

The IPFIX protocol defines how IP Flow information can be exported from routers, measurement probes, or other devices. IP Flow information provides important input data for a variety of applications. The IPFIX protocol is a general data transport protocol that is easily extensible to suit the needs of such applications. In this document, we describe how typical applications can use the IPFIX protocol and show opportunities and limitations of the protocol. Furthermore, we describe the relationship of IPFIX to other frameworks and architectures. Although examples in this document are shown for IPv4 only, the applicability statements apply to IPv4 and IPv6. IPFIX provides appropriate Information Elements for both IP versions.

1.1. Terminology

IPFIX-specific terminology used in this document is defined in Section 2 of [RFC5101]. In this document, as in [RFC5101], the first letter of each IPFIX-specific term is capitalized.

2. Applications of IPFIX

IPFIX data enables several critical applications. The IPFIX target applications and the requirements that originate from those applications are described in [RFC3917]. Those requirements were used as basis for the design of the IPFIX protocol. This section describes how these target applications can use the IPFIX protocol. Considerations for using IPFIX for other applications than those described in [RFC3917] can be found in Section 4.1.

2.1. Accounting

Usage-based accounting is one of the target applications for IPFIX as defined in [RFC3917]. IPFIX records provide fine-grained measurement results for highly flexible and detailed usage reporting. Such data is used to realize usage-based accounting. Nevertheless, IPFIX does not provide the reliability required by usage-based billing systems as defined in [RFC2975] (see Section 4.2). The accounting scenarios described in this document only provide limited reliability as explained in Section 4.2 and should not be used in environments where reliability as demanded by [RFC2975] is mandatory.

In order to realize usage-based accounting with IPFIX, the Flow definition has to be chosen in accordance to the accounting purpose, such as trend analysis, capacity planning, auditing, or billing and cost allocation where some loss of data can be tolerated (see Section 4.2).

Flows can be distinguished by various IEs (e.g., packet header fields) from [RFC5102]. Due to the flexible IPFIX Flow definition, arbitrary Flow-based accounting models can be realized without extensions to the IPFIX protocol.

Accounting can, for instance, be based on individual end-to-end Flows. In this case, it can be realized with a Flow definition determined by the quintuple consisting of source address (sourceIPv4Address), destination address (destinationIPv4Address), protocol (protocolIdentifier), and port numbers (udpSourcePort, udpDestinationPort). Another example is class-dependent accounting (e.g., in a Diffserv network). In this case, Flows could be distinguished just by the Diffserv codepoint (DSCP) (ipDiffServCodePoint) and IP addresses (sourceIPv4Address, destinationIPv4Address). The essential elements needed for accounting are the number of transferred packets and bytes per Flow, which can be represented by the per-flow counter IEs (e.g., packetTotalCount, octetTotalCount).

For accounting purposes, it would be advantageous to have the ability to use IPFIX Flow Records as accounting input in an Authentication, Authorization, and Accounting (AAA) infrastructure. AAA servers then could provide the mapping between user and Flow information. Again for such scenarios the limited reliability currently provided by IPFIX has to be taken into account.

2.1.1. Example

Please note: As noted in [RFC3330], the address block 192.0.2.0/24 may be used for example addresses. In the example below, we use two example networks. In order to be conformant to [RFC3330], we divide the given address block into two networks by subnetting with a 25-bit netmask (192.0.2.0/25) as follows:

Network A: 192.0.2.0 ... 192.0.2.127
Network B: 192.0.2.128 ... 192.0.2.255

Let's suppose someone needs to monitor the individual Flows in a Diffserv network in order to compare traffic amount trend with the terms outlined in a Service Level Agreement (SLA). Flows are distinguished by source and destination address. The information to export in this case is:

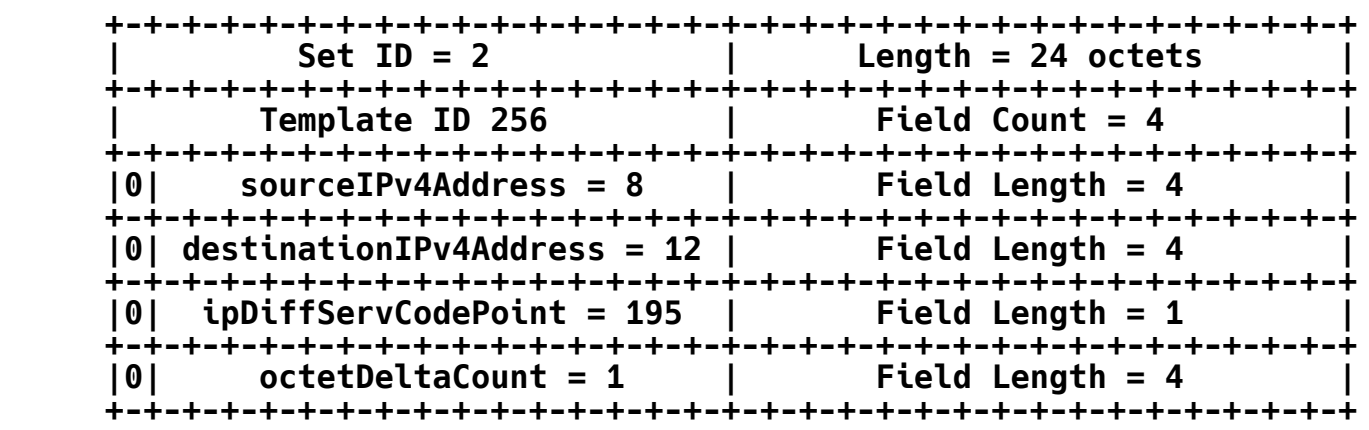
- IPv4 source IP address: sourceIPv4Address in [RFC5102], with a length of 4 octets

- IPv4 destination IP address: destinationIPv4Address in [RFC5102], with a length of 4 octets

- DSCP: ipDiffServCodePoint in [RFC5102], with a length of 1 octet

- Number of octets of the Flow: octetDeltaCount in [RFC5102], with a length of 4 octets

The Template set will look as follows:

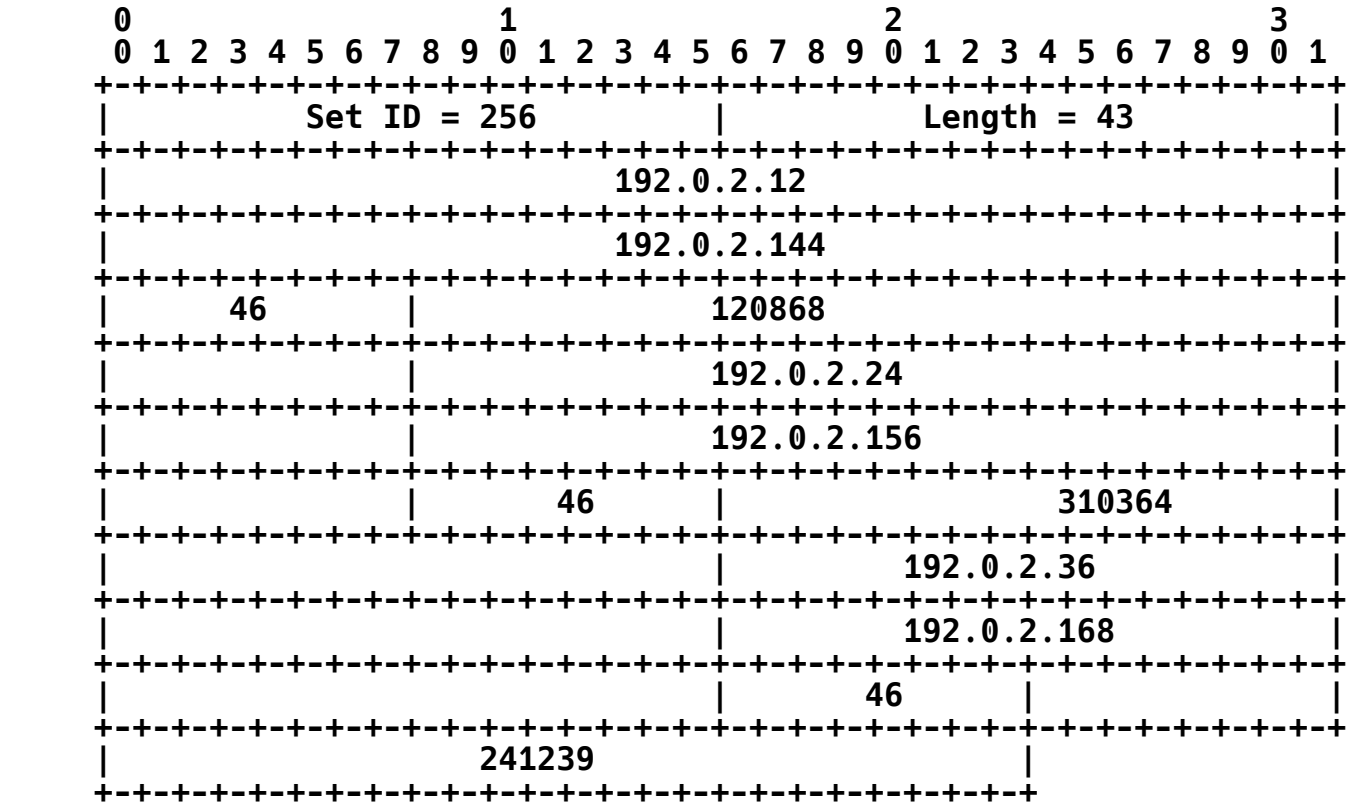


The information to be exported might be as listed in the following example table:

Src. IP addr.	Dst. IP addr.	DSCP	Octets Number
192.0.2.12	192.0.2.144	46	120868
192.0.2.24	192.0.2.156	46	310364
192.0.2.36	192.0.2.168	46	241239

In the example we use Diffserv codepoint 46, recommended for the Expedited Forwarding Per Hop Behavior (EF PHB) in [RFC3246].

The Flow Records will then look as follows:



2.2. Traffic Profiling

Measurement results reported in IPFIX records can provide useful input for traffic profiling. IPFIX records captured over a long period of time can be used to track and anticipate network growth and usage. Such information is valuable for trend analysis and network planning.

The parameters of interest are determined by the profiling objectives. Example parameters for traffic profiling are Flow duration, Flow volume, burstiness, the distribution of used services and protocols, the amount of packets of a specific type, etc. [RFC3917].

The distribution of services and protocols in use can be analyzed by configuring appropriate Flows Keys for Flow discrimination. Protocols can be distinguished by the protocolIdentifier IE. Portnumbers (e.g., udpDestinationPort) often provide information about services in use. Those Flow Keys are defined in [RFC5102]. If

portnumbers are not sufficient for service discrimination, further parts of the packet may be needed. Header fields can be expressed by IEs from [RFC5102].

Packet payload can be reported by using the IE ipPayloadPacketSection in [RFC5477].

The Flow duration can be calculated from the Flow Timestamp IEs defined in [RFC5102] (e.g., flowEndMicroseconds - flowStartMicroseconds). The number of packets and number of bytes of a Flow are represented in the per-flow counter IEs (e.g., packetTotalCount, octetTotalCount). The burstiness of a Flow can be calculated from the Flow volume measured at different time intervals.

2.3. Traffic Engineering

Traffic engineering aims at the optimization of network resource utilization and traffic performance [RFC2702]. Typical parameters are link utilization, load between specific network, nodes, number, size and entry/exit points of active Flows, and routing information [RFC3917].

The size of Flows in packets and bytes can be reported by the IEs packetTotalCount and octetTotalCount. Utilization of a physical link can be reported by using a coarse-grained Flow definition (e.g., based on identifier IEs such as egressInterface or ingressInterface) and per-flow counter IEs (e.g., packetTotalCount, octetTotalCount) defined in [RFC5102].

The load between specific network nodes can be reported in the same way if one interface of a network node receives only traffic from exactly one neighbor node (as is usually the case). If the ingress interface is not sufficient for an unambiguous identification of the neighbor node, sub-IP header fields IEs (like sourceMacAddress) can be added as Flow Keys.

The IE observedFlowTotalCount provides the number of all Flows exported for the Observation Domain since the last initialization of the Metering Process [RFC5102]. If this IE is exported at subsequent points in time, one can derive the number of active Flows in a specific time interval from the difference of the reported counters. The configured Flow termination criteria have to be taken into account to interpret those numbers correctly.

Entry and exit points can be derived from Flow Records if Metering Processes are installed at all edges of the network and results are mapped in accordance to Flow Keys. For this and other analysis methods that require the mapping of records from different

Observation Points, the same Flow Keys should be used at all Observation Points. The path that packets take through a network can be investigated by using hash-based sampling techniques as described in [DuGr00] and [RFC5475]. For this, IEs from [RFC5477] are needed.

Neither [RFC5102] nor [RFC5477] defines IEs suitable for exporting routing information.

2.4. Network Security

Attack and intrusion detection are among the IPFIX target applications described in [RFC3917]. Due to the enormous amount of different network attack types, only general requirements could be addressed in [RFC3917].

The number of metrics useful for attack detection is as diverse as attack patterns themselves. Attackers adapt rapidly to circumvent detection methods and try to hide attack patterns using slow or stealth attacks. Furthermore, unusual traffic patterns are not always caused by malicious activities. A sudden traffic increase may be caused by legitimate users who seek access to a recently published web content. Strange traffic patterns may also be caused by misconfiguration.

IPFIX can export Flow information for arbitrary Flow definitions as defined in [RFC5101]. Packet information can be exported with IPFIX by using the additional Information Elements described in [RFC5477]. With this, theoretically all information about traffic in the network at the IP layer and above is accessible. This data either can be used directly to detect anomalies or can provide the basis for further post-processing to generate more complex attack detection metrics.

Depending on the attack type, different metrics are useful. A sudden increase of traffic load can be a hint that an attack has been launched. The overall traffic at an Observation Point can be monitored using per-flow counter IEs like packetTotalCount or octetTotalCount as described in Section 2.3. The number of active Flows can be monitored by regular reporting of the observedFlowTotalCount defined in [RFC5102].

A sudden increase of Flows from different sources to one destination may be caused by an attack on a specific host or network node using spoofed addresses. The number of Flows from or to specific networks or hosts can be observed by using source and destination addresses as Flow Keys and observing the number of active Flows as explained above. Many Flows to the same machine, but on different ports, or many Flows to the same port and different machines may be an

indicator for vertical or horizontal port scanning activities. The number of Flows to different ports can be reported by using the portnumber Information Elements (udpSourcePort, udpDestinationPort, tcpSourcePort, tcpDestinationPort) defined in [RFC5102] as Flow Keys.

An unusual ratio of TCP-SYN to TCP-FIN packets can refer to SYN-flooding. The number of SYN and FIN packets in a Flow can be reported with the IPFIX Information Elements tcpSynTotalCount and tcpFinTotalCount defined in [RFC5102].

Worms may leave signatures in traffic patterns. Detecting such events requires more detailed measurements and post-processing than detecting simple changes in traffic volumes.

A difficult task is the separation of good from bad packets to prepare and launch counteraction. This may require a deeper look into packet content by using further header field IEs from [RFC5102] and/or packet payloads from IE ipPayloadPacketSection in [RFC5477].

Furthermore, the amount of resources needed for measurement and reporting increases with the level of granularity required to detect an attack. Multi-step analysis techniques may be useful, e.g., to launch an in-depth analysis (e.g., based on packet information) in case the Flow information shows suspicious patterns. In order to supervise traffic to a specific host or network node, it is useful to apply filtering methods such as those described in [RFC5475].

Mapping the two directions of communication is often useful for checking correct protocol behavior (see Section 4.6). A correlation of IPFIX data from multiple Observation Points (see Section 2.5.1) allows assessing the propagation of an attack and can help to locate its source.

The integration of previous measurement results helps to review traffic changes over time for detection of traffic anomalies and provides the basis for forensic analysis. A standardized storage format for IPFIX data would support the offline analysis of data from different operators.

Nevertheless, capturing full packet traces at all Observation Points in the network is not viable due to resource limitations and privacy concerns. Therefore, metrics should be chosen wisely to allow a solid detection with minimal resource consumption. Resources can be saved, for instance, by using coarser-grained Flow definitions, reporting pre-processed metrics (e.g., with additional Information Elements), or deploying sampling methods.

In many cases, only derived metrics provide sufficient evidence about security incidents. For example, comparing the number of SYN and FIN packets for a specific time interval can reveal an ongoing SYN attack, which is not obvious from unprocessed packet and Flow data. Further metrics like the cumulated sum of various counters, distributions of packet attributes, or spectrum coefficients have been used to identify a variety of attacks.

In order to detect attacks early, it is useful to process the data as soon as possible in order to generate significant metrics for the detection. Pre-processing of raw packet and Flow data already at the measurement device can speed up the detection process and reduces the amount of data that need to be exported. Furthermore, it is possible to directly report derived metrics by defining appropriate Information Elements. Immediate data export in case of a potential incident is desired. IPFIX supports such source-triggered exporting of information due to the push model approach. Nevertheless, further exporting criteria have to be implemented to export IPFIX records upon incident detection events and not only upon flow-end or fixed-time intervals.

Intrusion detection would profit from the combination of IPFIX functions with AAA functions (see Section 3.5). Such an interoperation enables further means for attacker detection, advanced defense strategies, and secure inter-domain cooperation.

2.5. QoS Monitoring

Quality of service (QoS) monitoring is one target application of the IPFIX protocol [RFC3917]. QoS monitoring is the passive observation of the transmission quality for single Flows or traffic aggregates in the network. One example of its use is the validation of QoS guarantees in service level agreements (SLAs). Typical QoS parameters are loss [RFC2680], one-way [RFC2679] and round-trip delay [RFC2681], and delay variation [RFC3393]. Whenever applicable, the IP Performance Metrics (IPPM) definitions [RFC4148] should be used when reporting QoS metrics.

The calculation of those QoS metrics requires per-packet processing. Reporting packet information with IPFIX is possible by simply considering a single packet as Flow. [RFC5101] also allows the reporting of multiple identical Information Elements in one Flow Record. Using this feature for reporting information about multiple packets in one record would require additional agreement on semantics regarding the order of Information Elements (e.g., which timestamp belongs to which packet payload in a sequence of Information Elements). [RFC5477] defines useful additional Information Elements for exporting per-packet information with IPFIX.

2.5.1. Correlating Events from Multiple Observation Points

Some QoS metrics require the correlation of data from multiple Observation Points. For this, the clocks of the involved Metering Processes must be synchronized. Furthermore, it is necessary to recognize that the same packet was observed at different Observation Points.

This can be done by capturing parts of the packet content (packet header and/or parts of the payload) that do not change on the way to the destination. Based on the packet content, it can be recognized when the same packet arrived at another Observation Point. To reduce the amount of measurement data, a unique packet ID can be calculated from the packet content, e.g., by using a Cyclic Redundancy Check (CRC) or hash function instead of transferring and comparing the unprocessed content. Considerations on collision probability and efficiency of using such packet IDs are described in [GrDM98], [DuGr00], and [ZsZC01].

IPFIX allows the reporting of several IP and transport header fields (see Sections 5.3 and 5.4 in [RFC5102]). Using only those fields for packet recognition or ID generation can be sufficient in scenarios where those header fields vary a lot among subsequent packets, where a certain amount of packet ID collisions are tolerable, or where packet IDs need to be unique only for a small time interval.

For including packet payload information, the Information Element ipPayloadPacketSection defined in [RFC5477] can be used. The Information Element ipHeaderPacketSection can also be used. However, header fields that can change on the way from source to destination have to be excluded from the packet ID generation because they may differ at different Observation Points.

For reporting packet IDs generated by a CRC or hash function, the Information Element digestHashValue defined in [RFC5477] can be used.

2.5.2. Examples

The following examples show which Information Elements need to be reported by IPFIX to generate specific QoS metrics. As an alternative, the metrics can be generated directly at the exporter and IPFIX can be used to export the metrics (see Section 2.7).

2.5.2.1. RTT Measurements with Packet Pair Matching (Single-Point)

The passive measurement of round-trip time (RTT) can be performed by using packet pair matching techniques as described in [Brow00]. For the measurements, request/response packet pairs from protocols such

as DNS, ICMP, SNMP or TCP (SYN/SYN_ACK, DATA/ACK) are utilized to passively observe the RTT [Brow00]. This technique requires the correlation of data from both directions.

Required Information Elements per packet (DNS example):
- Packet arrival time: observationTimeMicroseconds [RFC5477]
- DNS header: ipPayloadPacketSection [RFC5477]

Required functions:
- Recognition of request/response packet pairs

Remarks:
- Requires Information Elements from [RFC5477].
- observationTimeMicroseconds can be substituted by flowStartMicroseconds [RFC5102] because a single packet can be represented as a Flow.
- If time values with a finer granularity are needed, observationTimeNanoseconds can be used.

2.5.2.2. One-Way Delay Measurements (Multi-Point)

Passive one-way delay measurements require the collection of data at two Observation Points. As mentioned above, synchronized clocks are needed to avoid time-differences at the involved Observation Points.

The recognition of packets at the second Observation Point can be based on parts of the packet content directly. A more efficient way is to use a packet ID (generated from packet content).

Required Information Elements per packet (with packet ID):
- Packet arrival time: observationTimeMicroseconds [RFC5477]
- Packet ID: digestHashValue [RFC5477]

Required functions:
- Packet ID generation
- Delay calculation (from arrival times at the two Observation Points)

Remarks:
- Requires Information Elements from [RFC5477].
- observationTimeMicroseconds can be substituted by flowStartMicroseconds [RFC5102], because a single packet can be represented as a Flow.
- If time values with a finer granularity are needed, observationTimeNanoseconds can be used.

- The amount of content used for ID generation influences the number of collisions (different packets that map to the same ID) that can occur. Investigations on this and other considerations on packet ID generation can be found in [GrDM98], [DuGr00], and [ZsZC01].

2.6. Inter-Domain Exchange of IPFIX Data

IPFIX data can be used to share information with neighbor providers. A few recommendations should be considered if IPFIX records travel over the public Internet, compared to its usage within a single domain. First of all, security threat levels are higher if data travels over the public Internet. Protection against disclosure or manipulation of data is even more important than for intra-domain usage. Therefore, Transport Layer Security (TLS) or Datagram Transport Layer Security should be used as described in [RFC5101].

Furthermore, data transfer should be congestion-aware in order to allow untroubled coexistence with other data Flows in public or foreign networks. That means transport over Stream Control Transmission Protocol (SCTP) or TCP is required.

Some ISPs are still reluctant to share information due to concerns that competing ISPs might exploit network information from neighbor providers to strengthen their own position in the market. Nevertheless, technical needs have already triggered the exchange of data in the past (e.g., exchange of routing information by BGP). The need to provide inter-domain guarantees is one big incentive to increase inter-domain cooperation. The necessity to defend networks against current and future threats (denial-of-service attacks, worm distributions, etc.) will hopefully increase the willingness to exchange measurement data between providers.

2.7. Export of Derived Metrics

The IPFIX protocol is used to transport Flow and packet information to provide the input for the calculation of a variety of metrics (e.g., for QoS validation or attack detection). IPFIX can also be used to transfer these metrics directly, e.g., if the metric calculation is co-located with the Metering and Exporting Processes.

It doesn't matter which measurement and post-processing functions are applied to generate a specific metric. IPFIX can be used to transport the results from passive and active measurements and from post-processing operations. For the reporting of derived metrics, additional Information Elements need to be defined.

For most QoS metrics like loss, delay, delay variation, etc., standard IPPM definitions exist. In case such metrics are reported with IPFIX, the IPPM standard definition should be used.

2.8. Summary

The following table shows an overview of the Information Elements required for the target applications described in [RFC3917] (M-mandatory, R-recommended, O-optional).

Application	[RFC5102]	[RFC5477]	additional IEs
Accounting	M	-	-
Traffic Profiling	M	O	-
Traffic Engineering	M	-	O (routing info)
Attack Detection	M	R	R (derived metrics)
QoS Monitoring	M	M (most metrics)	O (derived metrics)

For accounting, the IEs in [RFC5102] are sufficient. As mentioned above, IPFIX does not conform to the reliability requirements demanded by [RFC2975] for usage-based billing systems (see Section 4.2). For traffic profiling, additional IEs from [RFC5477] can be useful to gain more insight into the traffic. For traffic engineering, Flow information from [RFC5102] is sufficient, but it would profit from routing information, which could be exported by IPFIX. Attack detection usually profits from further insight into the traffic. This can be achieved with IEs from [RFC5477]. Furthermore, the reporting of derived metrics in additional IEs would be useful. Most QoS metrics require the use of IEs from [RFC5477]. IEs from [RFC5477] are also useful for the mapping of results from different Observation Points as described in Section 2.5.1.

3. Relation of IPFIX to Other Frameworks and Protocols

3.1. IPFIX and IPv6

From the beginning, IPFIX has been designed for IPv4 and IPv6. Therefore, IPFIX can be used in IPv4 and IPv6 networks without limitations. The usage of IPFIX in IPv6 networks has two aspects:

- Generation and reporting of IPFIX records about IPv6 traffic
- Exporting IPFIX records over IPv6

The generation and reporting of IPFIX records about IPv6 traffic is possible. Appropriate Information Elements for the reporting of IPv6 traffic are defined in [RFC5102]. Exporting IPFIX records over IPv6 is not explicitly addressed in [RFC5101]. Since IPFIX runs over a transport protocol (SCTP, PR-SCTP, UDP, or TCP) and all potential IPFIX transport protocols can run in IPv6 networks, one just needs to provide the chosen transport protocol in the IPv6 network to run IPFIX over IPv6.

3.2. IPFIX and PSAMP

PSAMP defines packet selection methods, their configuration at routers and probes, and the reporting of packet information.

PSAMP uses IPFIX as a basis for exporting packet information [RFC5476]. [RFC5477] describes further Information Elements for exporting packet information and reporting configuration information.

The main difference between IPFIX and PSAMP is that IPFIX addresses the export of Flow Records, whereas PSAMP addresses the export of packet records. Furthermore, PSAMP explicitly addresses remote configuration. It defines a MIB for the configuration of packet selection processes. Remote configuration is not (yet) addressed in IPFIX, but one could consider extending the PSAMP MIB to also allow configuration of IPFIX processes.

3.3. IPFIX and RMON

Remote Monitoring (RMON) [RFC3577] is a widely used monitoring system that gathers traffic data from RMON Agents in network devices. One major difference between RMON and IPFIX is that RMON uses SNMP for data export, whereas IPFIX defines its own push-oriented protocol. RMON defines MIBs that contain the information to be exported. In IPFIX, the data to be exported is defined as Information Elements.

The most relevant MIBs for comparison with IPFIX are the Application Performance Measurement MIB (APM-MIB) [RFC3729] and the Transport Performance Metrics MIB (TPM-MIB) [RFC4150]. The APM-MIB has a complex system for tracking user application performance, with reporting about transactions and SLA threshold notification-trigger configuration, and persistence across DHCP lease expirations. It requires a full RMON2-MIB protocolDirTable implementation.

The APM-MIB reports the performance of transactions. A transaction is a service-oriented term and describes the data exchange from the transaction start (when a user requests a service) until its completion. The performance parameters include response times, throughput, streaming responsiveness, and availability of services.

The RMON transaction concept differs from the IPFIX Flow concept. A Flow is a very generic term that allows one to group IP packets in accordance with common properties. In contrast to this, the term transaction is service-oriented and contains all data exchange required for service completion.

In order to report such data with IPFIX, one would probably need a specific combination of multiple Flows and the ability to map those to the transaction. Due to the service-oriented focus of APM, the required metrics also differ. For instance, the RMON APM requires a metric for the responsiveness of services. Such metrics are not addressed in IPFIX.

Furthermore, the APM-MIB allows the configuration of the transaction type to be monitored, which is currently not addressed in IPFIX.

The APM MIB could be considered as an extension of the IPFIX Metering Process where the application performance of a combination of multiple Flows is measured. If appropriate, IEs would be defined in the IPFIX information model and the IPFIX Device would support the APM MIB data collection, the solutions could be complementary. That means one could use IPFIX to export APM MIB transaction information.

The TPM-MIB breaks out the APM-MIB transactions into sub-application level transactions. For instance, a web request is broken down into DNS, TCP, and HTTP sub-transactions. Such sub-transactions can be considered as bidirectional Flows. With an appropriate Flow definition and the ability to map both directions of a Flow (see Section 4.6), one could measure and report Flow characteristics of such sub-application level transaction with IPFIX.

The TPM-MIB requires APM-MIB and RMON2-MIB.

3.4. IPFIX and IPPM

The IPFIX protocol can be used to carry IPPM network performance metrics or information that can be used to calculate those metrics (see Sections 2.5 and 2.7 for details and references).

3.5. IPFIX and AAA

AAA defines a protocol and architecture for authentication, authorization, and accounting for service usage [RFC2903]. The DIAMETER protocol [RFC3588] is used for AAA communication, which is needed for network access services (Mobile IP, NASREQ, and ROAMOPS). The AAA architecture [RFC2903] provides a framework for extending AAA support to other services. DIAMETER defines the exchange of messages between AAA entities, e.g., between AAA clients at access devices and AAA servers, and among AAA servers. DIAMETER is used for the transfer of accounting records. In order to form accounting records for usage-based accounting measurement, data from the network is required. IPFIX defines a protocol to export such data from routers, measurement probes, and other devices. Therefore, it looks promising to connect those two architectures.

For all scenarios described here, one has to keep in mind that IPFIX does not conform to the reliability requirements for usage-based billing described in [RFC2975] (see Section 4.2). Using IPFIX without reliability extensions together with AAA would result in accounting scenarios that do not conform to usage-based billing requirements described in [RFC2975].

As shown in Section 2.1, accounting applications can directly incorporate an IPFIX Collecting Process to receive IPFIX records with information about the transmitted volume. Nevertheless, if a AAA infrastructure is in place, the cooperation between IPFIX and AAA provides many valuable synergistic benefits. IPFIX records can provide the input for AAA accounting functions and provide the basis for the generation of DIAMETER accounting records. However, as stated in Section 4.2, the use of IPFIX as described in [RFC5101] is currently limited to situations where the purpose of the accounting does not require reliability.

Further potential features include the mapping of a user ID to Flow information (by using authentication information) or using the secure authorized exchange of DIAMETER accounting records with neighbor domains. The last feature is especially useful in roaming scenarios where the user connects to a foreign network and the home provider generates the invoice.

Coupling an IPFIX Collecting Process with AAA functions also has high potential for intrusion and attack detection. AAA controls network access and maintains data about users and nodes. AAA functions can help to identify the source of malicious traffic. Authorization functions are able to deny access to suspicious users or nodes. Therefore, coupling those functions with an IPFIX Collecting Process can provide an efficient defense against network attacks.

Sharing IPFIX records (either directly or encapsulated in DIAMETER) with neighbor providers allows an efficient inter-domain attack detection. For this, it would be useful to allow remote configuration of measurement and record generation in order to provide information in the required granularity and accuracy. Since remote configuration is currently not addressed in IPFIX, this would require additional work. The AAA infrastructure itself may be used to configure measurement functions in the network as proposed in [RFC3334].

Furthermore, the transport of IPFIX records with DIAMETER would require the translation of IPFIX Information Elements into DIAMETER attribute value pairs (AVPs) defined in [RFC3588]. Since the DIAMETER AVPs do not comprise all IPFIX Information Elements, it is necessary to define new AVPs to transport them over DIAMETER.

Two possibilities exist to connect IPFIX and AAA:

- Connecting via a AAA Client
- Connecting via an Application Specific Module (ASM)

Both are explained in the following sections. The approaches only require a few additional functions. They do not require any changes to IPFIX or DIAMETER.

3.5.1. Connecting via a AAA Client

One possibility of connecting IPFIX and AAA is to run a AAA client on the IPFIX Collector. This client can generate DIAMETER accounting messages and send them to a AAA server. The mapping of the Flow information to a user ID can be done in the AAA server by using data from the authentication process. DIAMETER accounting messages can be sent to the accounting application or to other AAA servers (e.g., in roaming scenarios).

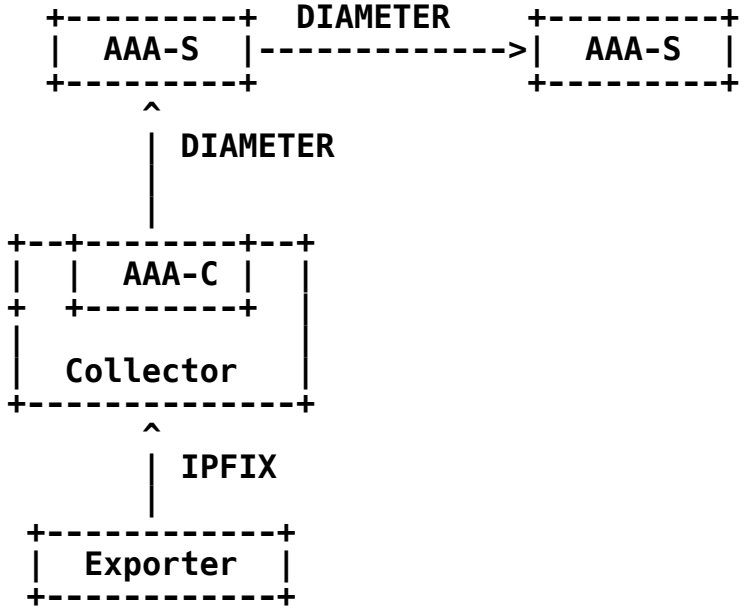


Figure 1: IPFIX Collector connects to AAA server via AAA client

3.5.2. Connecting via an Application Specific Module (ASM)

Another possibility is to directly connect the IPFIX Collector with the AAA server via an application specific module (ASM). Application specific modules have been proposed by the IRTF AAA architecture research group (AAARCH) in [RFC2903]. They act as an interface between AAA server and service equipment. In this case, the IPFIX Collector is part of the ASM. The ASM acts as an interface between the IPFIX protocol and the input interface of the AAA server. The ASM translates the received IPFIX data into an appropriate format for the AAA server. The AAA server then can add information about the user ID and generate a DIAMETER accounting record. This accounting record can be sent to an accounting application or to other AAA servers.

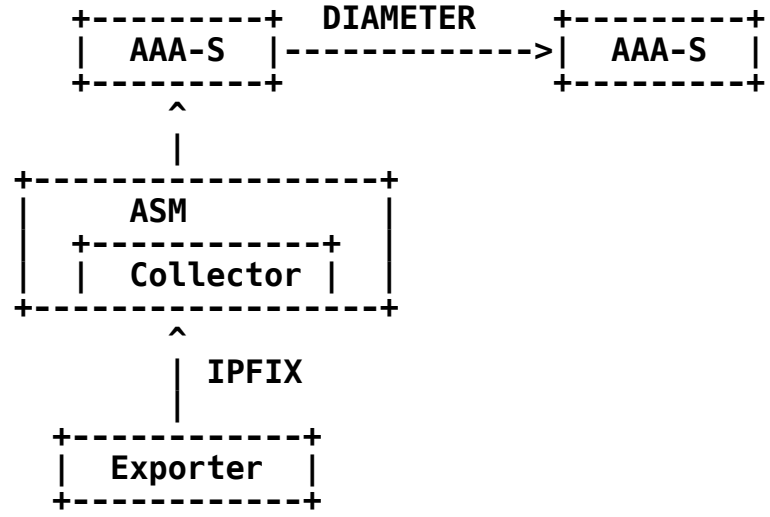


Figure 2: IPFIX connects to AAA server via ASM

3.6. IPFIX and RTFM

The Realtime Traffic Flow Measurement (RTFM) working group defined an architecture for Flow measurement [RFC2722]. This section compares the RTFM framework with the IPFIX framework.

3.6.1. Architecture

The RTFM architecture [RFC2722] is very similar to the IPFIX architecture. It defines meter, meter reader, and a manager as building blocks of the measurement architecture. The manager configures the meter, and the meter reader collects data from the meter. In RTFM, the building blocks communicate via SNMP.

The IPFIX architecture [RFC5470] defines Metering, Exporting, and Collecting Processes. IPFIX speaks about processes instead of devices to clarify that multiple of those processes may be co-located on the same machine.

These definitions do not contradict each other. One could see the Metering Process as part of the meter, and the Collecting Process as part of the meter reader.

One difference is that IPFIX currently does not define a managing process because remote configuration was (at least initially) out of scope for the working group.

3.6.2. Flow Definition

RTFM and IPFIX both consider Flows as a group of packets that share a common set of properties. A Flow is completely specified by that set of values, together with a termination criterion (like inactivity timeout).

A difference is that RTFM defines Flows as bidirectional. An RTFM meter matches packets from B to A and A to B as separate parts of a single Flow, and it maintains two sets of packet and byte counters, one for each direction.

IPFIX does not explicitly state whether Flows are uni- or bidirectional. Nevertheless, Information Elements for describing Flow properties were defined for only one direction in [RFC5102]. There are several solutions for reporting bidirectional Flow information (see Section 4.6).

3.6.3. Configuration and Management

In RTFM, remote configuration is the only way to configure a meter. This is done by using SNMP and a specific Meter MIB [RFC2720]. The IPFIX group currently does not address IPFIX remote configuration.

IPFIX Metering Processes export the layout of data within their Templates, from time to time. IPFIX Collecting Processes use that Template information to determine how they should interpret the IPFIX Flow data they receive.

3.6.4. Data Collection

One major difference between IPFIX and RTFM is the data collection model. RTFM retrieves data in pull mode, whereas IPFIX uses a push mode model to send data to Collecting Processes.

An RTFM meter reader pulls data from a meter by using SNMP. SNMP security on the meter determines whether a reader is allowed to pull data from it. An IPFIX Exporting Process is configured to export records to a specified list of IPFIX Collecting Processes. The condition of when to send IPFIX records (e.g., Flow termination) has to be configured in the Exporting or Metering Process.

3.6.5. Data Model Details

RTFM defines all its attributes in the RTFM Meter MIB [RFC2720]. IPFIX Information Elements are defined in [RFC5102].

RTFM uses continuously-incrementing 64-bit counters for the storage of the number of packets of a Flow. The counters are never reset and just wrap back to zero if the maximum value is exceeded. Flows can be read at any time. The difference between counter readings gives the counts for activity in the interval between readings.

IPFIX allows absolute (totalCounter) and relative counters (deltaCounter) [RFC5102]. The totalCounter is never reset and just wraps to zero if values are too large, exactly as the counters used in RTFM. The deltaCounter is reset to zero when the associated Flow Record is exported.

3.6.6. Transport Protocol

RTFM has a Standards-Track Meter MIB [RFC2720], which is used both to configure a meter and to store metering results. The MIB provides a way to read lists of attributes with a single Object Identifier (called a 'package'), which reduces the SNMP overhead for Flow data collection. SNMP, of course, normally uses UDP as its transport protocol. Since RTFM requires a reliable Flow data transport system, an RTFM meter reader must time out and resend unanswered SNMP requests. Apart from being clumsy, this can limit the maximum data transfer rate from meter to meter reader.

IPFIX is designed to work over a variety of different transport protocols. SCTP [RFC4960] and PR-SCTP [RFC3758] are mandatory. UDP and TCP are optional. In addition, the IPFIX protocol encodes data much more efficiently than SNMP does, hence IPFIX has lower data transport overheads than RTFM.

3.6.7. Summary

IPFIX exports Flow information in a push model by using SCTP, TCP, or UDP. It currently does not address remote configuration. RTFM data collection is using the pull model and runs over SNMP. RTFM

addresses remote configuration, which also runs over SNMP. Both frameworks allow a very flexible Flow definition, although RTFM is based on a bidirectional Flow definition.

4. Limitations

The goal of this section is to show the limitations of IPFIX and to give advice where not to use IPFIX or in which cases additional considerations are required.

4.1. Using IPFIX for Other Applications than Listed in RFC 3917

IPFIX provides a generic export mechanism. Due to its Template-based structure, it is a quite flexible protocol. Network operators and users may want to use it for other applications than those described in [RFC3917].

Apart from sending raw Flow information, it can be used to send per-packet data, aggregated or post-processed data. For this, new Templates and Information Elements can be defined if needed. Due to its push mode operation, IPFIX is also suited to send network initiated events like alarms and other notifications. It can be used for exchanging information among network nodes to autonomously improve network operation.

Nevertheless, the IPFIX design is based on the requirements that originate only from the target applications stated in [RFC3917]. Using IPFIX for other purposes requires a careful checking of IPFIX capabilities against application requirements. Only with this, one can decide whether IPFIX is a suitable protocol to meet the needs of a specific application.

4.2. Using IPFIX for Billing (Reliability Limitations)

The reliability requirements defined in [RFC3917] are not sufficient to guarantee the level of reliability that is needed for usage-based billing systems as described in [RFC2975]. In particular, IPFIX does not support the following features required by [RFC2975]:

- Record loss: IPFIX allows the usage of different transport protocols for the transfer of data records. Resilience against the loss of IPFIX data records can be only provided if TCP or SCTP is used for the transfer of data records.

- Network or device failures: IPFIX does allow the usage of multiple Collectors for one Exporter, but it neither specifies nor demands the use of multiple Collectors for the provisioning of fault tolerance.

- Detection and elimination of duplicate records: This is currently not supported by IPFIX.

- Application layer acknowledgements: IPFIX does not support the control of measurement and Exporting Processes by higher-level applications. Application layer acknowledgements are necessary, e.g., to inform the Exporter in case the application is not able to process the data exported with IPFIX. Such acknowledgements are not supported in IPFIX.

Further features like archival accounting and pre-authorization are out of scope of the IPFIX specification but need to be realized in billing system architectures as described in [RFC2975].

4.3. Using a Different Transport Protocol than SCTP

SCTP is the preferred protocol for IPFIX, i.e., a conforming implementation must work over SCTP. Although IPFIX can also work over TCP or UDP, both protocols have drawbacks [RFC5101]. Users should make sure they have good reasons before using protocols other than SCTP in a specific environment.

4.4. Push vs. Pull Mode

IPFIX works in push mode. That means IPFIX records are automatically exported without the need to wait for a request. The responsibility for initiating a data export lies with the Exporting Process.

Criteria for exporting data need to be configured at the Exporting Process. Therefore, push mode has more benefits if the trigger for data export is related to events at the Exporting Process (e.g., Flow termination, memory shortage due to large amount of Flows, etc.). If the protocol used pull mode, the Exporting Process would need to wait for a request to send the data. With push mode, it can send data immediately, e.g., before memory shortage would require a discarding of data.

With push mode, one can prevent the overloading of resources at the Exporting Process by simply exporting the information as soon as certain thresholds are about to be exceeded. Therefore, exporting criteria are often related to traffic characteristics (e.g., Flow timeout) or resource limitations (e.g., size of Flow cache). However, traffic characteristics are usually quite dynamic and often impossible to predict. If they are used to trigger Flow export, the exporting rate and the resource consumption for Flow export becomes variable and unpredictable.

Pull mode has advantages if the trigger for data export is related to events at the Collecting Process (e.g., a specific application requests immediate input).

In a pull mode, a request could simply be forwarded to the Exporting Process. In a push mode, the exporting configuration must be changed to trigger the export of the requested data. Furthermore, with pull mode, one can prevent the overloading of the Collecting Process by the arrival of more records than it can process.

Whether this is a relevant drawback depends on the flexibility of the IPFIX configuration and how IPFIX configuration rules are implemented.

4.5. Template ID Number

The IPFIX specification limits the different Template ID numbers that can be assigned to the newly generated Template records in an Observation Domain. In particular, Template IDs up to 255 are reserved for Template or option sets (or other sets to be created) and Template IDs from 256 to 65535 are assigned to data sets. In the case of many exports requiring many different Templates, the set of Template IDs could be exhausted.

4.6. Exporting Bidirectional Flow Information

Although IPFIX does not explicitly state that Flows are unidirectional, Information Elements that describe Flow characteristics are defined only for one direction in [RFC5102]. [RFC5101] allows the reporting of multiple identical Information Elements in one Flow Record. With this, Information Elements for forward and reverse directions can be reported in one Flow Record.

However, this is not sufficient. Using this feature for reporting bidirectional Flow information would require an agreement on the semantics of Information Elements (e.g., first counter is the counter for the forward direction, the second counter for the reverse direction).

Another option is to use two adjacent Flow Records to report both directions of a bidirectional Flow separately. This approach requires additional means for mapping those records and is quite inefficient due to the redundant reporting of Flow Keys.

4.7. Remote Configuration

Remote configuration was initially out of scope of the IPFIX working group in order to concentrate on the protocol specification. Therefore, there is currently no standardized way to configure IPFIX processes remotely. Nevertheless, due to the broad need for this feature, it is quite likely that solutions for this will be standardized soon.

5. Security Considerations

This document describes the usage of IPFIX in various scenarios. Security requirements for IPFIX target applications and security considerations for IPFIX are addressed in [RFC3917] and [RFC5101]. Those requirements have to be met for the usage of IPFIX for all scenarios described in this document. To our current knowledge, the usage scenarios proposed in Section 2 do not induce further security hazards.

The threat level to IPIFX itself may depend on the usage scenario of IPFIX. The usage of IPFIX for accounting or attack detection may increase the incentive to attack IPFIX itself. Nevertheless, security considerations have to be taken into account in all described scenarios.

As described in the security considerations in [RFC5101], security incidents can become a threat to IPFIX processes themselves, even if IPIFX is not the target of the attack. If an attack generates a large amount of Flows (e.g., by sending packets with spoofed addresses or simulating Flow termination), Exporting and Collecting Processes may get overloaded by the immense amount of records that are exported. A flexible deployment of packet or Flow sampling methods can be useful to prevent the exhaustion of resources.

Section 3 of this document describes how IPFIX can be used in combination with other technologies. New security hazards can arise when two individually secure technologies or architectures are combined. For the combination of AAA with IPFIX, an application specific module (ASM) or an IPFIX Collector can function as a transit point for the messages. One has to ensure that at this point the applied security mechanisms (e.g., encryption of messages) are maintained.

6. Acknowledgements

We would like to thank the following people for their contributions, discussions on the mailing list, and valuable comments:

Sebastian Zander
Robert Loewe
Reinaldo Penno
Lutz Mark
Andy Biermann

Part of the work has been developed in the research project 6QM, co-funded with support from the European Commission.

7. Normative References

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.

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

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

[RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.

8. Informative References

[Brow00] Brownlee, N., "Packet Matching for NeTraMet Distributions", <http://www.caida.org/tools/measurement/netramet/packetmatching/>.

[DuGr00] Duffield, N. and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, August 28 - September 1, 2000.

[GrDM98] Graham, I., Donnelly, S., Martin, S., Martens, J., and J. Cleary, "Nonintrusive and Accurate Measurement of Unidirectional Delay and Delay Variation on the Internet", INET'98, Geneva, Switzerland, 21-24 July, 1998.

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC2720] Brownlee, N., "Traffic Flow Measurement: Meter MIB", RFC 2720, October 1999.

[RFC2722] Brownlee, N., Mills, C., and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999.

[RFC2903] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J., and D. Spence, "Generic AAA Architecture", RFC 2903, August 2000.

[RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to Accounting Management", RFC 2975, October 2000.

[RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[RFC3334] Zseby, T., Zander, S., and C. Carle, "Policy-Based Accounting", RFC 3334, October 2002.

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.

[RFC3577] Waldbusser, S., Cole, R., Kalbfleisch, C., and D. Romascanu, "Introduction to the Remote Monitoring (RMON) Family of MIB Modules", RFC 3577, August 2003.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3729] Waldbusser, S., "Application Performance Measurement MIB", RFC 3729, March 2004.

[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.

[RFC4150] Dietz, R. and R. Cole, "Transport Performance Metrics MIB", RFC 4150, August 2005.

[RFC4960] Stewart, R., Ed., "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.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, March 2009.

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

[ZsZC01] Zseby, T., Zander, S., and G. Carle, "Evaluation of Building Blocks for Passive One-way-delay Measurements", Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 23-24, 2001

Authors' Addresses

Tanja Zseby
Fraunhofer Institute for Open Communication Systems (FOKUS)
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
Phone: +49 30 3463 7153
EMail: tanja.zseby@fokus.fraunhofer.de

Elisa Boschi
Hitachi Europe
c/o ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland
Phone: +41 44 6327057
EMail: elisa.boschi@hitachi-eu.com

Nevil Brownlee
CAIDA (UCSD/SDSC)
9500 Gilman Drive
La Jolla, CA 92093-0505
Phone: +1 858 534 8338
EMail: nevil@caida.org

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
1831 Diegem
Belgium
Phone: +32 2 704 5622
EMail: bclaise@cisco.com