Network Working Group
Request for Comments: 3955
Category: Informational

S. Leinen
SWITCH

October 2004

# Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)

## Status of this Memo



## Copyright Notice



## Abstract

This document contains an evaluation of the five candidate protocols for an IP Flow Information Export (IPFIX) protocol, based on the requirements document produced by the IPFIX Working Group. The protocols are characterized and grouped in broad categories, and evaluated against specific requirements. Finally, a recommendation is made to select the NetFlow v9 protocol as the basis for the IPFIX specification.

## Table of Contents

## 1. Introduction

The IP Flow Information Export (IPFIX) Working Group has been chartered to select a protocol for the export of flow information from traffic-observing devices (such as routers or dedicated probes). To this end, an evaluation team was formed to evaluate submitted protocols. Each protocol was represented by an advocate, who submitted a specific evaluation document for the respective protocol against the requirements document [1]. The specification of each protocol was itself available as one or several Internet-Drafts, sometimes referring normatively to documents from outside the IETF.

This document contains an evaluation of the submitted protocols with respect to the requirements document, and on a more general level, to the working group charter.

The following IPFIX candidate protocol submissions were evaluated:

- CRANE [7], [8]
- Diameter [9], [10]
- LFAP [11], [12], [13]
- NetFlow v9 [2], [15], [16]
- Streaming IPDR [17], [18]

This document uses terminology defined in [1] intermixed with that from submissions to explain the mapping between the two.

## 2. Protocol Summaries

In the following, each candidate protocol is described briefly, highlighting its specific distinguishing features.

## 2.1. CRANE

XACCT's Common Reliable Accounting for Network Element Protocol Version 1.0 [7][8] is described as a protocol for the transmission of accounting information from "Network Elements" to "mediation" and "business support systems".

### 2.1.1. CRANE Protocol Operation

The exporting side is the CRANE client, the collecting side is the CRANE server. Note that it is the server that is responsible for initiating the connection to the client. A client can have multiple simultaneous connections to different servers for robustness. Each server has an associated priority. A client only exports to the server with the highest priority that is perceived operational.

Clients and servers exchange messages over a reliable protocol such as TCP [3] or (preferably) the Stream Control Transmission Protocol (SCTP) [5]. The protocol uses application-layer acknowledgements as an indication of successful processing by the server. Strong authentication or data confidentiality aren't supported by the protocol, but can be supported by lower-layer mechanisms such as IPsec [20] or TLS [21].

The protocol is bidirectional over the entire duration of a session. There are 20 different message types. The protocol supports template negotiation, not only at startup but also later on in a session, as well as general status inquiries. There is a separate version negotiation protocol defined over UDP.

### 2.1.2. CRANE Data Encoding

Data encoding is based on templates. Templates contain "keys" representing items in data records. Clients (exporters) publish templates to servers (collectors). Servers can then select the subset of fields in a template that they are interested in. The client will suppress keys that haven't been selected by the server.

Data records contain references to template and configuration instances. They also carry sequence numbers (DSNs for Data Sequence Numbers). These sequence numbers can be used to de-duplicate data records that have been delivered multiple times during failover/fail-back in redundant configurations. A "duplicate" bit is set in these situations as a hint for the de-duplication process.

The encoding of (flow information) data records themselves is very compact. The client (exporter) can choose to send data in big-endian (network byte order) or little-endian format. There are eighteen fixed-size key types, as well as five variable-length string and binary data (BLOB) types.

## 2.2. Diameter

Diameter [9][10] is an evolution of the Remote Authentication Dial In User Service (RADIUS) protocol [22]. RADIUS is widely used to outsource authentication and authorization in dialup access environments. Diameter is a generalized and extensible protocol intended to support Authentication, Authorization and Accounting (AAA) requirements of different applications. Dialup and Mobile IPv4 are examples of such applications defined in the IETF.

### 2.2.1. Diameter Protocol Operation

Diameter is a peer-to-peer protocol. The base protocol defines fourteen command codes, organized as seven request/response command pairs. Presumably, only a subset of these would be used in a pure IPFIX application. Diameter includes capability negotiation and error notifications. Diameter operates over TCP or (preferred) SCTP. There is a framework for end-to-end security, the mechanisms for which are defined in a separate document. IPsec or TLS can be used to provide authentication or encryption at the underlying layers.

### 2.2.2. Diameter Data Encoding

Diameter conveys data in the form of attribute/value pairs (AVPs). An AVP consists of eight bytes of header plus the space to store the data, which depends on the data format. There are numerous predefined AVP data formats, including signed and unsigned integer types, each in 32 and 64 bit variants, IPv4 and IPv6 addresses, as well as others. The advocacy document [10] suggests that the predefined data formats IPFilterRule and/or QoSFilterRule could be extended to represent IP Flow Information. Such rules are represented as readable UTF-8 strings. Alternatively, new AVPs could be defined to represent flow information.

## 2.3. LFAP

LFAP [11][12][13] started out as the "Lightweight Flow Admission Protocol" and was used to outsource shortcut creation decisions on flow-based routers, as well as to provide per-flow statistics. Later versions removed the admission function and changed the name to "Lightweight Flow Accounting Protocol".

### 2.3.1. LFAP Protocol Operation

The exporter in LFAP is called the Connection Control Entity (CCE), and the collector is the Flow Accounting Server (FAS). These entities communicate with each other over a TCP connection. LFAP knows thirteen message types, including operations for connection management, version negotiation, flow information messages and administrative requests. Authentication and encryption can be provided by IPsec or TLS at lower layers. Additionally, the LFAP protocol itself supports four levels of security using HMAC-MD5 authentication and DES-CBC encryption. Note that DES is now widely regarded as not adequately secure, because its small key size makes brute-force attacks viable.

A distinguishing feature is that LFAP has two different message types for flow information: A Flow Accounting Request (FAR) message is sent when a new flow is identified at the CCE (meter/exporter). Accounting information is sent later in one or multiple Flow Update Notification (FUN) messages. A collector must match each FUN to a Flow ID previously sent in a FAR.

The LFAP document also defines a set of useful statistics about the accounting process. A separate MIB document [14] is provided for management of LFAP entities using SNMP.

### 2.3.2. LFAP Data Encoding

LFAP encodes data in a Type/Length/Value format with four bytes of overhead per data item (two bytes for the type and two bytes for the length field).

## 2.4. NetFlow v9

NetFlow v9 [2][15] is a generalized version of Cisco’s NetFlow protocol. Previous versions of NetFlow, in particular version 5, have been widely implemented and used for the exporting and collecting of IP flow information.

### 2.4.1. NetFlow Protocol Operation

NetFlow uses a very simple protocol, with the exporter sending template, options, and data "FlowSets" to the collector. FlowSets are sequences of data records of similar format. NetFlow is the only one of the candidate protocols that works over UDP [4]. Because of the simple unidirectional nature of the protocol, it should be relatively straightforward to add mappings to other transport protocols such as SCTP or TCP.

The use of SCTP to transport NetFlow v9 has been suggested in [16]. The suggested mapping describes how control and data can be mapped to different streams within a single SCTP connection, and suggests that the Partial Reliability extension [23] be used on data streams. In the proposed mapping, the exporter would initiate the connection.

### 2.4.2. NetFlow Data Encoding

NetFlow v9 uses a template facility to describe exported data. The data itself is represented in a compact way using network byte order.

## 2.5. Streaming IPDR

Streaming IPDR [17][18] is an application of the Network Data Management-Usage (NDM-U) for IP Services specification version 3.1 [19]. It has been developed by the Internet Protocol Detail Record Organization (IPDR, Inc. or ipdr.org). The terminology used is similar to CRANE's, talking about Service Elements (SEs), mediation systems and Business Support Systems (BSS).

### 2.5.1. Streaming IPDR Protocol Operation

Streaming IPDR operates over TCP. There is a "Trivial TCP Delivery" mode as well as an "Acknowledged TCP Delivery" or "Reliable Streaming" mode. The latter uses application-layer acknowledgements for increased reliability.

The protocol is basically unidirectional. The exporter opens a connection towards the collector, then sends a header followed by a set of record descriptors. Then it can send "Usage Event" records corresponding to these descriptors until the connection is terminated. New record descriptors can be sent at any time. Messages carry sequence numbers that are used for de-duplication during failover. They are also referenced by application-level acknowledgements when Reliable Streaming is used.

### 2.5.2. Streaming IPDR Data Encoding

IPDR uses an information modeling technique based on the XML-Schema language [24]. Data can be represented in XML or in a streamlined encoding based on the External Data Representation [25]. XDR forms the basis of Sun's Remote Procedure Call and Network File System protocols, and has proven to be both space- and processing-efficient.

# 3. Broad Classification of Candidate Protocols

In order to evaluate the candidate protocols against the higher-level requirements laid out in the IPFIX Working Group charter, it is useful to group them into broader categories.

## 3.1. Design Goals

One way to look at the candidate protocols is to study the goals that have directed their respective design. Note that the intention is not to exclude protocols that have been designed with a different class of applications in mind, but simply to better understand the different tradeoffs that distinguish the protocols.

### 3.1.1. High-Performance Flow Metering (NetFlow, LFAP)

Of the candidate protocols, Cisco's NetFlow is the purest example of a highly specialized protocol that has been designed with the sole objective of conveying accounting data from flow-aware routers at high rates. Starting from a fixed set of accounting fields, it has been extended a few times over the years to support additional fields and various types of aggregation in the metering/exporting process.

Riverstone's LFAP is similarly focused, except that it originated in a protocol to outsource the decision whether to create shortcuts in flow-based routers. This is still manifest in an increased emphasis on reliable operation, and in the split reporting of flow information using Flow Accounting Request (FAR) and Flow Update Notification (FUN) messages.

It has been pointed out that split reporting as done by LFAP can reduce memory requirements at the exporter. This concerns a subset of attributes that are neither "key" attributes which define flows, nor attributes such as packet or byte counters that must be updated for each packet anyway. On the other hand, when there are many short-lived flows, the number of flow export messages will be significantly higher than with "unitary" flow export models, and the collector will have to keep state about active flows until they are terminated.

### 3.1.2. Carrier-Grade Multi-Purpose Accounting (IPDR, CRANE)

Streaming IPDR and CRANE describe themselves as protocols to facilitate the reliable transfer of accounting information between Network Elements (or more generally "Service Elements" in the case of IPDR) and Mediation Systems or Business Support Systems (BSS). They

reflect a view of the accounting problem and of network system architectures that originates in traditional "vertically integrated" telecommunications.

Both protocols also emphasize extensibility with the goal of applicability to a wide range of accounting tasks.

IPDR is based on NDM-U, which uses the XML-Schema language for machine-readable specification of accounting data structures, while using the efficient XDR encoding for the actual data transfer.

CRANE uses templates to describe exported data. These templates are negotiated between collector and exporter and can change during a session.

#### 3.1.3. General-Purpose AAA (Diameter)

Diameter is another example of a broader-purpose protocol, in that it covers aspects of authentication and authorization as well as accounting. This explains its strong emphasis on security and reliability. The design also takes into account various types of intermediate agents.

### 3.2. Data Representation

IPFIX is intended to be deployed, among others, in high-speed routers and to be used for exporting detailed flow data at high flow rates. Therefore it is useful to look at the tradeoffs between the efficiency of data representation and the extensibility of data models. The two main efficiency goals should be (1) to minimize the export data rate and (2) to minimize data encoding overhead in the exporter. The overhead of decoding flow data at the collector is deemed less critical, and is partly covered by efficiency target (2), since an encoding that is easy on the encoder is often also easy on the decoder.

#### 3.2.1. Externally Described Encoding (CRANE, IPDR, NetFlow)

The protocols in this group use an external mechanism to fully describe the format in which flow data is encoded. The mechanisms are "templates" in the case of CRANE and NetFlow, and a subset of the XML-Schema language, or alternatively XDR IDL, for IPDR.

A fully external data format description allows for very compact encoding, with data components such as 32-bit integers taking up only four octets. The XDR representation used in IPDR additionally ensures that larger fields are always aligned on 32-bit boundaries, which can reduce processing requirements at both the exporter and the collector, at a slight cost of space (thus bandwidth) due to padding.

Most protocols specify "network byte order" or "big-endian" format in the export data format. CRANE is the only protocol where the exporter may choose the byte ordering. The principal benefit is that this lowers the processing demand on exporters based on little-endian architectures.

#### 3.2.2. Partly Self-describing Encoding (Diameter, LFAP)

Diameter and LFAP represent flow data using Type/Length/Value encodings. While this makes it possible to partly decode flow data without full context information - possibly useful for debugging - it does increase the encoding size and thus the bandwidth requirements both on the wire and in the exporter and collector.

LFAP has a "multi-record" encoding which claims to provide similar wire efficiency as the externally described encodings while still supporting diagnostic tools.

### 3.3. Protocol Flow

Another criterion for classification is the flow of protocol messages between exporter and collector.

#### 3.3.1. Mainly Unidirectional Protocols (IPDR, NetFlow)

In IPDR and NetFlow, the data flow is essentially from exporter to collector, with the collector only sending acknowledgements. The protocols send data descriptions (templates) on session establishment, and then start sending flow export data based on these templates. "Meta-information" about the operational status of the metering and exporting processes (for example about the sampling parameters in force at a given moment) is conveyed using a special type of "Option" template in NetFlow v9. IPDR currently doesn't have definitions for such "meta-data" types, but they could easily be defined outside the protocol proper.

### 3.3.2. Bidirectional Protocols (CRANE, LFAP)

CRANE allows for negotiation of the templates used for data export at the start of a session, and also allows negotiated template updates later on. CRANE sessions include an exporter and potentially several collectors, so these negotiations can involve more than two parties.

LFAP has an initial phase of version negotiation, followed by a phase of "data negotiation". After these startup phases, the exporter sends FAR and FUN messages to the collector. However, either party may also send Administrative Request (AR) messages to the other, and will normally receive Administrative Request Answers (ARA) in response. Administrative Requests can be used for status inquiries, including information about a specific active flow, or for negotiation of the "Information Elements" that the collector wants the exporter to export.

### 3.3.3. Unidirectional after Negotiation (Diameter)

Diameter has a general capabilities negotiation mechanism. The use of Diameter for IPFIX hasn't been described in sufficient detail to determine how capabilities negotiation would be used. After negotiation, the protocol would operate in essentially unidirectional mode, with Accounting-Request (ACR) messages flowing from the exporter to the collector, and Accounting-Answer (ACA) messages flowing back.

## 4. Item-Level Compliance Evaluation

The template for protocol advocates noted that not all requirements in [1] apply directly to the flow export protocol. In particular, sections 4 (Distinguishing Flows) and 5 (Metering Process) mainly specify requirements on the metering mechanism that "feeds" the exporter. However, in some cases they require information about the metering process to be reported to collectors, so the flow export protocol must support conveying this information.

### 4.1. Meter Reliability (5.1)

CRANE, Diameter, IPDR consider requirement 5.1 (reliability of the metering process or indication of "missing reliability") out of scope for the IPFIX protocol, which presumably means that they assume the metering process to be reliable.

The NetFlow v9 advocacy document takes a similar stance when it claims "Total Compliance. The metering process is reliable." (although this has been documented not to be true for all current Cisco implementations of NetFlow v5).

LFAP is the only protocol that explicitly addresses the possibility that data might be lost in the metering process, and provides useful statistics for the collectors to estimate, not just the amount of flow data that was lost, but also the amount of data that was not unaccounted for.

Note that in the general case, it can be considered unrealistic to assume total reliability of a flow-based metering process in all situations, unless sampling or coarse flow definitions are used. With the fine-grained flow classification mechanisms mandated by IPFIX, it is easy to imagine traffic where each - possibly very small - packet would create a new flow. This kind of traffic is in fact encountered in practice during aggressive port scans, and will eventually lead to table overflows or exceeding of memory bandwidth at the meter.

While some of these situations can be handled by dropping data later on in the exporter, data transfer, or collector, or by transitioning the meter to sampling mode (or increasing the sampling interval), it will sometimes be considered the lesser evil to simply report on the data that couldn't be accounted for. Currently LFAP is the only protocol that supports this.

## 4.2. Sampling (5.2)

CRANE and IPDR don't mention the possibility of sampling. This is natural because they are targeted towards telco-grade accounting, where sampling would be considered inadmissible. Since support for sampling is a "MAY" requirement, its lack could be tolerated, but severely restricts the applicability of these protocols in places of high aggregation, where absolute precision is not necessary. This includes applications such as traffic profiling, traffic engineering, and large-scale attack/intrusion detection, but also usage-based accounting applications where charging based on sampling is agreed upon.

The Diameter advocate acknowledges the existence of sampling and suggests to define new (grouped) AVPs to carry information about the sampling parameters in use.

LFAP does not currently support sampling, although its advocate contends that adding support for this would be relatively straightforward, without going into too much detail.

NetFlow v9 does support sampling (and many implementations and deployments of sampled NetFlow exist for previous NetFlow versions). Option Data is supposed to convey sampling configuration, although no sampling-related field types have yet been defined in the document.

### 4.3. Overload Behavior (5.3)

The requirements document suggests that meters adapt to overload situations, for example by changing to sampling (or reducing the sampling rate if sampling is already in effect), by changing the flow definition to coarser flow categories (thinning), by stopping to meter, or by reducing packet processing.

In these situations, the requirements document mandates that flow information from before the modification of metering behavior can be cleanly distinguished from flow information from after the modification. For the suggested mitigation methods of sampling or thinning, this essentially means that all existing flows have to be expired, and an entirely new set of flows must be started. This is undesirable because it causes a peak of resource usage in an already overloaded situation.

LFAP and NetFlow claim to handle this requirement, both by supporting only the simple overload mitigation methods that don't require the entire set of existing flows to be expired. The NetFlow advocate claims that the reporting requirement could be easily met by expiring existing flows with the old template, while sending a new template for new flows. While it is true that NetFlow handles this requirement in a very graceful manner, the general performance issue remains.

CRANE, Diameter, and IPDR consider the requirement out of scope for the protocol, although Diameter summarily acknowledges the possible need for new AVP definitions related to mitigation methods.

### 4.4. Timestamps (5.4)

All protocols support reporting of timestamps with the required (one centisecond) or better precision.

### 4.5. Time Synchronization (5.5)

While all other protocols have timestamp types that are relative to a well-known reference time, timestamps in NetFlow are reported relative to the sysUpTime of the exporting device. For applications that require the absolute start/end times of flows, this means that exporter sysUpTime has to be matched with absolute time. Although every NetFlow export packet header contains a "UNIX Secs" field, it cannot be used for UTC synchronization without loss of precision, because this field only has 1-second resolution.

### 4.6. Flow Expiration (5.6)

As currently specified, this requirement concerns the metering process only and has no bearing on the export protocol.

If it is desired to export the reason for flow expiration (e.g., inactivity timeout, active flow timeout, expiration to reclaim resources, or observation of a flow termination indication such as a TCP FIN segment), then none of the protocols currently supports this, although each could be extended to do so.

### 4.7. Ignore Port Copy (5.9)

This requirement only concerns the metering process and has no bearing on the export protocol.

### 4.8. Information Model (6.1)

All candidate protocols have information models that can represent all required and all optional attributes. The Diameter contribution lacks some detail on how exactly the IPFIX-specific attributes should be mapped.

### 4.9. Data Model (6.2)

#### 4.9.1. Data Model Extensibility

Each candidate protocol defines a data model that allows for some degree of extensibility.

CRANE uses Keys to specify fields in templates. A key "specification MUST consist of the description and the data type of the accounting item." Apparently extensibility is intended, but it is not clear whether adding a new Key really only involves writing a textual description and deciding upon a base type. Every Key also has a 32-bit Key ID, but from the current specification they don’t seem to carry global semantics.

Diameter’s Attribute/Value Pairs (AVP) have a 32-bit identifier (AVP Code) administered by IANA. In addition, there is an optional 32-bit Vendor-ID that can contain an SMI Enterprise Number for vendor-defined attributes. If the Vendor-ID (and a corresponding flag in the attribute) is set, the AVP Code becomes local to that vendor.

IPDR uses a subset of the XML-Schema language for extensibility, thus allowing for vendor- and application-specific extensions of the data model.

In LFAP, flow attributes are defined as Information Elements. There is a 16-bit IE type code (which is carried in the export protocol for every IE). One type code is reserved for vendor-specific extensions. Arbitrary sub-types of the vendor-specific IE can be defined using ASN.1 Object IDs (OIDs).

In NetFlow v9 as reviewed, data items are identified by a sixteen-bit field type. 26 field types are defined in the document. The document suggests to look check a Web page at Cisco Systems' site for the current list of field types. It would be preferable if the administration of the field type space would be delegated to IANA.

#### 4.9.2. Flexible Flow Record Definition

All protocols allow for flexible flow record definitions. CRANE and LFAP make the selection/negotiation of the attributes to be included in flow records a part of the protocol, the other protocols leave this to outside configuration mechanisms.

### 4.10. Data Transfer (6.3)

#### 4.10.1. Congestion Awareness (6.3.1)

All protocols except for NetFlow v9 operate over a single TCP or SCTP transport connection, and inherit the congestion-friendliness of these protocols.

NetFlow v9 was initially defined to operate over UDP, but specified in a transport-independent manner. Recently, a document [16] has been issued that describes how NetFlow v9 can be run over SCTP with the proposed Partial Reliability extension. This transport mapping would fill the congestion awareness requirement.

#### 4.10.2. Reliability (6.3.2)

The requirements in the area of reliability are specified as follows: If flow records can be lost during transfer, this must be indicated to the collector in a way that permits the number of lost records to be gauged; and the protocol must be open to reliability extensions including retransmission of lost flow records, detection of exporter/collector disconnection and fail-over, and acknowledgement of flow records by the collecting process (application-level acknowledgements).

Here are a few observations regarding the candidate protocols' approaches to reliability. Note that the requirement for multiple collectors (8.3) also touches on the issue of reliability.

CRANE, Diameter, and IPDR, as protocols that strive to be carrier-grade accounting protocols, understandably exhibit a strong emphasis on near-total reliability of the flow export process. All three protocols use application-level acknowledgements (in case of IPDR, optionally) to include the entire collection process in the feedback loop. Indications of "lack of reliability" (lost flow data) are somewhat unnatural to these protocols, because they take every effort to never lose anything. These protocols seem suitable in situations where one would rather drop a packet than forward it unaccounted for.

LFAP has application-level acknowledgements, and it also reports detailed statistics about lost flows and the amount of data that couldn't be accounted for. It represents a middle ground in that it acknowledges that accounting reliability will sometimes be sacrificed for the benefit of other tasks, such as switching packets, and provides the tools to gracefully deal with such situations.

NetFlow v9 is the only protocol for which the use of a "reliable" transport protocol is optional, and the only protocol that doesn't support application-level acknowledgements. In all fairness, it should be noted that it is a very simple and efficient protocol, so in an actual deployment it might exhibit a higher level of reliability than some of the other protocols given the same amount of resources.

#### 4.10.3. Security (6.3.3)

##### 4.10.3.1. IPsec and TLS

All protocols can use, and their descriptions in fact recommend them to use, lower-layer security mechanisms such as IPsec and, with the exception of NetFlow v9 over UDP, TLS. It can be argued that in all envisioned usage scenarios for IPFIX, both IPsec and TLS provide sufficient protection against the main identified threats of flow data disclosure and forgery.

The Diameter document is the only protocol definition that goes into sufficient level of detail with respect to the application of these mechanisms, in particular the negotiation of certificates and ciphers in TLS, and the use of IKE [6] for IPsec. Diameter also mandates that either IPsec or TLS be used.

##### 4.10.3.2. Application-level Security

Diameter suggests an additional end-to-end security framework for dealing with untrusted third-party agents. I am not entirely convinced that this additional level of security justifies the additional complexity in the context of IPFIX.

LFAP [11] is the only other protocol that includes some higher-level security mechanisms, providing four levels of security including no security, authenticated peers, flow data authentication, and flow data encryption using HMAC-MD5-96 and DES-CBC.

As far as the author can judge (not being a security expert), LFAP's built-in support for authentication and encryption doesn't provide significant additional security compared with the use of TLS or IPsec. It is potentially useful in situations where TLS or IPsec are unavailable for some reason, although in the context of IPFIX scenarios, it should be possible to assume support for these lower-layer mechanisms if the participating devices are capable of the necessary cryptographic methods at all.

#### 4.10.4. Push and Pull Mode Reporting (6.4)

All protocols support the mandatory "push" mode.

The optional "pull" mode could be supported relatively easily in Diameter, and is foreseen in NDM-U, the basis of the Streaming IPDR proposal. CRANE, LFAP and NetFlow don't have a "pull" mode. For CRANE and LFAP, adding one would not violate the spirit of the protocols because they are already two-way, and in fact LFAP already foresees inquiries about specific active flows using Administrative Request (AR) messages with a RETURN_INDICATED_FLOWS Command Code IE.

#### 4.10.5. Regular Reporting Interval (6.5)

As stated, this requirement concerns the metering process only and has no bearing on the export protocol.

#### 4.10.6. Notification on Specific Events (6.6)

The specific events listed in the requirements documents as examples for "specific events" are "the arrival of the first packet of a new flow and the termination of a flow after flow timeout". For the former, only LFAP explicitly generates messages upon creation of a new flow. NetFlow always exported flow information on expiration of flows, either due to timeout or due to an indication of flow termination. The other protocols are unspecific about when flow information is exported.

On "specific events" in general, all protocols have some mechanism that could be used for notification of asynchronous events. An example for such an event would be that the sampling rate of the meter was changed in response to a change in the load on the exporting process.

CRANE has Status Request/Status Response messages, but as defined, Status Requests can only be issued by the server (collector), so they cannot be used by the server to signal asynchronous events. As in IPDR, this could be circumvented by defining templates for meta-information.

Diameter could use special Accounting-Request messages for event notification.

IPDR would presumably define pseudo-"Usage Events" using an XML Schema so that events can be reported along with usage data.

LFAP has Administrative Requests (AR) that can be initiated from either side. The currently defined ARs are all information inquiries or reconfiguration requests, but new ARs could be defined to provide unsolicited information about specific asynchronous events. The LFAP MIB also defines some traps/notifications. SNMP notifications are useful to signal events to a network management system, but they are less attractive as a mechanism to signal events that should be somehow handled by a collector.

In NetFlow v9, Option Data FlowSets are defined to convey information about the metering and export processes. The current document specifies that Option Data should be exported periodically, although this requirement will be relaxed for asynchronous events. It should be noted that periodical export of option flowsets (and also of templates) may have been considered necessary because NetFlow can run over an unreliable transport; it seems less natural when a reliable transport such as TCP is used.

#### 4.10.7. Anonymization (6.7)

None of the protocols include explicit support for anonymization. All protocols could be extended to convey when and how anonymization is being performed by an exporter, using mechanisms similar to those that would be used to report on sampling.

#### 4.10.8. Several Collecting Processes (8.3)

CRANE, Diameter, and IPDR all support multiple collectors in a backup configuration. The failover case is analyzed in some detail, with support for data buffering and de-duplication in failover situations.

NetFlow takes a more simple-minded approach in that it allows multiple (currently: two) collectors to be configured in an exporter. Both collectors will generally receive all data and could use sequence numbers and inter-collector communication to de-duplicate them. This is a simple way to improve availability but may also be

considered to be wasteful, both in terms of bandwidth and in terms of other exporter resources. With the current UDP mapping it is easy enough to send multiple copies of datagrams to different collectors, but when SCTP or TCP is used, sending all data over multiple connections will exacerbate performance issues.

Failover in LFAP must take into account that flow information is split into FARs and FUNs. When a (primary) FAS A fails, a secondary FAS B will receive FUNs for flows whose FARs had only been sent to A. If such FUNs are to be handled correctly in the failover case, then either the set of active flows must be kept in sync between the primary and backup FASs, or the exporting CCE must have a way to generate new FARs on failover.

## 5. Conclusions

Every candidate protocol has its strengths and weaknesses. If the primary goal of the IPFIX standardization effort were to define a carrier-grade accounting protocol that can also be used to carry IP flow information, then one of CRANE, Diameter and Streaming IPDR would probably be the candidate of choice.

But since the goal is to standardize existing practice in the area of IP Flow Information Export, it makes sense to analyze why previous versions of NetFlow have been so widely implemented and used. The strong position of Cisco in the router market certainly played a major role, but we should not underestimate the value of having a simple and streamlined protocol that "does one thing and does it well". It has been extremely easy to write NetFlow collecting processes, as all the protocol demands from a collector is to sit there and receive data. This model is no longer adequate when one wants to support increased levels of reliability or dynamically changing semantics for data export. But NetFlow remains a simple protocol, mainly by leaving out issues of configuration/negotiation.

So far, the biggest issue with NetFlow is that it could not resolve itself to mandate a reliable (and congestion-friendly) transport. This could easily be fixed, and bring with it some additional possibilities for simplifications. For example it would no longer be necessary to periodically retransmit Template FlowSets, and Option Data FlowSets could become a more versatile way of reporting meta-information about the metering and exporting processes either synchronously or asynchronously. Application-level acknowledgements - possibly as an option - would be a low-impact addition to improve overall reliability.

LFAP is also relatively focused on flow information export, but carries around too much baggage from its youth as the Lightweight Flow Admission Protocol. The bidirectional nature and large number of message types in the protocol are one symptom of this, the separation of flow information into FARs and FUNs - which must be matched at the collector - are another. Data encoding is less space-efficient than that of CRANE, NetFlow or IPDR, and will present a performance issue at high flow rates.

LFAP's indications of unaccounted data and its MIB are excellent features that would be very useful in many operational situations.

## 5.1. Recommendation

It is the opinion of the evaluation team that the goals of the IPFIX WG charter would best be served by starting with NetFlow v9, working on lacking mechanisms in the areas of transport, security, reliability, and redundant configurations, and doing so very carefully in order to retain as much simplicity as possible and to avoid overloading the protocol. By starting from the simplest protocol that meets a large percentage of the specific requirements, we can hope to arrive at a protocol that meets all requirements and still allows widespread and cost-effective implementation.

As evaluated, NetFlow v9 doesn't specify any security mechanisms. The IPFIX protocol specification must specify how the security requirements in section 6.3.3 of [1] can be assured. The IPFIX specification must be specific about the choice of security-supporting protocol(s) and about all relevant issues such as security negotiation, protocol modes permitted, and key management.

The other important requirement that isn't fulfilled by NetFlow v9 today is support for a congestion-aware protocol (see section 6.3.1 of [1]). So a mapping to a known congestion-friendly protocol such as TCP, or, as suggested in [16], (PR-)SCTP, is considered as another necessary step in the preparation of the IPFIX specification.

# 6. Security Considerations

The security mechanisms of the candidate protocols were discussed in Section 4.10.3.

# 7. Acknowledgements

Many of the issues have been discussed with the other members of the IPFIX evaluation team: Juergen Quittek, Mark Fullmer, Ram Gopal, and Reinaldo Penno. Many participants on the ipfix mailing list provided valuable feedback, including Vamsidhar Valluri, Paul Calato, Tal

Givoly, Jeff Meyer, Robert Lowe, Benoit Claise, and Carter Bullard. Bert Wijnen, Steve Bellovin, Russ Housley, and Allison Mankin provided valuable feedback during AD and IESG review.

## 8. References

### 8.1. Normative References

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

[2] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.

[3] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[4] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

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

[6] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

### 8.2. Informative References

[7] Zhang, K. and E. Elkin, "XACCT's Common Reliable Accounting for Network Element (CRANE) Protocol Specification Version 1.0", RFC 3423, November 2002.

[8] Zhang, K., "Evaluation of the CRANE Protocol Against IPFIX Requirements", Work in Progress, September 2002.

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

[10] Zander, S., "Evaluation of Diameter Protocol against IPFIX Requirements", Work in Progress, September 2002.

[11] Calato, P. and M. MacFaden, "Light-weight Flow Accounting Protocol Specification Version 5.0", July 2002.

[12] Calato, P. and M. MacFaden, "Light-weight Flow Accounting Protocol Data Definition Specification Version 5.0", July 2002.

[13] Calato, P., "Evaluation Of Protocol LFAP Against IPFIX Requirements", Work in Progress, September 2002.

[14] Calato, P. and M. MacFaden, "Light-weight Flow Accounting Protocol MIB", Work in Progress, September 2002.

[15] Claise, B., "Evaluation Of NetFlow Version 9 Against IPFIX Requirements", Work in Progress, September 2002.

[16] Djernaes, M., "Cisco Systems NetFlow Services Export Version 9 Transport", Work in Progress, February 2003.

[17] Meyer, J., "Reliable Streaming Internet Protocol Detail Records", Work in Progress, August 2002.

[18] Meyer, J., "Evaluation Of Streaming IPDR Against IPFIX Requirements", Work in Progress, September 2002.

[19] Internet Protocol Detail Record Organization, "Network Data Management - Usage (NDM-U) For IP-Based Services Version 3.1", April 2002. URL: http://www.ipdr.org/documents/NDM-U_3.1.pdf

[20] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[21] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[22] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

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

[24] DeRose, S., Maler, E. and D. Orchard, "XML 1.0 Recommendation", W3C FirstEdition REC-xml-19980210, February 1998.

[25] Srinivasan, R., "XDR: External Data Representation Standard", RFC 1832, August 1995.

[26] <http://www.nmops.org/>

[27] <http://www.ipdr.org/>

## Appendix A. A Note on References to the Candidate Protocol Documents

At the time of the evaluation, the candidate protocol definitions, as well as their respective accompanying advocacy documents, were available as Internet-Drafts. As of the time of publication of this document, some of the protocols have been published as RFCs, others are still being revised as Internet-Drafts, and some will have expired. This document attempts to extract the relevant information from the individual protocol definitions and, in the context of the IPFIX requirements, provide a meaningful comparison between them.

Since this evaluation proposes to use NetFlow v9 as the basis for the IPFIX protocol, only the reference to this protocol is considered "normative", although strictly spoken, the present document doesn't define any protocol, and the selected protocol will have to be further refined to become the IPFIX protocol.

In the interest of stable references, the bibliography points to RFCs where those have become available (for DIAMETER and CRANE). Other protocols are still available only as Internet-Drafts and may eventually expire. The LFAP drafts - which already have expired - are still available from the www.nmops.org Web site [26] (as well as other places). The IPDR documents are available on the IPDR Web site [27].

## Author's Address

Simon Leinen
SWITCH
Limmatquai 138
P.O. Box
CH-8021 Zurich
Switzerland

Phone: +41 1 268 1536
EMail: simon@switch.ch

## Full Copyright Statement



## Intellectual Property



## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.