

INTERNET-DRAFT  
[draft-ietf-ipfix-reqs-16.txt](#)  
Category: Informational  
Expires: December 2004

J. Quittek  
NEC Europe Ltd.  
T. Zseby  
Fraunhofer FOKUS  
B. Claise  
Cisco Systems  
S. Zander  
Fraunhofer FOKUS

June 2004

## Requirements for IP Flow Information Export

<[draft-ietf-ipfix-reqs-16.txt](#)>

### Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

### Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

### Abstract

This memo defines requirements for the export of measured IP flow information out of routers, traffic measurement probes and middleboxes.

## Table of Contents

<a href="#">1. Introduction</a>	<a href="#">3</a>
<a href="#">2. Terminology</a>	<a href="#">4</a>
<a href="#">2.1. IP Traffic Flow</a>	<a href="#">4</a>
<a href="#">2.2. Observation Point</a>	<a href="#">4</a>
<a href="#">2.3. Metering Process</a>	<a href="#">5</a>
<a href="#">2.4. Flow Record</a>	<a href="#">5</a>
<a href="#">2.5. Exporting Process</a>	<a href="#">6</a>
<a href="#">2.6. Collecting Process</a>	<a href="#">6</a>
<a href="#">3. Applications Requiring IP Flow Information Export</a>	<a href="#">6</a>
<a href="#">3.1. Usage-based Accounting</a>	<a href="#">6</a>
<a href="#">3.2. Traffic Profiling</a>	<a href="#">7</a>
<a href="#">3.3. Traffic Engineering</a>	<a href="#">7</a>
<a href="#">3.4. Attack/Intrusion Detection</a>	<a href="#">7</a>
<a href="#">3.5. QoS Monitoring</a>	<a href="#">8</a>
<a href="#">4. Distinguishing Flows</a>	<a href="#">9</a>
<a href="#">4.1. Encryption</a>	<a href="#">9</a>
<a href="#">4.2. Interfaces</a>	<a href="#">9</a>
<a href="#">4.3. IP Header Fields</a>	<a href="#">10</a>
<a href="#">4.4. Transport Header Fields</a>	<a href="#">10</a>
<a href="#">4.5. MPLS Label</a>	<a href="#">10</a>
<a href="#">4.6. DiffServ Code Point</a>	<a href="#">10</a>
<a href="#">5. Metering Process</a>	<a href="#">10</a>
<a href="#">5.1. Reliability</a>	<a href="#">11</a>
<a href="#">5.2. Sampling</a>	<a href="#">11</a>
<a href="#">5.3. Overload Behavior</a>	<a href="#">11</a>
<a href="#">5.4. Timestamps</a>	<a href="#">12</a>
<a href="#">5.5. Time Synchronization</a>	<a href="#">12</a>
<a href="#">5.6. Flow Expiration</a>	<a href="#">13</a>
<a href="#">5.7. Multicast Flows</a>	<a href="#">13</a>
<a href="#">5.8. Packet Fragmentation</a>	<a href="#">13</a>
<a href="#">5.9. Ignore Port Copy</a>	<a href="#">13</a>
<a href="#">6. Data Export</a>	<a href="#">13</a>
<a href="#">6.1. Information Model</a>	<a href="#">14</a>
<a href="#">6.2. Data Model</a>	<a href="#">16</a>
<a href="#">6.3. Data Transfer</a>	<a href="#">16</a>
<a href="#">6.3.1. Congestion Awareness</a>	<a href="#">16</a>
<a href="#">6.3.2. Reliability</a>	<a href="#">16</a>
<a href="#">6.3.3. Security</a>	<a href="#">17</a>
<a href="#">6.4. Push and Pull Mode Reporting</a>	<a href="#">17</a>
<a href="#">6.5. Regular Reporting Interval</a>	<a href="#">18</a>
<a href="#">6.6. Notification on Specific Events</a>	<a href="#">18</a>
<a href="#">6.7. Anonymization</a>	<a href="#">18</a>
<a href="#">7. Configuration</a>	<a href="#">18</a>
<a href="#">7.1. Configuration of the Metering Process</a>	<a href="#">19</a>
<a href="#">7.2. Configuration of the Exporting Process</a>	<a href="#">19</a>
<a href="#">8. General Requirements</a>	<a href="#">19</a>
<a href="#">8.1. Openness</a>	<a href="#">19</a>
<a href="#">8.2. Scalability</a>	<a href="#">20</a>



<a href="#">8.3. Several Collecting Processes .....</a>	<a href="#">20</a>
<a href="#">9. Special Device Considerations .....</a>	<a href="#">20</a>
<a href="#">10. Security Considerations .....</a>	<a href="#">22</a>
<a href="#">10.1. Disclosure of Flow Information Data .....</a>	<a href="#">23</a>
<a href="#">10.2. Forgery of Flow Records .....</a>	<a href="#">23</a>
<a href="#">10.3. Denial of Service (DoS) Attacks .....</a>	<a href="#">24</a>
<a href="#">11. Acknowledgments .....</a>	<a href="#">24</a>
<a href="#">12. Appendix: Derivation of Requirements from Applications .....</a>	<a href="#">25</a>
<a href="#">13. Normative References .....</a>	<a href="#">30</a>
<a href="#">14. Informative References .....</a>	<a href="#">30</a>
<a href="#">15. Authors' Addresses .....</a>	<a href="#">31</a>
<a href="#">16. IPR Notices .....</a>	<a href="#">32</a>
<a href="#">17. Full Copyright Statement .....</a>	<a href="#">32</a>

## **[1. Introduction](#)**

There are several applications that require flow-based IP traffic measurements. Such measurements could be performed by a router while forwarding the traffic, by a middlebox [[RFC3234](#)], or by a traffic measurement probe attached to a line or a monitored port. This memo defines requirements for exporting traffic flow information out of these boxes for further processing by applications located on other devices. They serve as input to the standardization of an IPFIX protocol.

In [section 3](#), a selection of such applications is presented. The following sections list requirements derived from these applications.

In its early discussions the IPFIX Working Group chose to evaluate existing flow export protocols at the same time as it was developing this 'requirements' document.

Flow export, however, is not performed by a protocol acting alone, it also requires a system of co-operating processes. In producing IPFIX requirements, therefore, the Working Group decided to specify what was required by these various processes - the metering process, the exporting process, etc. In these specifications we use lower-case for the words must, may and should, to indicate that IPFIX implementors have some freedom as to how to meet the requirements.

The Working Group's goal is to produce standards-track documents describing the IPFIX architecture, i.e. information model and flow export protocol RFCs. As well as meeting the requirements set out in this document, the information model and protocol documents will provide a full specification of the IPFIX system, and will use uppercase [[RFC 2119](#)] keywords as is usual in standards-track RFCs.



## **2. Terminology**

The following terminology is used in this document:

### **2.1. IP Traffic Flow**

There are several definitions of the term 'flow' being used by the Internet community. Within this document we use the following one:

A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. Each property is defined as the result of applying a function to the values of:

1. one or more packet header field (e.g. destination IP address), transport header field (e.g. destination port number), or application header field (e.g. RTP header fields [[RFC1889](#)])
2. one or more characteristics of the packet itself (e.g. number of MPLS labels, etc...)
3. one or more of fields derived from packet treatment (e.g. next hop IP address, the output interface, etc...)

A packet is defined to belong to a flow if it completely satisfies all the defined properties of the flow.

This definition covers the range from a flow containing all packets observed at a network interface to a flow consisting of just a single packet between two applications with a specific sequence number. Please note that the flow definition does not necessarily match a general application-level end-to-end stream. However, an application may derive properties of application-level streams by processing measured flow data. Also, please note that although packet properties may depend on application headers, there is no requirement defined in this document related to application headers.

### **2.2. Observation Point**

The observation point is a location in the network where IP packets can be observed. Examples are a line to which a probe is attached, a shared medium such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router.

Note that one observation point may be a superset of several other observation points. For example one observation point can be an entire line card. This would be the superset of the individual observation points at the line card's interfaces.



### 2.3. Metering Process

The metering process generates flow records. Input to the process are packet headers observed at an observation point and packet treatment at the observation point, for example the selected output interface. The metering process consists of a set of functions that includes packet header capturing, timestamping, sampling, classifying, and maintaining flow records.

The maintenance of flow records may include creating new records, updating existing ones, computing flow statistics, deriving further flow properties, detecting flow expiration, passing flow records to the exporting process, and deleting flow records.

The sampling function and the classifying function may be applied more than once with different parameters. Figure 1 shows the sequence in which the functions are applied. Sampling is not illustrated in the figure; it may be applied before any other function.

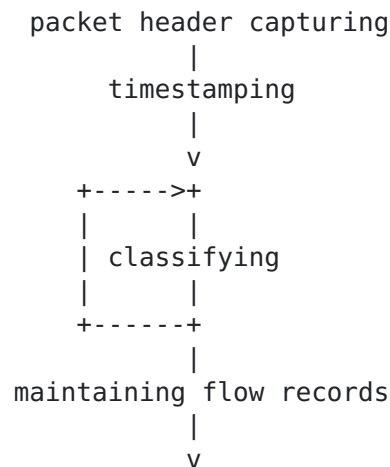


Figure 1: Functions of the metering process

### 2.4. Flow Record

A flow record contains information about a specific flow that was metered at an observation point. A flow record contains measured properties of the flow (e.g. the total number of bytes of all packets of the flow) and usually also characteristic properties of the flow (e.g. source IP address).





## **2.5. Exporting Process**

The exporting process sends flow records to one or more collecting processes. The flow records are generated by one or more metering processes.

## **2.6. Collecting Process**

The collecting process receives flow records from one or more exporting processes. The collecting process might store received flow records or further process them, but these actions are out of the scope of this document.

# **3. Applications Requiring IP Flow Information Export**

This section describes a selection of applications requiring IP flow information export. Because requirements for flow export listed in further sections below are derived from these applications, their selection is crucial. The goal of this requirements document is not to cover all possible applications with all their flow export requirements, but to cover applications which are considered to be of significant importance in today's and/or future IP networks, and for which requirements can be met with reasonable technical effort.

The list of applications should lead to a better understanding of the requirements which is particularly important when designing or implementing traffic flow metering functions. A detailed overview of which requirement was derived from which application(s) is given in the appendix.

Please note that the described applications can have a large number of differing implementations. Requirement details or requirement significance (required (must), recommended (should), optional (may)) could differ for specific implementations and/or for specific application scenarios. Therefore we derive the requirements from the general functionality of the selected applications. Some particular cases will even mandate more stringent requirements than the ones defined in this document. For example, usage-based accounting is certainly the application that will probably mandate the highest degree of reliability amongst the applications discussed below. The reliability requirements defined in sections [5.1](#) and [6.3.2](#). are not sufficient to guarantee the level of reliability that is needed for many usage-based accounting systems. Particular reliability requirements for accounting systems are discussed in [[RFC2975](#)].

## **3.1. Usage-based Accounting**

Several new business models for selling IP services and IP-based services are currently under investigation. Beyond flat rate



services which do not need accounting, accounting can be based on time or volume. Accounting data can serve as input for billing systems. Accounting can be performed per user or per user group, it can be performed just for basic IP service or individually per high-level service and/or per content type delivered. For advanced/future services, accounting may also be performed per class of service, per application, per time of day, per (label switched) path used, etc.

### **3.2. Traffic Profiling**

Traffic profiling is the process of characterizing IP flows by using a model that represents key parameters of the flows such as flow duration, volume, time and burstiness. It is a prerequisite for network planning, network dimensioning, trend analysis, business model development, and other activities. It depends heavily on the particular traffic profiling objective(s), which statistics, and which accuracy are required from the measurements. Typical information needed for traffic profiling is the distribution of used services and protocols in the network, the amount of packets of a specific type (e.g. percentage of IPv6 packets) and specific flow profiles.

Since objectives for traffic profiling can vary, this application requires a high flexibility of the measurement infrastructure, especially regarding the options for measurement configuration and packet classification.

### **3.3. Traffic Engineering**

Traffic Engineering (TE) comprises methods for measurement, modeling, characterization and control of a network. The goal of TE is the optimization of network resource utilization and traffic performance [[RFC2702](#)]. Since control and administrative reaction to measurement results requires access to the involved network nodes, TE mechanisms and the required measurement function usually are performed within one administrative domain. Typical parameters required for TE are link utilization, load between specific network nodes, number, size and entry/exit points of the active flows and routing information.

### **3.4. Attack/Intrusion Detection**

Capturing flow information plays an important role for network security, both for detection of security violation, and for subsequent defense. In case of a Denial of Service (DOS) attack, flow monitoring can allow detection of unusual situations or suspicious flows. In a second step, flow analysis can be performed in order to gather information about the attacking flows, and for deriving a defense strategy.

Intrusion detection is a potentially more demanding application which



would not only look at specific characteristics of flows, but may also use a stateful packet flow analysis for detecting specific, suspicious activities, or unusually frequent activities. Such activities may be characterized by specific communication patterns, detectable by characteristic sequences of certain packet types.

### **3.5. QoS Monitoring**

QoS monitoring is the passive measurement of quality parameters for IP flows. In contrast to active measurements, passive measurements utilize the existing traffic in the network for QoS analysis. Since no test traffic is sent, passive measurements can only be applied in situations where the traffic of interest is already present in the network. One example application is the validation of QoS parameters negotiated in a service level specification. Note that passive/active measurement is also referred to as non-intrusive/intrusive measurement or as measurement of observed/synthetic traffic.

Passive measurements cannot provide the kind of controllable experiments that can be achieved with active measurements. On the other hand passive measurements do not suffer from undesired side effects caused by sending test traffic (e.g. additional load, potential differences in treatment of test traffic and real customer traffic).

QoS monitoring often requires the correlation of data from multiple observation points (e.g. for measuring one-way metrics). This requires proper clock synchronization of the involved metering processes. For some measurements, flow records and/or notifications on specific events at the different observation points must be correlated, for example the arrival of a certain packet. For this, the provisioning of post-processing functions (e.g. the generation of packet IDs) at the metering processes would be useful. Since QoS monitoring can lead to a huge amount of measurement result data, it would highly benefit from mechanisms to reduce the measurement data, like aggregation of results and sampling.

Please note that not all requirements for QoS monitoring are covered by the IPFIX requirements specified in the following sections. The IPFIX requirements are targeted at per flow information including summaries of per-packet properties for packets within a flow, but not per-packet information itself. For example jitter measurement requires timestamping each packet and reporting of all timestamps of a flow, but the IPFIX requirements only cover timestamps of first and last packet of a flow.



## **4. Distinguishing Flows**

Packets are mapped to flows by evaluating their properties. Packets with common properties are considered to belong to the same flow. A packet showing at least one difference in the set of properties is considered to belong to a different flow.

The following subsections list a set of properties which a metering process must, should, or may be able to evaluate for mapping packets to flows. Please note that requiring the ability to evaluate a certain property does not imply that this property must be evaluated for each packet. In other words, meeting the IPFIX requirements means that the metering process in general must be able, via its configuration, to somehow support to distinguish flows via all the must fields, even if in certain circumstances/for certain applications, only a subset of the must fields is needed and effectively used to distinguish flows.

Which combination of properties is used for distinguishing flows and how these properties are evaluated depends on the configuration of the metering process. The configured choice of evaluated properties strongly depends on the environment and purpose of the measurement and on the information required by the collecting process. But in any case, it must be ensured that a collecting process is able to clearly identify for each received flow record which set of properties was used for distinguishing this flow from other ones.

For specific deployments, only a subset of the required properties listed below can be used to distinguish flows, for example in order to aggregate the flow records and reduce the number of flow records exported. On the other hand, some other deployments will require distinguishing flows by some extra parameters, such as the TTL field of the IP header or the BGP Autonomous System number [[RFC1771](#)] of the IP destination address.

### **4.1. Encryption**

If encryption is used, the metering process might not be able to access all header fields. A metering process must meet the requirements stated in this [section 4](#) only for packets that have the relevant header fields not encrypted.

### **4.2. Interfaces**

The metering process must be able to separate flows by the incoming interface or by the outgoing interface or by both of them.





### **4.3. IP Header Fields**

The metering process must be able to separate flows by the following fields of the IP header:

1. source IP address
2. destination IP address
3. protocol type (TCP, UDP, ICMP, ...)

For source address and destination address, separating by full match must be supported as well as separation by prefix match.

The metering process should be able to separate flows by the IP version number if the observation point is located at a device that is supporting more than one IP version.

### **4.4. Transport Header Fields**

The metering process must be able to separate flows by the port numbers of the transport header in case of TCP or UDP being used as transport protocol. The metering process should be able to separate flows by the port numbers of the transport header in case of SCTP [[RFC2960](#)].

For separation, both, source and destination port number must be supported for distinguishing flows, individually as well as in combination.

### **4.5. MPLS Label**

If the observation point is located at a device supporting Multiprotocol Label Switching (MPLS, see [[RFC3031](#)]) then the metering process must be able to separate flows by the MPLS label.

### **4.6. DiffServ Code Point**

If the observation point is located at a device supporting Differentiated Services (DiffServ) then the metering process must be able to separate flows by the DiffServ Code Point (DSCP, see [[RFC2474](#)]).

## **5. Metering Process**

The following are requirements for the metering process. All measurements must be conducted from the point of view of the observation point.



### **5.1. Reliability**

The metering process must either be reliable or the absence of reliability must be known and indicated. The metering process is reliable if each packet passing the observation point is metered according to the configuration of the metering process. If, e.g. due to some overload, not all passing packets can be included into the metering process, then the metering process must be able to detect this failure and to report it.

### **5.2. Sampling**

Sampling describes the systematic or random selection of a subset of elements (the sample) out of a set of elements (the parent population). Usually the purpose of applying sampling techniques is to estimate a parameter of the parent population by using only the elements of the subset. Sampling techniques can be applied for instance to select a subset of packets out of all packets of a flow or to select a subset of flows out of all flows on a link. Sampling methods differ in their sampling strategy (e.g. systematic or random) and in the event that triggers the selection of an element. The selection of one packet can for instance be triggered by its arrival time (time-based sampling), by its position in the flow (count-based sampling) or by the packet content (content-based sampling).

The metering process may support packet sampling. If sampling is supported the sampling configuration must be well defined. The sampling configuration includes the sampling method and all its parameters.

If the sampling configuration is changed during operation, the new sampling configuration with its parameters must be indicated to all collecting processes receiving the affected flow records. Changing the sampling configuration includes: adding a sampling function to the metering process, removing a sampling function from the metering process, change sampling method, and change sampling parameter(s).

In case of any change in the sampling configuration, all flow records metered by the previous sampling configuration must be terminated and exported according to the export configuration. The metering process must not merge the flow records generated with the new sampling configuration with the flow records generated with the previous sampling configuration.

### **5.3. Overload Behavior**

In case of an overload, for example lack of memory or processing power, the metering process may change its behavior in order to cope with the lack of resources. Possible reactions include:



- Reduce the number of flows to be metered. This can be achieved by more coarse-grained flow measurement or by a restriction of the flow records to a subset of the set of original ones.
- Start sampling packets before they are processed by the metering process or - if sampling is already performed - reduce the sampling frequency.
- Stop metering.
- Reducing the resource usage of competing processes on the same device. Example: reducing the packet forwarding throughput

Overload behavior is not restricted to the four options listed above. But in case the overload behavior induces a change of the metering process behavior, the overload behavior must be clearly defined.

For some flows, the change of behavior might have an impact on the data that would be stored in the associated flow records after the change, for example if the packet classification is changed or the sampling frequency. These flows must be considered as terminated and the associated flow records must be exported separately from new ones generated after the behavior change. The terminated flow records and new ones generated after the behavior change must not be merged by the metering process. The collecting process must be able to distinguish the affected flow records generated before and after the change of behavior. This requirement does not apply to flows and associated flow records not affected by the change of metering process behavior.

#### **5.4. Timestamps**

The metering process must be able to generate timestamps for the first and the last observation of a packet of a flow at the observation point. The timestamp resolution must be at least the one of the sysUpTime [[RFC3418](#)], which is one centisecond.

#### **5.5. Time Synchronization**

It must be possible to synchronize timestamps generated by a metering process with Coordinated Universal Time (UTC).

Note that the possibility of synchronizing timestamps of each single metering process with UTC implies the possibility of synchronizing timestamps generated by different metering processes.

Note that this does not necessarily imply that timestamps generated by the metering process are UTC timestamps. For example, this requirement can be met by using local system clock values as timestamps and adding an additional timestamp when exporting a report



to a collecting process. Then the collecting process can synchronize the timestamps by calculating the offset between UTC and the system clock of the metering process.

#### **5.6. Flow Expiration**

The metering process must be able to detect flow expirations. A flow is considered to be expired if no packet of this flow has been observed for a given timeout interval. The metering process may support means for detecting the expiration of a flow before a timeout occurs, for example by detecting the FIN or RST bits in a TCP connection. The procedure for detecting a flow expiration must be clearly defined.

#### **5.7. Multicast Flows**

For multicast flows containing packets replicated to multiple output interfaces, the metering process should be able to maintain discrete flow records per different output interface. For example, the metering process should be able to report an incoming multicast packet that is replicated to four output interfaces in four different flow records that differ by the output interface.

#### **5.8. Packet Fragmentation**

In case of IP packet fragmentation and depending on the classification scheme, only the zero-offset fragment of a single initial packet might contain sufficient information to classify the packet. Note that this fragment should be the first one generated by the router imposing the fragmentation [[RFC791](#)], but might not be the first one observed by the IPFIX device, due reordering reasons. The metering process may keep state of IP packet fragmentation in order to map fragments that do not contain sufficient header information correctly to flows.

#### **5.9. Ignore Port Copy**

The metering process may be able to ignore packets which are generated by a port copy function acting at the device where the observation point of a flow is located.

### **6. Data Export**

The following are requirements for exporting flow records out of the exporting process. Beside requirements on the data transfer, we separate requirements concerning the information model from requirements concerning the data model. Furthermore, we list requirements on reporting times and notification on specific events, and on anonymization of flow records.





### **6.1. Information Model**

The information model for the flow information export is the list of attributes of a flow to be contained in the report (including the semantics of the attributes).

This section lists attributes an exporting process must, should or may be able to report. This does not imply that each exported flow record must contain all required attributes. But it implies that it must be possible to configure the exporting process in a way that the information of all required attributes can be transmitted from the exporting process to the receiving collecting process(es) for each exported flow.

In other words, meeting the IPFIX requirements means that the exporting process in general must be able, via its configuration, to somehow support to report all the must fields, even if in certain circumstance or for certain applications, only a subset of the set of all must fields is needed and effectively reported.

Beyond that, the exporting process might offer to report further attributes not mentioned here. A particular flow record may contain some of the "required" attributes as well as some additional ones, for example covering future technologies.

This document does not impose that the following attributes are reported for every single flow record, especially for repetitive attributes. For example, if the observation point is the incoming packet stream at the IP interface with the ifIndex value 3, then this observation point does not have to be exported as part of every single flow record. Exporting it just once might give sufficient information to the collecting process.

The exporting process must be able to report the following attributes for each metered flow:

1. IP version number  
This requirement only applies if the observation point is located at a device supporting more than one version of IP.
2. source IP address
3. destination IP address
4. IP protocol type (TCP,UDP,ICMP,...)
5. if protocol type is TCP or UDP: source TCP/UDP port number
6. if protocol type is TCP or UDP: destination TCP/UDP port number
7. packet counter  
If a packet is fragmented, each fragment is counted as an individual packet.
8. byte counter  
The sum of the total length in bytes of all IP packets belonging to the flow. The total length of a packet covers IP



header and IP payload.

9. type of service octet (in case of IPv4), traffic class octet (in case of IPv6). According to [RFC 2474](#) these octets include the DiffServ Code Point that has a length of 6 bits.
10. in case of IPv6: Flow Label
11. if MPLS is supported at the observation point: the top MPLS label or the corresponding forwarding equivalence class (FEC, [\[RFC3031\]](#)) bound to that label. The FEC is typically defined by an IP prefix.
12. timestamp of the first packet of the flow
13. timestamp of the last packet of the flow
14. if sampling is used: sampling configuration
15. unique identifier of the observation point
16. unique identifier of the exporting process

The exporting process should be able to report the following attributes for each metered flow:

17. if protocol type is ICMP: ICMP type and code
18. input interface (ifIndex)  
This requirement does not apply if the observation point is located at a probe device.
19. output interface (ifIndex)  
This requirement does not apply if the observation point is located at a probe device.
20. multicast replication factor  
the number of outgoing packets originating from a single incoming multicast packet. This is a dynamic property of multicast flows, that may change over time. For unicast flows it has the constant value 1. The reported value must be the value of the factor at the time the flow record is exported.

The exporting process may be able to report the following attributes for each metered flow:

21. Time To Live (in case of IPv4) or Hop Limit (in case of IPv6)
22. IP header flags
23. TCP header flags
24. dropped packet counter at the observation point  
If a packet is fragmented, each fragment must be counted as an individual packet.
25. fragmented packet counter  
counter of all packets for which the fragmented bit is set in the IP header
26. next hop IP address
27. source BGP Autonomous System number (see [\[RFC1771\]](#))
28. destination BGP Autonomous System number
29. next hop BGP Autonomous System number



## **6.2. Data Model**

The data model describes how information is represented in flow records.

The data model must be extensible for future attributes to be added. Even if a set of attributes is fixed in the flow record, the data model must provide a way of extending the record by configuration or for certain implementations.

The data model used for exporting flow information must be flexible concerning the flow attributes contained in flow records. A flexible record format would offer the possibility of defining records in a flexible (customizable) way regarding the number and type of contained attributes.

The Data Model should be independent of the underlying transport protocol, i.e. the data transfer.

## **6.3. Data Transfer**

Requirements for the data transfer include reliability, congestion awareness, and security requirements. For meeting these requirements the exporting process can utilize existing security features provided by the device hosting the process and/or provided by the transport network. For example it can use existing security technologies for authentication and encryption or it can rely on physical protection of a separated network for transferring flow information.

### **6.3.1. Congestion Awareness**

For the data transfer, a congestion aware protocol must be supported.

### **6.3.2. Reliability**

Loss of flow records during the data transfer from the exporting process to the collecting process must be indicated at the collecting process. This indication must allow the collecting process to gauge the number of flow records lost. Possible reasons for flow records loss include but are not limited to:

1. Metering process limitations: lack of memory, processing power, etc. These limitations are already covered in [section 5.1](#).
2. Exporting process limitations: lack of memory, processing power, etc.
3. Data transfer problems: packets that carry flow records sent from the exporting process to the collecting process, are dropped by the network. Examples are connection failures and



losses by a transport protocol that specifically offers congestion avoidance without persistent transport-level reliability.

4. Collecting process limitations: it may be experiencing congestion and not able to buffer new flows records.
5. Operation and Maintenance: the collecting process is taken down for maintenance or other administrative purposes.

Please note that if an unreliable transport protocol is used, reliability can be provided by higher layers. If reliability is provided by higher layers, only lack of overall reliability must be indicated. For example reordering could be dealt with by adding a sequence number to each packet.

The data transfer between exporting process and collecting process must be open to reliability extensions including at least

- retransmission of lost flow records,
- detection of disconnection and fail-over, and
- acknowledgement of flow records by the collecting process.

This extensibility may be used to provide additional reliability. The extended protocol must still meet the requirements described in this section, particularly, it must still be congestion aware. Therefore, extensions using retransmissions must use exponential backoff.

### **6.3.3. Security**

Confidentiality of IPFIX data transferred from an exporting process to a collecting process must be ensured.

Integrity of IPFIX data transferred from an exporting process to a collecting process must be ensured.

Authenticity of IPFIX data transferred from an exporting process to a collecting process must be ensured.

The security requirements have been derived from an analysis of potential security threads. The analysis is summarized in [Section 10](#).

### **6.4. Push and Pull Mode Reporting**

In general, there are two ways of deciding on reporting times: push mode and pull mode. In push mode, the exporting process decides without an external trigger when to send flow records. In pull mode, sending flow records is triggered by an explicit request from a collecting process. The exporting process must support push mode reporting, it may support pull mode reporting.





### **6.5. Regular Reporting Interval**

The exporting process should be capable of reporting measured traffic data regularly according to a given interval length.

### **6.6. Notification on Specific Events**

The exporting process may be capable of sending notifications to a collecting process, if a specific event occurs. Such an event can be for instance the arrival of the first packet of a new flow, or the termination of a flow after flow timeout.

### **6.7. Anonymization**

The exporting process may be capable of anonymizing source and destination IP addresses in flow data before exporting them. It may support anonymization of port numbers and other fields. Please note that anonymization is not originally an application requirement, but derived from general requirements for treatment of measured traffic data within a network.

For several applications anonymization cannot be applied, for example for accounting and traffic engineering. However, for protecting the network user's privacy, anonymization should be applied whenever possible. In many cases it is sufficient if anonymization is performed at the collecting process after flow information has been exported. This provides a reasonable protection of privacy as long as confidentiality of the export is provided.

It would be desirable to request that all IPFIX exporters provide anonymization of flow records, but algorithms for anonymization are still a research issue. Several are known but the security they provide and their other properties are not yet studied sufficiently. Also, there is no standardized method for anonymization. Therefore, the requirement for the exporting process supporting anonymization is qualified with 'may' and not with 'must'.

If anonymized flow data is exported, this must be clearly indicated to all receiving collecting processes, such that they can distinguish anonymized data from non-anonymized data.

## **7. Configuration**

If configuration is done remotely, security should be provided for the configuration process covering confidentiality, integrity and authenticity. The means used for remote configuration are out of the scope of this document.



### **7.1. Configuration of the Metering Process**

The metering process must provide a way of configuring traffic measurement. The following parameters of the metering process should be configurable:

1. specification of the observation point  
e.g. an interface or a list of interfaces to be monitored.
2. specifications of flows to be metered
3. flow timeouts

The following parameters may be configurable:

4. sampling method and parameters, if feature is supported
5. overload behavior, if feature is supported

### **7.2. Configuration of the Exporting Process**

The exporting process must provide a way of configuring the data export. The following parameters of the exporting process should be configurable:

1. reporting data format  
Specifying the reporting data format must include a selection of attributes to be reported for each flow.
2. the collecting process(es) to which flows are reported
3. the reporting interval  
This requirement only applies if the exporting process supports reporting in regular intervals.
4. notifications to be sent to the collecting process(es)  
This requirement only applies if the exporting process supports notifications.
5. flow anonymization  
This requirement only applies if the exporting process supports flow anonymization.

## **8. General Requirements**

### **8.1. Openness**

IPFIX specifications should be open to future technologies. This includes extensibility of configuration of the metering process and the exporting process.

Openness is also required concerning the extensibility of the data model, as stated in [section 6.2](#).



### **8.2. Scalability**

Data collection from hundreds of different exporting processes must be supported. The collecting process must be able to distinguish several hundred exporting processes by their identifiers.

### **8.3. Several Collecting Processes**

The exporting process may be able to export flow information to more than one collecting process. If an exporting process is able to export flow records to multiple collecting processes then it must be able to ensure that the flow records can be identified so that duplicates can be detected between different collecting processes and double counting problems can be avoided.

## **9. Special Device Considerations**

This document intends to avoid constraining the architecture of probes, routers, and other devices hosting observation points, metering processes, exporting processes, and/or collecting processes. It can be expected that typically observation point, metering process, and exporting process are co-located at a single device. However, the requirements defined in this document do not exclude devices that derive from this configuration. Figure 2 shows some examples.

All examples are composed of one or more of the following elements: observation point (O), metering process (M), exporting process (E), collecting process (C). The observation points shown in the figure are always the most fine-granular ones supported by the respective device.



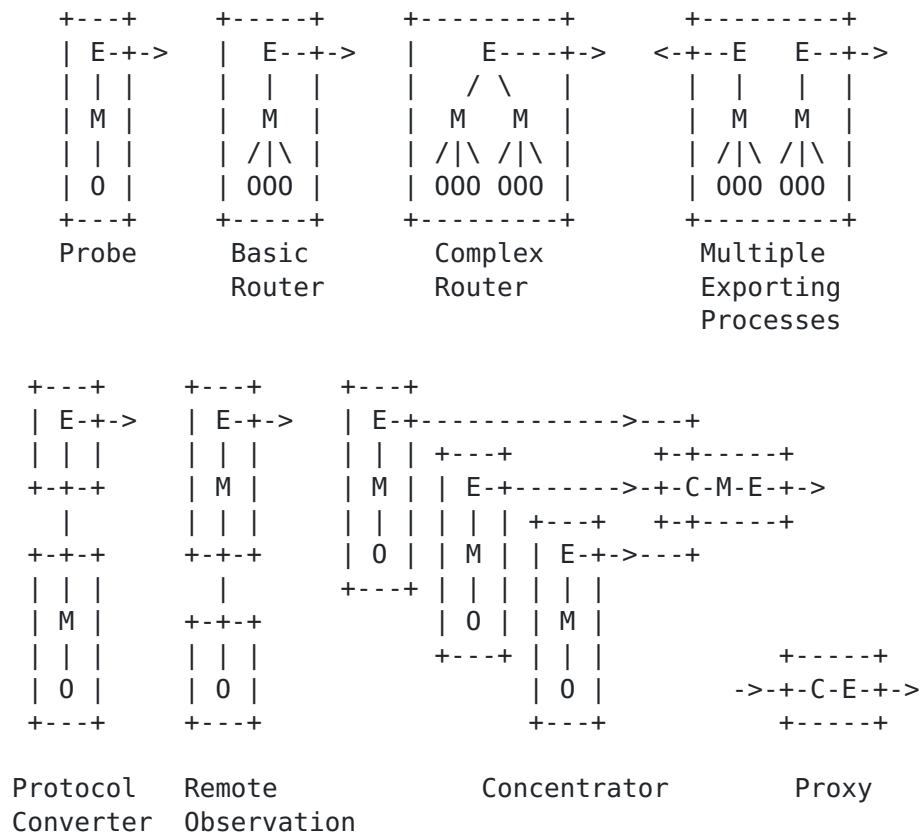


Figure 2: IPFIX-related Devices

A very simple device is a probe. A typical probe contains a single observation point, a single metering process, and a single exporting process.

A basic router extends this structure by multiple observation points. Here, the observation point of a particular flow may be one of the displayed most fine-granular observation points, but also it may be a set of them.

A more complex router may host more than one metering process, for example one per line card. Please note that here, the observation point of a single flow cannot exceed the set of most fine-granular observation points linked to a single metering process, because only the metering process can merge packets observed at different fine-granular observation points to a joint flow. An observation point containing all most fine-granular observation points of this router is not possible with this structure. Alternatively, a complex router may host different exporting processes for flow records generated by different metering processes.

A protocol converter makes use of a metering process that can be





accessed only by protocol(s) other than the one defined for IPFIX, for example, the SNMP and the Meter MIB module [[RFC2720](#)]. Then the exporting process receives flow record from a remote metering process and exports these records using the IPFIX protocol. Please note that this document does not make any particular assumption on how metering processes and export processes exchange information, as long as all individual requirements for these processes are met. Also the locations of metering processes are not of any relevance for this document (in contrast to the locations of observation points and the exporting processes).

In the example of remote packet observation in Figure 2 the metering process and the observation point are not co-located. Packet headers captured at an observation point may be exported as raw data to a device hosting metering process and exporting process. Again, this document does not make any particular assumption on how packet headers are transferred from observation points to metering processes, as long as all requirements for the metering processes are met.

An intermediate structure between protocol converter and remote observation (not shown in the Figure) would be a split metering process, for example performing timestamping and sampling at the device hosting the observation point and performing packet classification at another device hosting the exporting process.

A concentrator receives flow records via the IPFIX protocol, merges them into more aggregated flow records, and exports them again using the IPFIX protocol. Please note that for the final flow records the resulting observation point may be a superset of the more fine-granular observation points at the first level devices. The metering process of the final flow records is composed by the (partial) metering processes at the first level devices and the partial metering process at the concentrator.

Finally, a very simple IPFIX-related device is a proxy. It just receives flow records using the IPFIX protocol and sends them further using the same protocol. A proxy might be useful for traversing firewalls or other gateways.

## **10. Security Considerations**

An IPFIX protocol must be capable of transporting data over the public Internet. Therefore it cannot be excluded that an attacker captures or modifies packets or inserts additional packets.

This section describes security requirements for IPFIX. Like other requirements, the security requirements differ among the considered applications. The incentive to modify collected data for accounting



or intrusion detection for instance is usually higher than the incentive to change data collected for traffic profiling. A detailed list of the required security features per application can be found in the appendix.

The suggestion of concrete solutions for achieving the required security properties should be part of an IPFIX architecture and protocol. It is out of scope of this document. Also methods for remote configuration of the metering processes and exporting processes are out of scope. Therefore, threats that are caused by data exchange for remote configuration are not considered here.

The following potential security hazards for an IPFIX protocol have been identified: disclosure of IP flow information, forgery of flow records, and Denial of Service (DoS) attacks.

### **10.1. Disclosure of Flow Information Data**

The content of data exchanged by an IPFIX protocol (for example IPFIX flow records) should be kept confidential between the involved parties (exporting process and collecting process). Observation of IPFIX flow records gives an attacker information about the active flows in the network, communication endpoints and traffic patterns. This information cannot only be used to spy on user behavior but also to plan and conceal future attacks. Therefore, the requirements specified in [section 6.3.3](#) include confidentiality of the transferred data. This can be achieved for instance by encryption.

Also the privacy of users acting as sender or receiver of the measured traffic needs to be protected when they use the Internet. In many countries the right to store user-specific data (including the user's traffic profiles) is restricted by law or by regulations.

In addition to encryption, this kind of privacy can also be protected by anonymizing flow records. For many traffic flow measurements, anonymized data is as useful as precise data. Therefore, it is desirable to support anonymization in IPFIX implementations. It is beyond the scope of the IPFIX working group to develop and standardize anonymization methods. However, the requirements for extensibility of the IPFIX protocol are sufficient to support anonymized flow records when appropriate methods are standardized.

### **10.2. Forgery of Flow Records**

If flow records are used in accounting and/or security applications, there are potentially strong incentives to forge exported IPFIX flow records (for example to save money or prevent the detection of an attack). This can be done either by altering flow records on the path or by injecting forged flow records that pretend to be originated by the original exporting process.



Special caution is required if security applications rely on flow measurements. With forged flow records it is possible to trick security applications. For example, an application may be lead to falsely conclude that a DoS attack is in progress. If such an injection of IPFIX traffic flow records fools the security application, causing it to erroneously conclude that a DoS attack is underway, then the countermeasures employed by the security application may actually deny useful non-malicious services.

In order to make an IPFIX protocol resistant against such attacks, authentication and integrity must be provided, as specified in [section 6.3.3](#).

### **[10.3](#). Denial of Service (DoS) Attacks**

DoS attacks on routers or other middleboxes that have the IPFIX protocol implemented would also affect the IPFIX protocol and impair the sending of IPFIX records. Nevertheless, since such hazards are not induced specifically by the IPFIX protocol the prevention of such attacks is out of scope of this document.

However, IPFIX itself also causes potential hazards for DoS attacks. All processes that expect the reception of traffic can be target of a DoS attack. With the exporting process this is only the case if it supports the pull mode (which can be an optional feature of the IPFIX protocol according to this document). The collecting process always expects data and therefore can be flooded by flow records.

## **[11](#). Acknowledgments**

Many thanks Georg Carle for contributing to the application analysis, to K.C. Norseth for several fine-tunings, to Sandra Tartarelli for checking the appendix, and to a lot of further people on the mailing list providing valuable comments and suggestions including Nevil Brownlee, Carter Bullard, Paul Calato, Ram Gopal, Tal Givoly, Jeff Meyer, Reinaldo Penno, Sonia Panchen, Simon Leinen, David Plonka, Ganesh Sadasivan, Kevin Zhang, and many more.



## 12. Appendix: Derivation of Requirements from Applications

The following table documents, how the requirements stated in sections 3-7 are derived from requirements of the applications listed in section 2.

Used abbreviations:

M = must

S = should

O = may (optional)

- = DONT CARE

IPFIX								
E: QoS Monitoring								
D: Attack/Intrusion Detection								
C: Traffic Engineering								
B: Traffic Profiling								
A: Usage-based Accounting								
Sect.	Requirement	A	B	C	D	E	IPFIX	
4.	DISTINGUISHING FLOWS							
4.	Combination of required attributes	M	M	M	M	M	M	
4.1.	in/out IF	S	M	M	S	S	M	
4.2.	src/dst address	M	M	M	M	M	M	
4.2.	Masking of IP addresses	M	M	M	M	M	M	
4.2.	transport protocol	M	M	-	M	M	M	
4.2.	version field	-	S	S (b)	O	O	S	
4.3.	src/dst port	M	M	-	M	M	M	
4.4.	MPLS label (a)	S	S	M (c)	O	S	M	





Sect.	Requirement	A	B	C	D	E	IPFIX
4.5.	DSCP (a)	M	S	M	0	M	M
5.	METERING PROCESS						
5.1.	Reliability	M	S	S	S	S	M
5.1.	Indication of missing reliability	-	M	M	M	M	
5.2.	Sampling (d,e)	0	0	0	0	0	0
5.3.	Overload Behavior (f)	0	0	0	0	0	0
5.4.	Timestamps	M	0	0	S	M	M
5.5.	Time synchronization	M	S	S	S	M	M
5.6.	Flow timeout	M	S	-	0	0	M
		(g)					
5.7.	Multicast flows	S	0	0	0	S	S
5.8.	Packet fragmentation	0	0	-	-	-	0
5.9.	Ignore port copy	0	0	0	0	0	0
6.	DATA EXPORT						
6.1.	INFORMATION MODEL						
6.1.	IP Version	-	M	M	0	0	M
6.1.	src/dst address	M	M	M	M	M	M
6.1.	transport protocol	M	M	-	M	M	M
6.1.	src/dst port	M	M	-	M	M	M
6.1.	Packet counter (h)	S	M	M	S	S	M
6.1.	Byte counter	M	M	M	S	S	M
6.1.	ToS (IPv4) or traffic class octet (IPv6)	M	S	M	0	M	M



Sect.	Requirement	A	B	C	D	E	IPFIX
6.1.	Flow Label (IPv6)	M	S	M	0	M	M
6.1.	MPLS label (a)	S	S	M (c)	0	S	M
6.1.	Timestamps for first/last packet	M	0	0	S	S	M
6.1.	Sampling configuration	M	M	M	M	M	M
6.1.	observation point identifier	M	M	M	M	M	M
6.1.	export process identifier	M	M	M	M	M	M
6.1.	ICMP type and code (i)	S	S	-	S	S	S
6.1.	input/output interface (j)	S	S	S	S	S	S
6.1.	Multicast replication factor	0	S	S	-	S	S
6.1.	TTL	0	0	0	0	0	0
6.1.	IP header flags	-	0	0	0	0	0
6.1.	TCP header flags	-	0	0	0	-	0
6.1.	Dropped Packet Counter (h,k)	0	0	0	0	0	0
6.1.	Fragment counter	-	0	0	0	0	0
6.1.	next hop IP address	0	0	0	0	-	0
6.1.	src / dst / next hop BGP AS #	-	0	0	-	-	0
6.2.	DATA MODEL						
6.2.	Flexibility	M	S	M	M	M	M
6.2.	Extensibility	M	S	M	M	M	M



Sect.	Requirement	A	B	C	D	E	IPFIX
6.3.	DATA TRANSFER						
6.3.1.	Congestion aware	M	M	M	M	M	M
6.3.2.	Reliability	M	S	S	S	S	M
6.3.3.	Confidentiality	M	S	S	M	S	M
6.3.4.	Integrity	M	M	M	M	M	M
6.3.5.	Authenticity	M	M	M	M	M	M
6.4.	REPORTING TIMES						
6.4.	Push mode	M	0 (l)	0 (l)	M	S (l,m)	M
6.4.	Pull mode	0	0 (l)	0 (l)	0	0 (l)	0
6.4.1.	Regular interval	S	S	S	S	S	S
6.6.	Notifications	0	0	0	0	0	0
6.7.	Anonymization (n)	0	0	0	0	0	0
7.	CONFIGURATION						
7.	Secure remote configuration (a)	S	S	S	S	S	S
7.1.	Config observation point	S	S	S	S	S	S
7.1.	Config flow specifications	S	S	S	S	S	S
7.1.	Config flow timeouts	S	S	S	S	0	S
7.1.	Config sampling	0	0	0	0	0	0
7.1.	Config overload behavior (a)	0	0	0	0	0	0
7.2.	Config report data format	S	S	S	S	S	S



Sect.	Requirement	A	B	C	D	E	IPFIX
7.2.	Config notifications	S	S	S	S	S	S
8.	GENERAL REQUIREMENTS						
8.1.	Openness	S	S	S	S	S	S
8.2.	Scalability: data collection from hundreds of measurement devices	M	S	M	O	S	M
8.3.	Several collectors	O	O	O	O	O	O

## Remarks:

- (a) If feature is supported.
- (b) The differentiation of IPv4 and IPv6 is for TE of importance. So we tended to make this a must. Nevertheless, a should seems to be sufficient to perform most TE tasks and allows us to have a should for IPFIX instead of a must.
- (c) For TE in an MPLS network the label is essential. Therefore a must is given here leading to a must in IPFIX.
- (d) If sampling is supported, the methods and parameters must be well defined.
- (e) If sampling is supported, sampling configuration changes must be indicated to all collecting processes.
- (f) If overload behavior is supported and it induces changes in the metering process behavior, the overload behavior must be clearly defined.
- (g) Precise time-based accounting requires reaction to a flow timeout.
- (h) If a packet is fragmented, each fragment is counted as an individual packet.
- (i) If protocol type is ICMP.
- (j) This requirement does not apply if the observation point is located at a probe device.
- (k) Only if measurement is done on data path i.e. has access to forwarding decision.
- (l) Either push or pull has to be supported.
- (m) Required, in order to immediately report drop indications for SLA validation.
- (n) Anonymization must be clearly indicated to all receiving collecting processes.





### **13. Normative References**

- [RFC2960] 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.
- [RFC3031] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", [RFC 3031](#), January 2001.
- [RFC2474] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), December 1998.
- [RFC791] J. Postel. "Internet Protocol", [RFC791](#), September 1981.

### **14. Informative References**

- [RFC3234] B. Carpenter, "Middleboxes: taxonomy and issues", [RFC 3234](#), February 2002.
- [RFC2119] S. Bradner "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.
- [RFC1889] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", [RFC 1889](#), January 1996.
- [RFC2975] Aboba, B., Arkko, J. and D. Harrington, "Introduction to Accounting Management", [RFC 2975](#), October 2000.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and J. McManus, "Requirements for Traffic Engineering Over MPLS", [RFC 2702](#), September 1999.
- [RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", [RFC 1771](#), March 1995.
- [RFC3418] R. Presuhn (Ed.), "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", [RFC 3418](#), December 2002.
- [RFC2720] N. Brownlee, "Traffic Flow Measurement: Meter MIB", [RFC 2720](#), October 1999.



## **15. Authors' Addresses**

Juergen Quittek  
NEC Europe Ltd., Network Laboratories  
Kurfuersten-Anlage 36  
69115 Heidelberg  
Germany

Phone: +49 6221 90511 15  
EMail: quittek@netlab.nec.de

Tanja Zseby  
Fraunhofer Institute for Open Communication Systems (FOKUS)  
Kaiserin-Augusta-Allee 31  
10589 Berlin  
Germany

Phone: +49 30 3463 7153  
Email: zseby@fokus.fhg.de

Benoit Claise  
Cisco Systems  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

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

Sebastian Zander  
Fraunhofer Institute for Open Communication Systems (FOKUS)  
Kaiserin-Augusta-Allee 31  
10589 Berlin  
Germany

Phone: +49 30 3463 7287  
Email: zander@fokus.fhg.de



## **16. IPR Notices**

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

## **17. Full Copyright Statement**

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

