Network Working Group
Internet-Draft
Expires: March 26, 2006

J. Quittek
NEC
S. Bryant
B. Claise
Cisco Systems
J. Meyer
PayPal

September 22, 2005

Information Model for IP Flow Information Export
draft-ietf-ipfix-info-11

Abstract

This memo defines an information model for the IP Flow Information eXport (IPFIX) protocol. It is used by the IPFIX protocol for encoding measured traffic information and information related to the traffic Observation Point, the traffic Metering Process and the

Exporting Process. Although developed for the IPFIX protocol, the model is defined in an open way that easily allows using it in other protocols, interfaces, and applications.

Table of Contents

1. Introduction 7
2. Properties of IPFIX Protocol Information Elements 7
2.1. Information Elements Specification Template 8
2.2. Scope of Information Elements 9
2.3. Naming Conventions for Information Elements 9
3. Type Space 10
3.1. Data Types 10
3.1.1. octet 10
3.1.2. unsigned16 11
3.1.3. unsigned32 11
3.1.4. unsigned64 11
3.1.5. float32 11
3.1.6. boolean 11
3.1.7. macAddress 11
3.1.8. octetArray 11
3.1.9. string 11
3.1.10. dateTimeSeconds 11
3.1.11. dateTimeMilliSeconds 12
3.1.12. dateTimeMicroSeconds 12
3.1.13. dateTimeNanoSeconds 12
3.1.14. ipv4Address 12
3.1.15. ipv6Address 12
3.2. Data Type Semantics 12
3.2.1. quantity 12
3.2.2. totalCounter 12
3.2.3. deltaCounter 13
3.2.4. identifier 13
3.2.5. flags 13
4. Information Element Identifiers 13
5. Information Elements 17
5.1. Identifiers 18
5.1.1. lineCardId 18
5.1.2. portId 18
5.1.3. ingressInterface 18
5.1.4. egressInterface 19
5.1.5. meteringProcessId 19
5.1.6. exportingProcessId 19
5.1.7. flowId 19
5.1.8. templateId 20
5.1.9. sourceId 20
5.2. Metering and Exporting Process Properties 20

5.2.1. exporterIPv4Address 21
5.2.2. exporterIPv6Address 21
5.2.3. exportedMessageTotalCount 21
5.2.4. exportedOctetTotalCount 21
5.2.5. exportedFlowTotalCount 22
5.2.6. observedFlowTotalCount 22
5.2.7. ignoredPacketTotalCount 22
5.2.8. ignoredOctetTotalCount 22
5.2.9. notSentFlowTotalCount 23
5.2.10. notSentPacketTotalCount 23
5.2.11. notSentOctetTotalCount 23
5.2.12. flowKeyIndicator 24
5.3. IP Header Fields 24
5.3.1. ipVersion 25
5.3.2. sourceIPv4Address 25
5.3.3. sourceIPv6Address 26
5.3.4. sourceIPv4Mask 26
5.3.5. sourceIPv6Mask 26
5.3.6. sourceIPv4Prefix 26
5.3.7. sourceIPv6Prefix 26
5.3.8. destinationIPv4Address 27
5.3.9. destinationIPv6Address 27
5.3.10. destinationIPv4Mask 27
5.3.11. destinationIPv6Mask 27
5.3.12. destinationIPv4Prefix 28
5.3.13. destinationIPv6Prefix 28
5.3.14. ipTimeToLive 28
5.3.15. protocolIdentifier 28
5.3.16. nextHeaderIPv6 29
5.3.17. ipClassOfService 29
5.3.18. ipDiffServCodePoint 29
5.3.19. ipPrecedence 30
5.3.20. classOfServiceIPv4 30
5.3.21. postClassOfServiceIPv4 30
5.3.22. classOfServiceIPv6 31
5.3.23. postClassOfServiceIPv6 31
5.3.24. flowLabelIPv6 31
5.3.25. isMulticast 32
5.3.26. identificationIPv4 32
5.3.27. fragmentOffsetIPv4 33
5.3.28. fragmentFlagsIPv4 33
5.3.29. ipHeaderLength 33
5.3.30. headerLengthIPv4 34
5.3.31. internetHeaderLengthIPv4 34
5.3.32. totalLengthIPv4 34
5.3.33. payloadLengthIPv6 35
5.3.34. ipPayloadLength 35
5.4. Transport Header Fields 35

5.4.1. sourceTransportPort 36
5.4.2. destinationTransportPort 36
5.4.3. udpSourcePort 37
5.4.4. udpDestinationPort 37
5.4.5. udpMessageLength 37
5.4.6. tcpSourcePort 37
5.4.7. tcpDestinationPort 38
5.4.8. tcpSequenceNumber 38
5.4.9. tcpAcknowledgementNumber 38
5.4.10. tcpWindowSize 38
5.4.11. tcpUrgentPointer 38
5.4.12. tcpHeaderLength 39
5.4.13. icmpTypeCodeIPv4 39
5.4.14. icmpTypeIPv4 39
5.4.15. icmpCodeIPv4 39
5.4.16. icmpTypeCodeIPv6 40
5.4.17. icmpTypeIPv6 40
5.4.18. icmpCodeIPv6 40
5.4.19. igmpType 40
5.5. Sub-IP Header Fields 40
5.5.1. sourceMacAddress 41
5.5.2. postSourceMacAddress 41
5.5.3. vlanId 41
5.5.4. postVlanId 42
5.5.5. destinationMacAddress 42
5.5.6. postDestinationMacAddr 42
5.5.7. wlanChannelId 42
5.5.8. wlanSsid 43
5.5.9. mplsTopLabelTtl 43
5.5.10. mplsTopLabelExp 43
5.5.11. mplsLabelStackDepth 44
5.5.12. mplsLabelStackLength 44
5.5.13. mplsPayloadLength 44
5.5.14. mplsTopLabelStackEntry 44
5.5.15. mplsLabelStackEntry2 45
5.5.16. mplsLabelStackEntry3 45
5.5.17. mplsLabelStackEntry4 45
5.5.18. mplsLabelStackEntry5 46
5.5.19. mplsLabelStackEntry6 46
5.5.20. mplsLabelStackEntry7 46
5.5.21. mplsLabelStackEntry8 47
5.5.22. mplsLabelStackEntry9 47
5.5.23. mplsLabelStackEntry10 47
5.6. Derived Packet Properties 47
5.6.1. ipNextHopIPv4Address 48
5.6.2. ipNextHopIPv6Address 48
5.6.3. bgpSourceAsNumber 48
5.6.4. bgpDestinationAsNumber 49

5.6.5. bgpNextAdjacentAsNumber 49
5.6.6. bgpPrevAdjacentAsNumber 49
5.6.7. bgpNextHopIPv4Address 50
5.6.8. bgpNextHopIPv6Address 50
5.6.9. mplsTopLabelType 50
5.6.10. mplsTopLabelIPv4Address 50
5.6.11. mplsTopLabelIPv6Address 51
5.7. Min/Max Flow Properties 51
5.7.1. minimumPacketLength 51
5.7.2. maximumPacketLength 51
5.7.3. minimumTtl 52
5.7.4. maximumTtl 52
5.7.5. ipv4Options 52
5.7.6. ipv6ExtensionHeaders 54
5.7.7. tcpControlBits 56
5.7.8. tcpOptions 56
5.8. Flow Time Stamps 57
5.8.1. flowStartSeconds 57
5.8.2. flowEndSeconds 58
5.8.3. flowStartMilliSeconds 58
5.8.4. flowEndMilliSeconds 58
5.8.5. flowStartMicroSeconds 58
5.8.6. flowEndMicroSeconds 58
5.8.7. flowStartNanoSeconds 58
5.8.8. flowEndNanoSeconds 59
5.8.9. flowStartDeltaMicroSeconds 59
5.8.10. flowEndDeltaMicroSeconds 59
5.8.11. systemInitTimeMilliSeconds 59
5.8.12. flowStartSysUpTime 60
5.8.13. flowEndSysUpTime 60
5.9. Per-Flow Counters 60
5.9.1. octetDeltaCount 61
5.9.2. postOctetDeltaCount 61
5.9.3. octetDeltaSumOfSquares 61
5.9.4. octetTotalCount 62
5.9.5. postOctetTotalCount 62
5.9.6. octetTotalSumOfSquares 62
5.9.7. packetDeltaCount 62
5.9.8. postPacketDeltaCount 63
5.9.9. packetTotalCount 63
5.9.10. postPacketTotalCount 63
5.9.11. droppedOctetDeltaCount 64
5.9.12. droppedPacketDeltaCount 64
5.9.13. droppedOctetTotalCount 64
5.9.14. droppedPacketTotalCount 64
5.9.15. postMCastPacketDeltaCount 65
5.9.16. postMCastOctetDeltaCount 65
5.9.17. postMCastPacketTotalCount 65

5.9.18. postMCastOctetTotalCount 66
5.10. Miscellaneous Flow Properties 66
5.10.1. flowActiveTimeOut 66
5.10.2. flowInactiveTimeout 66
5.10.3. flowEndReason 67
5.10.4. flowDurationMilliSeconds 67
5.10.5. flowDurationMicroSeconds 68
5.11. Padding 68
5.11.1. paddingOctets 68
6. Extending the Information Model 68
7. IANA Considerations 69
8. Security Considerations 70
9. Acknowledgements 70
10. References 70
10.1. Normative References 70
10.2. Informative References 70
Appendix A. Formal Specification of IPFIX Information Element . 73
Appendix B. Formal Specification of Abstract Data Types 127
Authors' Addresses 139
Intellectual Property and Copyright Statements 140

1. Introduction

The IP Flow Information eXport (IPFIX) protocol serves for transmitting information related to measured IP traffic over the Internet. The protocol specification in [I-D.ietf-ipfix-protocol] defines how Information Elements are transmitted. For Information Elements, it specifies the encoding of a set of basic data types. However, the list of Information Elements that can be transmitted by the protocol, such as Flow attributes (source IP address, number of packets, etc.) and information about the Metering and Exporting Process (packet Observation Point, sampling rate, Flow timeout interval, etc.), is not specified in [I-D.ietf-ipfix-protocol].

This document complements the IPFIX protocol specification by providing the IPFIX information model. IPFIX-specific terminology used in this document is defined in section 3 of [I-D.ietf-ipfix-protocol]. As in [I-D.ietf-ipfix-protocol], these IPFIX-specific terms have the first letter of a word capitalized when used in this document.

The main part of this document is section 5 defining the (extensible) list of Information Elements to be transmitted by the IPFIX protocol. Section 2 defines a template for specifying IPFIX Information Elements in section 4. Section 3 defines the set of abstract data types that are available for IPFIX Information Elements. Section 5 discusses extensibility of the IPFIX information model.

The main bodies of sections 2, 3 and 4 were generated from XML documents. The XML-based specification of template, abstract data types and IPFIX Information Elements can be used for automatically checking syntactical correctness of the specification of IPFIX Information Elements. It can further be used for generating IPFIX protocol implementation code that deals with processing IPFIX Information Elements. Also code for applications that further process traffic information transmitted via the IPFIX protocol can be generated with the XML specification of IPFIX Information Elements.

For that reason, the XML document that served as source for section 4 and the XML schema that served as source for sections 2 and 3 are attached to this document in Appendices A and B.

Note that although partially generated from the attached XML documents, the main body of this document is normative while the appendices are informational.

2. Properties of IPFIX Protocol Information Elements

2.1. Information Elements Specification Template

Information in messages of the IPFIX protocol is modeled in terms of Information Elements of the IPFIX information model. IPFIX Information Elements are specified in section 4. For specifying these Information Elements, a template is used that is described below.

All Information Elements specified for the IPFIX protocol either in this document or by any future extension MUST have the following properties defined:

name - A unique and meaningful name for the Information Element.

description - The semantics of this Information Element. Describes how this Information Element is derived from the Flow or other information available to the observer.

dataType - One of the types listed in section 3.1 of this document or in a future extension of the information model. The type space for attributes is constrained to facilitate implementation. The existing type space does however encompass most basic types used in modern programming languages, as well as some derived types (such as ipv4Address) which are common to this domain and useful to distinguish.

status - The status of the specification of this Information Element. Allowed values are 'current', 'deprecated', and 'obsolete'.

Enterprise-specific Information Elements MUST have the following property defined:

enterpriseId - Enterprises may wish to define Information Elements without registering them with IANA, for example for enterprise-internal purposes. For such Information Elements the Information Element identifier described above is not sufficient when the Information Element is used outside the enterprise. If specifications of enterprise-specific Information Elements are made public and/or if enterprise-specific identifiers are used by the IPFIX protocol outside the enterprise, then the enterprise-specific identifier MUST be made globally unique by combining it with an enterprise identifier. Valid values for the enterpriseId are defined by IANA as SMI network management private enterprise codes. They are defined at http://www.iana.org/assignments/enterprise-numbers.

All Information Elements specified for the IPFIX protocol either in this document or by any future extension MAY have the following properties defined:

dataTypeSemantics - The integral types may be qualified by additional semantic details. Valid values for the data type semantics are specified in section 3.2 of this document or in a future extension of the information model.

units - If the Information Element is a measure of some kind, the units identify what the measure is.

range - Some Information Elements may only be able to take on a restricted set of values which can be expressed as a range (e.g. 0 through 511 inclusive). If this is the case, the valid inclusive range should be specified.

reference - Identifies additional specifications which more precisely define this item or provide additional context for its use.

2.2. Scope of Information Elements

By default, most Information Elements have a scope specified in their definitions.

- The Information Elements defined in section 5.2 have a default of "a specific Metering Process" or of "a specific Exporting Process", respectively.
- The Information Elements defined in sections 5.3 - 5.9 have a scope of "a specific Flow".

Within Data Records defined by Option Templates, the IPFIX protocol allows further limiting of the Information Element scope. The new scope is specified by one or more scope fields and defined as the combination of all specified scope values.

2.3. Naming Conventions for Information Elements

The following naming conventions were used for naming Information Elements in this document. It is recommended that extensions of the model use the same conventions.

- Names of Information Elements start with non-capitalized letters.

o Composed names use capital letters for the first letter of each component (except for the first one). All other letters are non-capitalized, even for acronyms. Exceptions are made for acronyms containing non-capitalized letter, such as 'IPv4' and 'IPv6'. Examples are sourceMacAddress and destinationIPv4Address.
o Middleboxes [RFC3234] may change Flow properties, such as the DSCP value or the source IP address. If an IPFIX Observation Point is located in the path of a Flow before one or more middleboxes that potentially modify packets of the Flow, then it may be desirable to report also flow properties after the modification performed by the middleboxes. An example is an observation point before a packet marker changing a packet's IPv4 TOS field that is encoded in Information Element classOfServiceIPv4. Then the value observed and reported by Information Element classOfServiceIPv4 is valid at the observation point, but not anymore after the packet passed the packet marker. For reporting the change value of the TOS field, the IPFIX information model uses Information Elements that have a name prefix "post", for example, "postClassOfServiceIPv4". Information Elements with prefix "post" report on Flow properties that are not necessarily observed at the observation point, but which are obtained withing the Flow's Observation Domain by other means that are considered to be sufficiently reliable, for example, by analyzing the packet marker's marking tables.

3. Type Space

This section describes the abstract data types that can be used for the specification of IPFIX Information Elements in section 4. Section 3.1 describes the set of data types.

Data types octet, unsigned16, unsigned32, and unsigned64 are integral data types. As described in section 3.2, their data type semantics can be further specified, for example, by 'totalCounter', 'deltaCounter', 'identifier' or 'flags'.

3.1. Data Types

This section describes the set of valid data types of the IPFIX information model. Note that further data types may be specified by future protocol extensions.

3.1.1. octet

The type "octet" represents a non-negative integer value in the range of 0 to 255.

3.1.2. unsigned16

The type "unsigned16" represents a non-negative integer value in the range of 0 to 65535.

3.1.3. unsigned32

The type "unsigned32" represents a non-negative integer value in the range of 0 to 4294967295.

3.1.4. unsigned64

The type "unsigned64" represents a non-negative integer value in the range of 0 to 18446744073709551615.

3.1.5. float32

The type "float32" corresponds to an IEEE single-precision 32-bit floating point type as defined in [IEEE.754.1985].

3.1.6. boolean

The type "boolean" represents a binary value. The only allowed values are "true" and false.

3.1.7. macAddress

The type "macAddress" represents a string of 6 octets.

3.1.8. octetArray

The type "octetArray" represents a finite length string of octets.

3.1.9. string

The type "string" represents a finite length string of valid characters from the Unicode character encoding set [ISO.10646-1.1993]. Unicode allows for ASCII [ISO.646.1991] and many other international character sets to be used. It is expected that strings will be encoded in UTF-8 format, which is identical in encoding for ASCII characters, but also accommodates other Unicode multi-byte characters.

3.1.10. dateTimeSeconds

The type "dateTimeSeconds" represents a time value in units of seconds normalized to the GMT time zone.

3.1.11. dateTimeMilliSeconds

The type "dateTimeMilliSeconds" represents a time value in units of milliseconds normalized to the GMT time zone.

3.1.12. dateTimeMicroSeconds

The type "dateTimeMicroSeconds" represents a time value in units of microseconds normalized to the GMT time zone.

3.1.13. dateTimeNanoSeconds

The type "dateTimeNanoSeconds" represents a time value in units of nanoseconds normalized to the GMT time zone.

3.1.14. ipv4Address

The type "ipv4Address" represents a value of an IPv4 address.

3.1.15. ipv6Address

The type "ipv6Address" represents a value of an IPv6 address.

3.2. Data Type Semantics

This section describes the set of valid data type semantics of the IPFIX information model. Note that further data type semantics may be specified by future protocol extensions.

3.2.1. quantity

A quantity value represents a discrete measured value pertaining to the record. This is distinguished from counters which represent an ongoing measured value whose "odometer" reading is captured as part of a given record. If no semantic qualifier is given, the Information Elements that have an integral data type should behave as a quantity.

3.2.2. totalCounter

An integral value reporting the value of a counter. Basically the same semantics as counters in SNMP. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of 2**64 - 1. At this point the next increment will wrap its value to zero and continue counting from zero. A running counter counts independently of the export of its value.

3.2.3. deltaCounter

An integral value reporting the value of a counter. Basically the same semantics as counters in SNMP. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of 2**64 - 1. At this point the next increment will wrap its value to zero and continue counting from zero. A delta counter is reset to zero each time its value is exported.

3.2.4. identifier

An integral value which serves as an identifier. Specifically mathematical operations on two identifiers (aside from the equality operation) are meaningless. For example, Autonomous System ID 1 * Autonomous System ID 2 is meaningless.

3.2.5. flags

An integral value which actually represents a set of bit fields. Logical operations are appropriate on such values, but not other mathematical operations. Flags should always be of an unsigned type.

4. Information Element Identifiers

All Information Elements defined in section 5 of this document or in future extensions of the IPFIX information model have their identifiers assigned by IANA. Their identifiers can be retrieved at http://www.iana.org/assignments/ipfix-element-numbers.

EDITORIAL NOTE: this URL needs probably to be updated after IANA created a URL for IPFIX Information Elements

The value of these identifiers are in the range of 1 - 32767. Within this range, Information Element identifier values in the sub-range of 1-127 are compatible with field types used by NetFlow version 9 [RFC3954].

Range of IANA-assigned Information Element identifiers	Description
0	Reserved.
1 - 127	Information Element identifiers compatible with NetFlow version 9 field types [RFC3954].
128 - 32767	Further Information Element identifiers.

Enterprise-specific Information Element identifiers have the same range of 1-32767, but they are coupled with an additional enterprise identifier.

Enterprise-specific Information Element identifiers can be chosen by an enterprise arbitrarily within the range of 1-32767. The same identifier may be assigned by other enterprises for different purposes.

Still, Collecting Processes can distinguish these Information Elements because the Information Element identifier is coupled with an enterprise identifier.

Enterprise identifiers MUST be registered as SMI network management private enterprise code numbers with IANA. The registry can be found at http://www.iana.org/assignments/enterprise-numbers.

The following list gives an overview of the Information Element identifiers that are specified in section 5 and are compatible with field types used by NetFlow version 9 [RFC3954].

ID	Name	ID	Name
1	octetDeltaCount	43	RESERVED
2	packetDeltaCount	44	sourceIPv4Prefix
3	RESERVED	45	destinationIPv4Prefix
4	protocolIdentifier	46	mplsTopLabelType
5	classOfServiceIPv4	47	mplsTopLabelIPv4Address
6	tcpControlBits	48-51	RESERVED
7	sourceTransportPort	52	minimumTtl
8	sourceIPv4Address	53	maximumTtl
9	sourceIPv4Mask	54	identificationIPv4
10	ingressInterface	55	postClassOfServiceIPv4
11	destinationTransportPort	56	sourceMacAddress
12	destinationIPv4Address	57	postDestinationMacAddr
13	destinationIPv4Mask	58	vlanId
14	egressInterface	59	postVlanId
15	ipNextHopIPv4Address	60	ipVersion
16	bgpSourceAsNumber	61	RESERVED
17	bgpDestinationAsNumber	62	ipNextHopIPv6Address
18	bgpNexthopIPv4Address	63	bgpNexthopIPv6Address
19	postMCastPacketDeltaCount	64	ipv6ExtensionHeaders
20	postMCastOctetDeltaCount	65-69	RESERVED
21	flowEndSysUpTime	70	mplsTopLabelStackEntry
22	flowStartSysUpTime	71	mplsLabelStackEntry2
23	postOctetDeltaCount	72	mplsLabelStackEntry3
24	postPacketDeltaCount	73	mplsLabelStackEntry4
25	minimumPacketLength	74	mplsLabelStackEntry5
26	maximumPacketLength	75	mplsLabelStackEntry6
27	sourceIPv6Address	76	mplsLabelStackEntry7
28	destinationIPv6Address	77	mplsLabelStackEntry8
29	sourceIPv6Mask	78	mplsLabelStackEntry9
30	destinationIPv6Mask	79	mplsLabelStackEntry10
31	flowLabelIPv6	80	destinationMacAddress
32	icmpTypeCodeIPv4	81	postSourceMacAddress
33	igmpType	82-84	RESERVED
34-35	RESERVED	85	octetTotalCount
36	flowActiveTimeOut	86	packetTotalCount
37	flowInactiveTimeout	87	RESERVED
38-39	RESERVED	88	fragmentOffsetIPv4
40	exportedOctetTotalCount	89-127	RESERVED
41	exportedMessageTotalCount		
42	exportedFlowTotalCount		

The following list gives an overview of the Information Element identifiers that are specified in section 5 and extend the list of Information Element identifiers specified already in [RFC3954].

ID	Name	ID	Name
128	bgpNextAdjacentAsNumber	172	postPacketTotalCount
129	bgpPrevAdjacentAsNumber	173	flowKeyIndicator
130	exporterIPv4Address	174	postMCastPacketTotalCount
131	exporterIPv6Address	175	postMCastOctetTotalCount
132	droppedOctetDeltaCount	176	icmpTypeIPv4
133	droppedPacketDeltaCount	177	icmpCodeIPv4
134	droppedOctetTotalCount	178	icmpTypeIPv6
135	droppedPacketTotalCount	179	icmpCodeIPv6
136	flowEndReason	180	udpSourcePort
137	classOfServiceIPv6	181	udpDestinationPort
138	postClassOfServiceIPv6	182	tcpSourcePort
139	icmpTypeCodeIPv6	183	tcpDestinationPort
140	mplsTopLabelIPv6Address	184	tcpSequenceNumber
141	lineCardId	185	tcpAcknowledgementNumber
142	portId	186	tcpWindowSize
143	meteringProcessId	187	tcpUrgentPointer
144	exportingProcessId	188	tcpHeaderLength
145	templateId	189	ipHeaderLength
146	wlanChannelId	190	totalLengthIPv4
147	wlanSsid	191	payloadLengthIPv6
148	flowId	192	ipTimeToLive
149	sourceId	193	nextHeaderIPv6
150	flowStartSeconds	194	ipClassOfService
151	flowEndSeconds	195	ipDiffServCodePoint
152	flowStartMilliSeconds	196	ipPrecedence
153	flowEndMilliSeconds	197	fragmentFlagsIPv4
154	flowStartMicroSeconds	198	octetDeltaSumOfSquares
155	flowEndMicroSeconds	199	octetTotalSumOfSquares
156	flowStartNanoSeconds	200	mplsTopLabelTtl
157	flowEndNanoSeconds	201	mplsLabelStackLength
158	flowStartDeltaMicroSeconds	202	mplsLabelStackDepth
159	flowEndDeltaMicroSeconds	203	mplsTopLabelExp
160	systemInitTimeMilliSeconds	204	ipPayloadLength
161	flowDurationMilliSeconds	205	udpMessageLength
162	flowDurationMicroSeconds	206	isMulticast
163	observedFlowTotalCount	207	internetHeaderLengthIPv4
164	ignoredPacketTotalCount	208	ipv4Options
165	ignoredOctetTotalCount	209	tcpOptions
166	notSentFlowTotalCount	210	paddingOctets
167	notSentPacketTotalCount	211	
168	notSentOctetTotalCount	212	
169	destinationIPv6Prefix	213	headerLengthIPv4
170	sourceIPv6Prefix	214	mplsPayloadLength
171	postOctetTotalCount		

Information Element identifiers 211-212 are already in use by some implementations, but their descriptions were not agreed when this document was edited.

5. Information Elements

This section describes the Flow attributes of the IPFIX information model. The elements are grouped into 9 groups according to their semantics and their applicability:

1. Identifiers
2. Metering and Exporting Process Properties
3. IP Header Fields
4. Transport Header Fields
5. Sub-IP Header Fields
6. Derived Packet Properties
7. Min/Max Flow Properties
8. Flow Time Stamps
9. Per-Flow Counters
10. Miscellaneous Flow Properties

The Information Elements that are derived from fields of packets or from packet treatment, such as the Information Elements in groups 3.-6., can serve as Flow Keys used for mapping packets to Flows.

If they do not serve as Flow Keys, their value may change from packet to packet within a single Flow. For Information Elements with values that are derived from fields of packets or from packet treatment and for which the value may change from packet to packet within a single Flow, the IPFIX information model defines that their value is determined by the first packet observed for the corresponding Flow, unless the description of the Information Element explicitly specifies a different semantics. This simple rule allows writing all Information Elements related to header fields once when the first packet of the Flow is observed. For further observed packets of the same Flow, only Flow properties that depend on more than one packet, such as the Information Elements in groups 7.-9., need to be updated.

Information Elements with a name having the "post" prefix, for example, "postClassOfServiceIPv4", do not report properties that were actually observed at the Observation Point, but retrieved by other means within the Observation Domain. These information Elements can be used if there are middlebox functions within the Observation Domain changing Flow properties after packets passed the Observation Point.

5.1. Identifiers

Information Elements grouped in the table below are identifying components of the IPFIX architecture, of an IPFIX Device, or of the IPFIX protocol. All of them have an integral data type and data type semantics "identifier" as described in section 3.2.4.

Typically, some of them are used for limiting scopes of other Information Elements. However, also other Information Elements MAY be used for limiting scopes. Note also that all Information Elements listed below MAY be used for other purposes than limiting scopes.

ID	Name	ID	Name
141	lineCardId	143	meteringProcessId
142	portId	144	exportingProcessId
10	ingressInterface	148	flowId
14	egressInterface	145	templateId
		149	sourceId

5.1.1. lineCardId

Description:
A locally unique identifier of a line card at an IPFIX Device hosting an Observation Point. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 141
Status: current

5.1.2. portId

Description:
A locally unique identifier of a line port at an IPFIX Device hosting an Observation Point. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 142
Status: current

5.1.3. ingressInterface

Description:
The index of the IP interface (ifIndex) where packets of this Flow are being received.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 10
Status: current
Reference: See RFC 2863 for the definition of the ifIndex object.

5.1.4. egressInterface

Description:
The index of the IP interface (ifIndex) where packets of this Flow are being sent.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 14
Status: current
Reference: See RFC 2863 for the definition of the ifIndex object.

5.1.5. meteringProcessId

Description:
A locally unique identifier of a Metering Process at an IPFIX Device. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 143
Status: current

5.1.6. exportingProcessId

Description:
A locally unique identifier of an Exporting Process at an IPFIX Device. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 144
Status: current

5.1.7. flowId

Description:
An identifier of a Flow that is unique within an Observation Domain. This Information Element can be used to distinguish between different Flows if Flow Keys such as IP addresses and port numbers are not reported or reported in separate records.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 148
Status: current

5.1.8. templateId

Description:
An identifier of a Template that is locally unique to an Exporting Process. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 145
Status: current

5.1.9. sourceId

Description:
An identifier of an Observation Domain that is locally unique to an Exporting Process. Typically, this Information Element is used for limiting the scope of other Information Elements.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 149
Status: current

5.2. Metering and Exporting Process Properties

Information Elements in this section describe static and dynamic properties of the Metering Process and/or the Exporting Process. The set of these Information Elements is listed in the table below

ID	Name	ID	Name
130	exporterIPv4Address	164	ignoredPacketTotalCount
131	exporterIPv6Address	165	ignoredOctetTotalCount
41	exportedMessageTotalCount	166	notSentFlowTotalCount
40	exportedOctetTotalCount	167	notSentPacketTotalCount
42	exportedFlowTotalCount	168	notSentOctetTotalCount
163	observedFlowTotalCount	173	flowKeyIndicator

5.2.1. exporterIPv4Address

Description:
The IPv4 address used by the Exporting Process. This is used by the Collector to identify the Exporter in cases where the identity of the Exporter may have been obscured by the use of a proxy.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 130
Status: current

5.2.2. exporterIPv6Address

Description:
The IPv6 address used by the Exporting Process. This is used by the Collector to identify the Exporter in cases where the identity of the Exporter may have been obscured by the use of a proxy.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 131
Status: current

5.2.3. exportedMessageTotalCount

Description:
The total number of IPFIX Messages that the Exporting Process successfully sent since the Exporting Process (re-)initialization to the Collecting Process receiving a report that contains this Information Element.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 41
Status: current
Units: messages

5.2.4. exportedOctetTotalCount

Description:
The total number of octets that the Exporting Process successfully sent since the Exporting Process (re-)initialization to the Collecting Process receiving a report that contains this Information Element. The value of this Information Element is calculated by summing up the IPFIX Message header length values of all IPFIX Messages that were successfully sent to the Collecting Process receiving a report that contains this Information Element.

Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 40
Status: current
Units: octets

5.2.5. exportedFlowTotalCount

Description:
The total number of Flows Records that the Exporting Process successfully sent as Data Records since the Exporting Process (re-)initialization to the Collecting Process receiving a report that contains this Information Element.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 42
Status: current
Units: Flows

5.2.6. observedFlowTotalCount

Description:
The total number of Flows observed in the Observation Domain since the Metering Process (re-)initialization for this Observation Point.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 163
Status: current
Units: Flows

5.2.7. ignoredPacketTotalCount

Description:
The total number of observed IP packets that the Metering Process did not process since the (re-)initialization of the Metering Process.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 164
Status: current
Units: packets

5.2.8. ignoredOctetTotalCount

Description:
The total number of octets in observed IP packets that the Metering Process did not process since the (re-)initialization of the Metering Process.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 165
Status: current
Units: octets

5.2.9. notSentFlowTotalCount

Description:
The total number of Flow Records that were generated by the Metering Process and but dropped by the Metering Process or by the Exporting Process instead of sending it to the Collecting Process. There are several potential reasons for this including resource shortage and special Flow export policies.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 166
Status: current
Units: Flows

5.2.10. notSentPacketTotalCount

Description:
The total number of packets in Flow Records that were generated by the Metering Process and but dropped by the Metering Process or by the Exporting Process instead of sending it to the Collecting Process. There are several potential reasons for this including resource shortage and special Flow export policies.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 167
Status: current
Units: packets

5.2.11. notSentOctetTotalCount

Description:
The total number of octets in packets in Flow Records that were generated by the Metering Process and but dropped by the Metering Process or by the Exporting Process instead of sending it to the Collecting Process. There are several potential reasons for this including resource shortage and special Flow export policies.

Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 168
Status: current
Units: octets

5.2.12. flowKeyIndicator

Description:
This set of bit fields is used for marking the Information Elements of a Data Record that serve as Flow Key. Each bit represents an Information Element in the Data Record with the n-th bit representing the n-th Information Element. A set bit with value 1 indicates that the corresponding Information element is a Flow Key of the reported Flow. A value of 0 indicates that this is not the case. If the Data Record contains more than 64 Information Elements, the corresponding Template SHOULD be designed such that all Flow Keys are among the first 64 Information Elements, because the flowKeyIndicator only contains 64 bits. If the Data Record contains less than 64 Information Elements, then the bits in the flowKeyIndicator for which no corresponding Information Element exists SHOULD have the value 0.
Abstract Data Type: unsigned64
Data Type Semantics: flags
ElementId: 173
Status: current

5.3. IP Header Fields

Information Elements in this section indicate values of IP header fields or are derived from IP header field values in combination with further information.

ID	Name	ID	Name
60	ipVersion	195	ipDiffServCodePoint
8	sourceIPv4Address	196	ipPrecedence
27	sourceIPv6Address	5	classOfServiceIPv4
9	sourceIPv4Mask	55	postClassOfServiceIPv4
29	sourceIPv6Mask	137	classOfServiceIPv6
44	sourceIPv4Prefix	138	postClassOfServiceIPv6
170	sourceIPv6Prefix	31	flowLabelIPv6
12	destinationIPv4Address	206	isMulticast
28	destinationIPv6Address	54	identificationIPv4
13	destinationIPv4Mask	88	fragmentOffsetIPv4
30	destinationIPv6Mask	197	fragmentFlagsIPv4
45	destinationIPv4Prefix	189	ipHeaderLength
169	destinationIPv6Prefix	213	headerLengthIPv4
192	ipTimeToLive	207	internetHeaderLengthIPv4
4	protocolIdentifier	190	totalLengthIPv4
193	nextHeaderIPv6	191	payloadLengthIPv6
194	ipClassOfService	204	ipPayloadLength

5.3.1. ipVersion

Description: The IP version field in the IP packet header.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 60
Status: current
Reference:
See RFC 791 for a definition of the version field in the IPv4 packet header. See RFC 2460 for a definition of the version field in the IPv6 packet header. Additional information on defined version numbers can be found at http://www.iana.org/assignments/version-numbers.

5.3.2. sourceIPv4Address

Description:
The IPv4 source address in the IP packet header.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 8
Status: current

Reference: See RFC 791 for the definition of the IPv4 source address
field.

5.3.3. sourceIPv6Address

Description:
The IPv6 source address in the IP packet header.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 27
Status: current

5.3.4. sourceIPv4Mask

Description:
The number of contiguous bits that are relevant in the
sourceIPv4Prefix Information Element.
Abstract Data Type: octet
ElementId: 9
Status: current
Units: bits
Range: The valid range is 0-32.

5.3.5. sourceIPv6Mask

Description:
The number of contiguous bits that are relevant in the
sourceIPv6Prefix Information Element.
Abstract Data Type: octet
ElementId: 29
Status: current
Units: bits
Range: The valid range is 0-128.

5.3.6. sourceIPv4Prefix

Description:
IPv4 source address prefix.
Abstract Data Type: ipv4Address
ElementId: 44
Status: current

5.3.7. sourceIPv6Prefix

Description:
IPv6 source address prefix.
Abstract Data Type: ipv6Address
ElementId: 170
Status: current

5.3.8. destinationIPv4Address

Description:
The IPv4 destination address in the IP packet header.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 12
Status: current
Reference: See RFC 791 for the definition of the IPv4 destination address field.

5.3.9. destinationIPv6Address

Description:
The IPv6 destination address in the IP packet header.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 28
Status: current

5.3.10. destinationIPv4Mask

Description:
The number of contiguous bits that are relevant in the destinationIPv4Prefix Information Element.
Abstract Data Type: octet
ElementId: 13
Status: current
Units: bits
Range: The valid range is 0-32.

5.3.11. destinationIPv6Mask

Description:
The number of contiguous bits that are relevant in the destinationIPv6Prefix Information Element.
Abstract Data Type: octet
ElementId: 30

Status: current
Units: bits
Range: The valid range is 0-128.

5.3.12. destinationIPv4Prefix

Description:
IPv4 destination address prefix.
Abstract Data Type: ipv4Address
ElementId: 45
Status: current

5.3.13. destinationIPv6Prefix

Description:
IPv6 destination address prefix.
Abstract Data Type: ipv6Address
ElementId: 169
Status: current

5.3.14. ipTimeToLive

Description:
For IPv4, the value of the Information Element matches the value of the Time to Live field in the IPv4 packet header. For IPv6, the value of the Information Element matches the value of the Hop Limit field in the IPv6 packet header.
Abstract Data Type: octet
ElementId: 192
Status: current
Units: hops
Reference: See RFC 791 for the definition of the IPv4 Time to Live field. See RFC 2460 for the definition of the IPv6 Hop Limit field.

5.3.15. protocolIdentifier

Description:
The value of the protocol number in the IP packet header. The protocol number identifies the IP packet payload type. Protocol numbers are defined in the IANA Protocol Numbers registry.
In Internet Protocol version 4 (IPv4) this is carried in the "Protocol" field. In Internet Protocol version 6 (IPv6) this is carried in the "Next Header" field in the last extension header of the packet.

Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 4
Status: current
Reference:
See RFC 791 for the specification of the IPv4 protocol field. See RFC 2460 for the specification of the IPv6 protocol field. See the list of protocol numbers assigned by IANA at http://www.iana.org/assignments/protocol-numbers.

5.3.16. nextHeaderIPv6

Description:
The value of the Next Header field of the IPv6 header. The value identifies the type of the following IPv6 extension header or of the following IP payload. Valid values are defined in the IANA Protocol Numbers registry.
Abstract Data Type: octet
ElementId: 193
Status: current
Reference: See RFC 2460 for the definition of the IPv6 Next Header field. See the list of protocol numbers assigned by IANA at http://www.iana.org/assignments/protocol-numbers.

5.3.17. ipClassOfService

Description:
For IPv4 packets, this is the value of the TOS field in the IPv4 packet header. For IPv6 packets, this is the value of the Traffic Class field in the IPv6 packet header.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 194
Status: current
Reference: See section 5.3.2 of RFC 1812 and RFC 791 for the definition of the IPv4 TOS field. See RFC 2460 for the definition of the IPv6 Traffic Class field.

5.3.18. ipDiffServCodePoint

Description:
The value of a Differentiated Services Code Point (DSCP) encoded in the Differentiated Services Field. The Differentiated Services Field spans the most significant 6 bits of the IPv4 TOS field or the IPv6 Traffic class field, respectively.

This Information Element encodes only the 6 bits of the Differentiated Services field. Therefore its value may range from 0 to 63.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 195
Status: current
Range: The valid range is 0-63.
Reference: See RFC 3260 for the definition of the Differentiated Services Field. See section 5.3.2 of RFC 1812 and RFC 791 for the definition of the IPv4 TOS field. See RFC 2460 for the definition of the IPv6 Traffic Class field.

5.3.19. ipPrecedence

Description:
The value of the IP Precedence. The IP Precedence value is encoded in the first 3 bits of the IPv4 TOS field or the IPv6 Traffic class field, respectively.
This Information Element encodes only the 3 bits of the Differentiated Services field. Therefore its value may range from 0 to 7.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 196
Status: current
Range: The valid range is 0-7.
Reference: See section 5.3.3 of RFC 1812 and RFC 791 for the definition of the IP Precedence. See section 5.3.2 of RFC 1812 and RFC 791 for the definition of the IPv4 TOS field. See RFC 2460 for the definition of the IPv6 Traffic Class field.

5.3.20. classOfServiceIPv4

Description:
The value of the TOS field in the IPv4 packet header.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 5
Status: current
Reference: See RFC 791 for the definition of the IPv4 TOS field.

5.3.21. postClassOfServiceIPv4

Description:
The definition of this Information Element is identical to the definition of Information Element 'classOfServiceIPv4', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 55
Status: current
Reference: See RFC 791 for the definition of the IPv4 TOS field. See RFC 3234 for the definition of middleboxes.

5.3.22. classOfServiceIPv6

Description:
The value of the Traffic Class field in the IPv6 packet header.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 137
Status: current
Reference: See RFC 2460 for the definition of the IPv6 Traffic Class field.

5.3.23. postClassOfServiceIPv6

Description:
The definition of this Information Element is identical to the definition of Information Element 'classOfServiceIPv6', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 138
Status: current
Reference: See RFC 2460 for the definition of the IPv6 traffic class field.

5.3.24. flowLabelIPv6

Description:
The value of the IPv6 Flow Label field in the IP packet header.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 31

Status: current
Reference: See RFC 2460 for a definition of the flow label field in the IPv6 packet header.

5.3.25. isMulticast

Description:
If the IP destination address is a reserved multicast address then the value of this Information Element is not equal to zero. Otherwise, the value of all bits of the octet is zero.
The first bit of this octet is set to 1 if the Version field of the IP header has the value 4 and if the Destination Address field contains a reserved multicast address in the range from 224.0.0.0 to 239.255.255.255. Otherwise, this bit is set to 0.
The second and third bit of this octet are reserved for future use.
The remaining bits of the octet are only set to values other than zero if th IP Destination Address is a reserved IPv6 multicast address. Then the fourth bit of the octet is set to the value of the T flag in the IPv6 multicast address and the remaining four bits are set to the value of the scope field in the IPv6 multicast address.

```
    0      1      2      3      4      5      6      7
+------+------+------+------+------+------+------+------+
| MCv4 | RES. | RES. |  T   |   IPv6 multicast scope    |
+------+------+------+------+------+------+------+------+

Bit 0:    set to 1 if IPv4 multicast
Bit 1-2:  reserved for future use
Bit 4:    set to value of T-flag, if IPv6 multicast
Bit 4-7:  set to value of multicast scope if IPv6 multicast
```

Abstract Data Type: octet
Data Type Semantics: flags
ElementId: 206
Status: current
Reference: See RFC 1112 for the specification of reserved IPv4 multicast addresses. See RFC 3513 for the specification of reserved IPv6 multicast addresses and the definition of the T-flag and the IPv6 multicast scope.

5.3.26. identificationIPv4

Description:
The value of the IPv4 packet identification field in the IP packet header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 54
Status: current
Reference: See RFC 791 for the definition of the IPv4 identification field.

5.3.27. fragmentOffsetIPv4

Description:
The value of the IPv4 fragment offset field in the IP packet header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 88
Status: current
Reference:
See RFC 791 for the specification of the IPv4 fragment offset.

5.3.28. fragmentFlagsIPv4

Description:
The value of the fragmentation bits in the IPv4 packet header.

```
Bit 0:    reserved, must be zero.
Bit 1:    (DF) 0 = May Fragment,  1 = Don't Fragment.
Bit 2:    (MF) 0 = Last Fragment, 1 = More Fragments.
Bits 3-7: (DC) Don't Care, value is irrelevant.

    0   1   2   3   4   5   6   7
  +---+---+---+---+---+---+---+---+
  |   | D | M | D | D | D | D | D |
  | 0 | F | F | C | C | C | C | C |
  +---+---+---+---+---+---+---+---+
```

Abstract Data Type: octet
Data Type Semantics: flags
ElementId: 197
Status: current
Reference:
See RFC 791 for the specification of the IPv4 fragment flags.

5.3.29. ipHeaderLength

Description:
The length of the IP header. For IPv6, the value of this Information Element is 40.
Abstract Data Type: octet
ElementId: 189
Status: current
Units: octets
Reference:
See RFC 791 for the specification of the IPv4 header. See RFC 2460 for the specification of the IPv6 header.

5.3.30. headerLengthIPv4

Description:
The length of the IPv4 header.
Abstract Data Type: octet
ElementId: 213
Status: current
Units: octets
Reference:
See RFC 791 for the specification of the IPv4 header.

5.3.31. internetHeaderLengthIPv4

Description:
The value of the Internet Header Length (IHL) field in the IPv4 header. It specifies the length of the header in units of 4 octets.
Abstract Data Type: octet
ElementId: 207
Status: current
Units: 4 octets
Reference:
See RFC 791 for the specification of the IPv4 header.

5.3.32. totalLengthIPv4

Description:
The total length of the IPv4 packet.
Abstract Data Type: unsigned16
ElementId: 190
Status: current
Units: octets
Reference:

See RFC 791 for the specification of the IPv4 total length.

5.3.33. payloadLengthIPv6

Description:
The length of the IPv6 payload, i.e., the rest of the packet following the IPv6 header, in octets. Note that any extension headers present are considered part of the payload, i.e., included in the length count.
This Information Element reports the value of the Payload Length field in the IPv6 header except in the case that the value of this field is zero and that there is a valid jumbo payload option. Then the value of the Jumbo Payload Length field in the jumbo payload option is reported.
Abstract Data Type: unsigned32
ElementId: 191
Status: current
Reference:
See RFC 2460 for the specification of the IPv6 payload length.
See RFC 2675 for the specification of the IPv6 jumbo payload length.

5.3.34. ipPayloadLength

Description:
The effective length of the IP payload.
For IPv4 packets the value of this Information Element is the difference between the total length of the IPv4 packet (as reported by Information Element totalLengthIPv4) and the length of the IPv4 header (as reported by Information Element headerLengthIPv4).
For IPv6, the value of the Payload Length field in the IPv6 header is reported except in the case that the value of this field is zero and that there is a valid jumbo payload option. In this case the value of the Jumbo Payload Length field in the jumbo payload option is reported.
Abstract Data Type: unsigned64
ElementId: 204
Status: current
Reference:
See RFC 791 for the specification of IPv4 packets. See RFC 2460 for the specification of the IPv6 payload length. See RFC 2675 for the specification of the IPv6 jumbo payload length.

5.4. Transport Header Fields

The set of Information Elements related to transport header fields and length includes the Information Elements listed in the table

below.

ID	Name	ID	Name
7	sourceTransportPort	187	tcpUrgentPointer
11	destinationTransportPort	188	tcpHeaderLength
180	udpSourcePort	32	icmpTypeCodeIPv4
181	udpDestinationPort	176	icmpTypeIPv4
205	udpMessageLength	177	icmpCodeIPv4
182	tcpSourcePort	139	icmpTypeCodeIPv6
183	tcpDestinationPort	178	icmpTypeIPv6
184	tcpSequenceNumber	179	icmpCodeIPv6
185	tcpAcknowledgementNumber	33	igmpType
186	tcpWindowSize		

5.4.1. sourceTransportPort

Description:
The source port identifier in the transport header. For the transport protocols UDP, TCP and SCTP this is the source port number given in the respective header. This field MAY also be used for future transport protocols that have 16 bit source port identifiers.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 7
Status: current
Reference:
See RFC 768 for the definition of the UDP source port field. See RFC 793 for the definition of the TCP source port field. See RFC 2960 for the definition of SCTP.
Additional information on defined UDP and TCP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.2. destinationTransportPort

Description:
The destination port identifier in the transport header. For the transport protocols UDP, TCP and SCTP this is the destination port number given in the respective header. This field MAY also be used for future transport protocols that have 16 bit destination port identifiers.

Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 11
Status: current
Reference:
See RFC 768 for the definition of the UDP source port field. See RFC 793 for the definition of the TCP source port field. See RFC 2960 for the definition of SCTP.
Additional information on defined UDP and TCP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.3. udpSourcePort

Description: The source port identifier in the UDP header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 180
Status: current
Reference: See RFC 768 for the definition of the UDP source port field. Additional information on defined UDP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.4. udpDestinationPort

Description: The destination port identifier in the UDP header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 181
Status: current
Reference: See RFC 768 for the definition of the UDP source port field. Additional information on defined UDP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.5. udpMessageLength

Description: The value of the Length field in the UDP header.
Abstract Data Type: unsigned16
ElementId: 205
Status: current
Units: octets
Reference: See RFC 768 for the specification of the UDP header.

5.4.6. tcpSourcePort

Description: The source port identifier in the TCP header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 182
Status: current
Reference: See RFC 793 for the definition of the TCP source port field. Additional information on defined TCP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.7. tcpDestinationPort

Description: The destination port identifier in the TCP header.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 183
Status: current
Reference: See RFC 793 for the definition of the TCP source port field. Additional information on defined TCP port numbers can be found at http://www.iana.org/assignments/port-numbers.

5.4.8. tcpSequenceNumber

Description: The sequence number in the TCP header.
Abstract Data Type: unsigned32
ElementId: 184
Status: current
Reference: See RFC 793 for the definition of the TCP sequence number.

5.4.9. tcpAcknowledgementNumber

Description: The acknowledgement number in the TCP header.
Abstract Data Type: unsigned32
ElementId: 185
Status: current
Reference: See RFC 793 for the definition of the TCP acknowledgement number.

5.4.10. tcpWindowSize

Description: The window field in the TCP header.
Abstract Data Type: unsigned16
ElementId: 186
Status: current
Reference: See RFC 793 for the definition of the TCP window field.

5.4.11. tcpUrgentPointer

Description: The urgent pointer in the TCP header.
Abstract Data Type: unsigned16
ElementId: 187
Status: current
Reference: See RFC 793 for the definition of the TCP urgent pointer.

5.4.12. tcpHeaderLength

Description: The length of the TCP header.
Abstract Data Type: unsigned16
ElementId: 188
Status: current
Units: octets
Reference: See RFC 793 for the definition of the TCP header.

5.4.13. icmpTypeCodeIPv4

Description:
Type and Code of the IPv4 ICMP message. The combination of both values is reported as (ICMP type * 256) + ICMP code.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 32
Status: current
Reference: See RFC 792 for a definition of the IPv4 ICMP type and code fields.

5.4.14. icmpTypeIPv4

Description:
Type of the IPv4 ICMP message.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 176
Status: current
Reference: See RFC 792 for a definition of the IPv4 ICMP type field.

5.4.15. icmpCodeIPv4

Description:
Code of the IPv4 ICMP message.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 177

Status: current
Reference: See RFC 792 for a definition of the IPv4 ICMP code field.

5.4.16. icmpTypeCodeIPv6

Description:
Type and Code of the IPv6 ICMP message. The combination of both values is reported as (ICMP type * 256) + ICMP code.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 139
Status: current
Reference: See RFC 2463 for a definition of the IPv6 ICMP type and code fields.

5.4.17. icmpTypeIPv6

Description:
Type of the IPv6 ICMP message.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 178
Status: current
Reference: See RFC 2463 for a definition of the IPv6 ICMP type field.

5.4.18. icmpCodeIPv6

Description:
Code of the IPv6 ICMP message.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 179
Status: current
Reference: See RFC 2463 for a definition of the IPv6 ICMP code field.

5.4.19. igmpType

Description: The type field of the IGMP message.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 33
Status: current
Reference: See RFC 2236 for a definition of the IGMP type field.

5.5. Sub-IP Header Fields

The set of Information Elements related to Sub-IP header fields includes the Information Elements listed in the table below.

ID	Name	ID	Name
56	sourceMacAddress	214	mplsPayloadLength
81	postSourceMacAddress	70	mplsTopLabelStackEntry
58	vlanId	71	mplsLabelStackEntry2
59	postVlanId	72	mplsLabelStackEntry3
80	destinationMacAddress	73	mplsLabelStackEntry4
57	postDestinationMacAddr	74	mplsLabelStackEntry5
146	wlanChannelId	75	mplsLabelStackEntry6
147	wlanSsid	76	mplsLabelStackEntry7
200	mplsTopLabelTtl	77	mplsLabelStackEntry8
203	mplsTopLabelExp	78	mplsLabelStackEntry9
202	mplsLabelStackDepth	79	mplsLabelStackEntry10
201	mplsLabelStackLength		

5.5.1. sourceMacAddress

Description:
The IEEE 802 source MAC address field.
Abstract Data Type: macAddress
Data Type Semantics: identifier
ElementId: 56
Status: current
Reference: See IEEE.802-3.2002.

5.5.2. postSourceMacAddress

Description:
The definition of this Information Element is identical to the definition of Information Element 'sourceMacAddress', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: macAddress
Data Type Semantics: identifier
ElementId: 81
Status: current
Reference: See IEEE.802-3.2002.

5.5.3. vlanId

Description:
The IEEE 802.1Q VLAN identifier (VID) extracted from the Tag Control Information field that was attached to the IP packet.

Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 58
Status: current
Reference: See IEEE.802-1Q.2003.

5.5.4. postVlanId

Description:
The definition of this Information Element is identical to the definition of Information Element 'vlanId', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 59
Status: current
Reference: See IEEE.802-1Q.2003.

5.5.5. destinationMacAddress

Description:
The IEEE 802 destination MAC address field.
Abstract Data Type: macAddress
Data Type Semantics: identifier
ElementId: 80
Status: current
Reference: See IEEE.802-3.2002.

5.5.6. postDestinationMacAddr

Description:
The definition of this Information Element is identical to the definition of Information Element 'destinationMacAddress', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: macAddress
Data Type Semantics: identifier
ElementId: 57
Status: current
Reference: See IEEE.802-3.2002.

5.5.7. wlanChannelId

Description:
The identifier of the 802.11 (Wi-Fi) channel used.
Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 146
Status: current
Reference: See IEEE.802-11.1999.

5.5.8. wlanSsid

Description:
The Service Set IDentifier (SSID) identifying an 802.11 (Wi-Fi) network used. According to IEEE.802-11.1999 the SSID is encoded into a string of up to 32 characters.
Abstract Data Type: string
ElementId: 147
Status: current
Reference: See IEEE.802-11.1999.

5.5.9. mplsTopLabelTtl

Description: The TTL field from the top MPLS label stack entry, i.e. the last label that was pushed.
Abstract Data Type: unsigned32
ElementId: 200
Status: current
Reference: See RFC 3032 for the specification of the TTL field.

5.5.10. mplsTopLabelExp

Description:
The Exp field from the top MPLS label stack entry, i.e. the last label that was pushed.

```
Bit 0-4:  Don't Care, value is irrelevant.
Bit 5-7:  MPLS Exp field

  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
|     don't care    |    Exp    |
+---+---+---+---+---+---+---+---+
```

Abstract Data Type: octet
Data Type Semantics: flags

ElementId: 203
Status: current
Reference: See RFC 3032 for the specification of the Exp field. See
RFC 3270 for usage of the Exp field.

5.5.11. mplsLabelStackDepth

Description: The number of labels in the MPLS label stack.
Abstract Data Type: unsigned32
ElementId: 202
Status: current
Units: label stack entries
Reference: See RFC 3032 for the specification of the MPLS label
stack.

5.5.12. mplsLabelStackLength

Description: The length of the MPLS label stack in units of octets.
Abstract Data Type: unsigned32
ElementId: 201
Status: current
Units: octets
Reference: See RFC 3032 for the specification of the MPLS label
stack.

5.5.13. mplsPayloadLength

Description: The size of the MPLS packet without the label stack.
Abstract Data Type: unsigned32
ElementId: 214
Status: current
Units: octets
Reference: See RFC 3031 for the specification of MPLS packets. See
RFC 3032 for the specification of the MPLS label stack.

5.5.14. mplsTopLabelStackEntry

Description:
The label, exp and s fields from the top MPLS label stack entry,
i.e. the last label that was pushed.

```
       0                   1                   2
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                Label                  | Exp |S|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      Label:  Label Value, 20 bits
      Exp:    Experimental Use, 3 bits
      S:      Bottom of Stack, 1 bit
```

Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 70
Status: current
Reference: See RFC 3032.

5.5.15. mplsLabelStackEntry2

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsTopLabelStackEntry. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 71
Status: current
Reference: See RFC 3032.

5.5.16. mplsLabelStackEntry3

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry2. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 72
Status: current
Reference: See RFC 3032.

5.5.17. mplsLabelStackEntry4

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry3. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 73
Status: current
Reference: See RFC 3032.

5.5.18. mplsLabelStackEntry5

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry4. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 74
Status: current
Reference: See RFC 3032.

5.5.19. mplsLabelStackEntry6

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry5. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 75
Status: current
Reference: See RFC 3032.

5.5.20. mplsLabelStackEntry7

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry6. See the definition of mplsTopLabelStackEntry for further details.

Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 76
Status: current
Reference: See RFC 3032.

5.5.21. mplsLabelStackEntry8

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry7. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 77
Status: current
Reference: See RFC 3032.

5.5.22. mplsLabelStackEntry9

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry8. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 78
Status: current
Reference: See RFC 3032.

5.5.23. mplsLabelStackEntry10

Description:
The label, exp, and s fields from the label stack entry that was pushed immediately before the label stack entry that would be reported by mplsLabelStackEntry9. See the definition of mplsTopLabelStackEntry for further details.
Abstract Data Type: unsigned32
Data Type Semantics: identifier
ElementId: 79
Status: current
Reference: See RFC 3032.

5.6. Derived Packet Properties

The set of Information Elements derived from values of header fields

and further information includes the Information Elements listed in the table below.

ID	Name	ID	Name
15	ipNextHopIPv4Address	18	bgpNextHopIPv4Address
62	ipNextHopIPv6Address	63	bgpNextHopIPv6Address
16	bgpSourceAsNumber	46	mplsTopLabelType
17	bgpDestinationAsNumber	47	mplsTopLabelIPv4Address
128	bgpNextAdjacentAsNumber	140	mplsTopLabelIPv6Address
129	bgpPrevAdjacentAsNumber		

5.6.1. ipNextHopIPv4Address

Description:
The IPv4 address of the next IPv4 hop.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 15
Status: current

5.6.2. ipNextHopIPv6Address

Description:
The IPv6 address of the next IPv6 hop.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 62
Status: current

5.6.3. bgpSourceAsNumber

Description:
The autonomous system (AS) number of the source IP address. If AS path information for this Flow is only available as unordered AS set (and not as ordered AS sequence), then the value of this Information Element is 0.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 16
Status: current
Reference: See RFC 1771 for a description of BGP-4 and see RFC 1930 for a definition of the AS number.

5.6.4. bgpDestinationAsNumber

Description:
The autonomous system (AS) number of the destination IP address. If AS path information for this Flow is only available as unordered AS set (and not as ordered AS sequence), then the value of this Information Element is 0.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 17
Status: current
Reference: See RFC 1771 for a description of BGP-4 and see RFC 1930 for a definition of the AS number.

5.6.5. bgpNextAdjacentAsNumber

Description:
The autonomous system (AS) number of the first AS in the AS path to the destination IP address. The path is deduced by looking up the destination IP address of the Flow in the BGP routing information base. If AS path information for this Flow is only available as unordered AS set (and not as ordered AS sequence), then the value of this Information Element is 0.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 128
Status: current
Reference: See RFC 1771 for a description of BGP-4 and see RFC 1930 for a definition of the AS number.

5.6.6. bgpPrevAdjacentAsNumber

Description:
The autonomous system (AS) number of the last AS in the AS path from the source IP address. The path is deduced by looking up the source IP address of the Flow in the BGP routing information base. If AS path information for this Flow is only available as unordered AS set (and not as ordered AS sequence), then the value of this Information Element is 0. In case of BGP asymmetry, the bgpPrevAdjacentAsNumber might not be able to report the correct value.
Abstract Data Type: unsigned16
Data Type Semantics: identifier
ElementId: 129

Status: current
Reference: See RFC 1771 for a description of BGP-4 and see RFC 1930
for a definition of the AS number.

5.6.7. bgpNextHopIPv4Address

Description:
The IPv4 address of the next (adjacent) BGP hop.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 18
Status: current
Reference: See RFC 1771 for a description of BGP-4 and

5.6.8. bgpNextHopIPv6Address

Description:
The IPv6 address of the next (adjacent) BGP hop.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 63
Status: current
Reference: See RFC 1771 for a description of BGP-4.

5.6.9. mplsTopLabelType

Description:
This field identifies the control protocol that allocated the top
of stack label. Defined values for this field include:

- 0x01 TE-MIDPT: Any TE tunnel mid-point or tail label
- 0x02 Pseudowire: Any PWE3 or Cisco AToM based label
- 0x03 VPN: Any label associated with VPN
- 0x04 BGP: Any label associated with BGP or BGP routing
- 0x05 LDP: Any label associated with dynamically assigned labels using LDP

Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 46
Status: current
Reference: See RFC 3031 for the MPLS label structure. See RFC 2547
for the association of MPLS labels with VPNs. See RFC 1771 for
BGP and BGP routing. See RFC 3036 for LDP. and IP addresses.

5.6.10. mplsTopLabelIPv4Address

Description:
The IPv4 address of the system that the MPLS top label will cause this Flow to be forwarded to.
Abstract Data Type: ipv4Address
Data Type Semantics: identifier
ElementId: 47
Status: current
Reference: See RFC 3031 for the association between MPLS labels and IP addresses.

5.6.11. mplsTopLabelIPv6Address

Description:
The IPv6 address of the system that the MPLS top label will cause this Flow to be forwarded to.
Abstract Data Type: ipv6Address
Data Type Semantics: identifier
ElementId: 140
Status: current
Reference: See RFC 3031 for the association between MPLS labels and IP addresses.

5.7. Min/Max Flow Properties

Information Elements in this section are results of minimum or maximum operations over all packets of a Flow.

ID	Name	ID	Name
25	minimumPacketLength	208	ipv4Options
26	maximumPacketLength	64	ipv6ExtensionHeaders
52	minimumTtl	6	tcpControlBits
53	maximumTtl	209	tcpOptions

5.7.1. minimumPacketLength

Description:
Length of the smallest packet observed for this Flow.
Abstract Data Type: unsigned16
ElementId: 25
Status: current
Units: octets

5.7.2. maximumPacketLength

Description:
Length of the largest packet observed for this Flow.
Abstract Data Type: unsigned16
ElementId: 26
Status: current
Units: octets

5.7.3. minimumTtl

Description:
Minimum TTL value observed for any packet in this Flow.
Abstract Data Type: octet
ElementId: 52
Status: current

5.7.4. maximumTtl

Description:
Maximum TTL value observed for any packet in this Flow.
Abstract Data Type: octet
ElementId: 53
Status: current

5.7.5. ipv4Options

Description:
IPv4 options in packets of this Flow. The information is encoded in a set of bit fields. For each valid IPv4 option type there is a bit in this set. The bit is set to 1 if any observed packet of this Flow contains the corresponding IPv4 option type. Otherwise, if no observed packet of this Flow contained the respective IPv4 option type, the value of the corresponding bit is 0.
The list of valid IPv4 options is maintained by IANA. Note that for identifying an option not just the 5-bit Option Number, but all 8 bits of the Option Type need to match one of the IPv4 options specified at
http://www.iana.org/assignments/ip-parameters.
Options are mapped to bits according to their option numbers. Option number X is mapped to bit X. The mapping is illustrated by the figure below.

```
    0      1      2      3      4      5      6      7
+------+------+------+------+------+------+------+------+
| EOOL | NOP  | SEC  | LSR  |  TS  |E-SEC |CIPSO |  RR  | ...
+------+------+------+------+------+------+------+------+

    8      9     10     11     12     13     14     15
+------+------+------+------+------+------+------+------+
```

```
... | SID  | SSR  | ZSU  | MTUP | MTUR | FINN | VISA |ENCODE| ...
    +------+------+------+------+------+------+------+------+

       16     17     18     19     20     21     22     23
    +------+------+------+------+------+------+------+------+
... |IMITD | EIP  |  TR  |ADDEXT|RTRALT| SDB  |NSAPA | DPS  | ...
    +------+------+------+------+------+------+------+------+

       24     25     26     27     28     29     30     31
    +------+------+------+------+------+------+------+------+
... | UMP  |             to be assigned by IANA              |
    +------+------+------+------+------+------+------+------+
```

Bit	Type Value	Option Name	Reference
0	0	EOOL	End of Options List, RFC 791
1	1	NOP	No Operation, RFC 791
2	130	SEC	Security, RFC 1108
3	131	LSR	Loose Source Route, RFC 791
4	68	TS	Time Stamp, RFC 791
5	133	E-SEC	Extended Security, RFC 1108
6	134	CIPSO	Commercial Security
7	7	RR	Record Route, RFC 791
8	136	SID	Stream ID, RFC 791
9	137	SSR	Strict Source Route, RFC 791
10	10	ZSU	Experimental Measurement
11	11	MTUP	(obsoleted) MTU Probe, RFC 1191
12	12	MTUR	(obsoleted) MTU Reply, RFC 1191
13	205	FINN	Experimental Flow Control
14	142	VISA	Experimental Access Control
15	15	ENDOCE	
16	144	IMITD	IMI Traffic Descriptor
17	145	EIP	Extended Internet Protocol, RFC 1385
18	82	TR	Traceroute, RFC 3193
19	147	ADDEXT	Address Extension
20	148	RTRALT	Router Alert, RFC 2113
21	149	SDB	Selective Directed Broadcast
22	150	NSAPA	NSAP Address
23	151	DPS	Dynamic Packet State
24	152	UMP	Upstream Multicast Pkt.
...	...	...	Further options numbers may be assigned by IANA

Abstract Data Type: unsigned32
Data Type Semantics: flags
ElementId: 208
Status: current
Reference: See RFC 791 for the definition of IPv4 options. See the list of IPv4 option numbers assigned by IANA at http://www.iana.org/assignments/ip-parameters.

5.7.6. ipv6ExtensionHeaders

Description:
IPv6 extension headers observed in packets of this Flow. The information is encoded in a set of bit fields. For each IPv6 option header there is a bit in this set. The bit is set to 1 if any observed packet of this Flow contains the corresponding IPv6 extension header. Otherwise, if no observed packet of this Flow contained the respective IPv6 extension header, the value of the corresponding bit is 0.

```
            0     1     2     3     4     5     6     7
        +-----+-----+-----+-----+-----+-----+-----+-----+
        | Res | FRA1| ROU | FRA0| UNK | Res | HOP | DST |  ...
        +-----+-----+-----+-----+-----+-----+-----+-----+

            8     9    10    11    12    13    14    15
        +-----+-----+-----+-----+-----+-----+-----+-----+
    ... | PAY | AUT | ENC |         Reserved            | ...
        +-----+-----+-----+-----+-----+-----+-----+-----+

           16    17    18    19    20    21    22    23
        +-----+-----+-----+-----+-----+-----+-----+-----+
    ... |                  Reserved                     | ...
        +-----+-----+-----+-----+-----+-----+-----+-----+

           24    25    26    27    28    29    30    31
        +-----+-----+-----+-----+-----+-----+-----+-----+
    ... |                  Reserved                     |
        +-----+-----+-----+-----+-----+-----+-----+-----+

      Bit    IPv6 Option   Description

     0, Res               Reserved
     1, FRA1     44       Fragmentation header - not first fragment
     2, ROU      43       Routing header
     3, FRA0     44       Fragment header - first fragment
     4, UNK               Unknown Layer 4 header
                          (compressed, encrypted, not supported)
     5, Res               Reserved
     6, HOP       0       Hop-by-hop option header
     7, DST      60       Destination option header
     8, PAY     108       Payload compression header
     9, AUT      51       Authentication Header
    10, ENC      50       Encrypted security payload
    11 to 31              Reserved
```

Abstract Data Type: unsigned32
Data Type Semantics: flags
ElementId: 64
Status: current
Reference: See RFC 2460 for the general definition of IPv6 extensions headers and for the specification of the hop-by-hop options header, the routing header, the fragment header, and the destination options header. See RFC 2402 for the specification of the authentication header. See RFC 2406 for the specification of the encapsulating security payload.

5.7.7. tcpControlBits

Description:
TCP control bits observed for packets of this Flow. The information is encoded in a set of bit fields. For each TCP control bit there is a bit in this set. A bit is set to 1 if any observed packet of this Flow has the corresponding TCP control bit set to 1. A value of 0 for a bit indicates that the corresponding bit was not set in any of the observed packets of this Flow.

```
    0     1     2     3     4     5     6     7
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Reserved | URG | ACK | PSH | RST | SYN | FIN |
+-----+-----+-----+-----+-----+-----+-----+-----+

Reserved:  Reserved for future use by TCP.  Must be zero.
     URG:  Urgent Pointer field significant
     ACK:  Acknowledgment field significant
     PSH:  Push Function
     RST:  Reset the connection
     SYN:  Synchronize sequence numbers
     FIN:  No more data from sender
```

Abstract Data Type: octet
Data Type Semantics: flags
ElementId: 6
Status: current
Reference: See RFC 793 for a definition of the TCP control bits in the TCP header.

5.7.8. tcpOptions

Description:
TCP options in packets of this Flow. The information is encoded in a set of bit fields. For each TCP option there is a bit in this set. The bit is set to 1 if any observed packet of this Flow contains the corresponding TCP option. Otherwise, if no observed packet of this Flow contained the respective TCP option, the value of the corresponding bit is 0.
Options are mapped to bits according to their option numbers. Option number X is mapped to bit X. TCP option numbers are maintained by IANA.
Abstract Data Type: unsigned64
Data Type Semantics: flags

ElementId: 209
Status: current
Reference: See RFC 793 for the definition of TCP options. See the list of TCP option numbers assigned by IANA at http://www.iana.org/assignments/tcp-parameters.

5.8. Flow Time Stamps

Information Elements in this section are time stamps of events.

Time stamps flowStartSeconds, flowEndSeconds, flowStartMilliSeconds, flowEndMilliSeconds, flowStartMicroSeconds, flowEndMicroSeconds, flowStartNanoSeconds, flowEndNanoSeconds, and systemInitTimeMilliSeconds are absolute and have a well defined fixed time base, such as, for example, the number of seconds since 0000 UTC Jan 1st 1970.

Time stamps flowStartDeltaMicroSeconds and flowEndDeltaMicroSeconds are relative time stamps only valid within the scope of a single IPFIX Message. They contain the negative time offsets relative to the export time specified in the IPFIX Message header.

Time stamps flowStartSysUpTime and flowEndSysUpTime are relative time stamps indicating the time relative to the last (re-)initialization of the IPFIX Device. For reporting the time of the last (re-)initialization, systemInitTimeMilliSeconds can be reported, for example, in Data Records defined by Option Templates.

ID	Name	ID	Name
150	flowStartSeconds	156	flowStartNanoSeconds
151	flowEndSeconds	157	flowEndNanoSeconds
152	flowStartMilliSeconds	158	flowStartDeltaMicroSeconds
153	flowEndMilliSeconds	159	flowEndDeltaMicroSeconds
154	flowStartMicroSeconds	160	systemInitTimeMilliSeconds
155	flowEndMicroSeconds	22	flowStartSysUpTime
		21	flowEndSysUpTime

5.8.1. flowStartSeconds

Description: The absolute timestamp of the first packet of this Flow.
Abstract Data Type: dateTimeSeconds

ElementId: 150
Status: current
Units: seconds

5.8.2. flowEndSeconds

Description: The absolute timestamp of the last packet of this Flow.
Abstract Data Type: dateTimeSeconds
ElementId: 151
Status: current
Units: seconds

5.8.3. flowStartMilliSeconds

Description: The absolute timestamp of the first packet of this Flow.
Abstract Data Type: dateTimeMilliSeconds
ElementId: 152
Status: current
Units: milliseconds

5.8.4. flowEndMilliSeconds

Description: The absolute timestamp of the last packet of this Flow.
Abstract Data Type: dateTimeMilliSeconds
ElementId: 153
Status: current
Units: milliseconds

5.8.5. flowStartMicroSeconds

Description: The absolute timestamp of the first packet of this Flow.
Abstract Data Type: dateTimeMicroSeconds
ElementId: 154
Status: current
Units: microseconds

5.8.6. flowEndMicroSeconds

Description: The absolute timestamp of the last packet of this Flow.
Abstract Data Type: dateTimeMicroSeconds
ElementId: 155
Status: current
Units: microseconds

5.8.7. flowStartNanoSeconds

Description: The absolute timestamp of the first packet of this Flow.
Abstract Data Type: dateTimeNanoSeconds
ElementId: 156
Status: current
Units: nanoseconds

5.8.8. flowEndNanoSeconds

Description: The absolute timestamp of the last packet of this Flow.
Abstract Data Type: dateTimeNanoSeconds
ElementId: 157
Status: current
Units: nanoseconds

5.8.9. flowStartDeltaMicroSeconds

Description: This is a relative time stamp only valid within the scope of a single IPFIX Message. It contains the negative time offset of the first observed packet of this Flow relative to the export time specified in the IPFIX Message header.
Abstract Data Type: unsigned32
ElementId: 158
Status: current
Units: microseconds
Reference: See [I-D.ietf-ipfix-protocol] for the definition of the IPFIX Message header.

5.8.10. flowEndDeltaMicroSeconds

Description: This is a relative time stamp only valid within the scope of a single IPFIX Message. It contains the negative time offset of the last observed packet of this Flow relative to the export time specified in the IPFIX Message header.
Abstract Data Type: unsigned32
ElementId: 159
Status: current
Units: microseconds
Reference: See [I-D.ietf-ipfix-protocol] for the definition of the IPFIX Message header.

5.8.11. systemInitTimeMilliSeconds

Description: The absolute timestamp of the last (re-)initialization of the IPFIX Device.

Abstract Data Type: dateTimeMilliSeconds
ElementId: 160
Status: current
Units: milliseconds

5.8.12. flowStartSysUpTime

Description: The relative timestamp of the first packet of this Flow. It indicates the number of milliseconds since the last (re-)initialization of the IPFIX Device (sysUpTime).
Abstract Data Type: unsigned32
ElementId: 22
Status: current
Units: milliseconds

5.8.13. flowEndSysUpTime

Description: The relative timestamp of the last packet of this Flow. It indicates the number of milliseconds since the last (re-)initialization of the IPFIX Device (sysUpTime).
Abstract Data Type: unsigned32
ElementId: 21
Status: current
Units: milliseconds

5.9. Per-Flow Counters

Information Elements in this section are counters all having integer values. Their values may change for every report they are used in. They cannot serve as part of a Flow Key used for mapping packets to Flows. However, potentially they can be used for selecting exported Flows, for example, by only exporting Flows with more than a threshold number of observed octets.

There are running counters and delta counters. Delta counters are reset to zero each time their values are exported. Running counters continue counting independently of the Exporting Process.

There are per-Flow counters and counters related to the Metering Process and/or the Exporting Process. Per-Flow counters are Flow properties that potentially change each time a packet belonging to the Flow is observed. The set of per-Flow counters includes the Information Elements listed in the table below.

ID	Name	ID	Name
1	octetDeltaCount	132	droppedOctetDeltaCount
23	postOctetDeltaCount	133	droppedPacketDeltaCount
198	octetDeltaSumOfSquares	134	droppedOctetTotalCount
85	octetTotalCount	135	droppedPacketTotalCount
171	postOctetTotalCount	19	postMCastPacketDeltaCount
199	octetTotalSumOfSquares	20	postMCastOctetDeltaCount
2	packetDeltaCount	174	postMCastPacketTotalCount
24	postPacketDeltaCount	175	postMCastOctetTotalCount
86	packetTotalCount		
172	postPacketTotalCount		

5.9.1. octetDeltaCount

Description:
The number of octets since the previous report (if any) in incoming packets for this Flow at the Observation Point. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 1
Status: current
Units: octets

5.9.2. postOctetDeltaCount

Description:
The definition of this Information Element is identical to the definition of Information Element 'octetDeltaCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 23
Status: current
Units: octets

5.9.3. octetDeltaSumOfSquares

Description:
The sum of the squared numbers of octets per incoming packet since the previous report (if any) for this Flow at the Observation Point. The number of octets include IP header(s) and IP payload.

Abstract Data Type: unsigned64
ElementId: 198
Status: current

5.9.4. octetTotalCount

Description:
The total number of octets in incoming packets for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 85
Status: current
Units: octets

5.9.5. postOctetTotalCount

Description:
The definition of this Information Element is identical to the definition of Information Element 'octetTotalCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 171
Status: current
Units: octets

5.9.6. octetTotalSumOfSquares

Description:
The total sum of the squared numbers of octets in incoming packets for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
ElementId: 199
Status: current
Units: octets

5.9.7. packetDeltaCount

Description:
The number of incoming packets since the previous report (if any) for this Flow at the Observation Point.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 2
Status: current
Units: packets

5.9.8. postPacketDeltaCount

Description:
The definition of this Information Element is identical to the definition of Information Element 'packetDeltaCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 24
Status: current
Units: packets

5.9.9. packetTotalCount

Description:
The total number of incoming packets for this Flow at the Observation Point since the Metering Process (re-)initialization for this Observation Point.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 86
Status: current
Units: packets

5.9.10. postPacketTotalCount

Description:
The definition of this Information Element is identical to the definition of Information Element 'packetTotalCount', except that it reports a potentially modified value caused by a middlebox function after the packet passed the observation point.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 172

Status: current
Units: packets

5.9.11. droppedOctetDeltaCount

Description:
The number of octets since the previous report (if any) in packets of this Flow dropped by packet treatment. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 132
Status: current
Units: octets

5.9.12. droppedPacketDeltaCount

Description:
The number of packets since the previous report (if any) of this Flow dropped by packet treatment.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 133
Status: current
Units: packets

5.9.13. droppedOctetTotalCount

Description:
The total number of octets in packets of this Flow dropped by packet treatment since the Metering Process (re-)initialization for this Observation Point. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 134
Status: current
Units: octets

5.9.14. droppedPacketTotalCount

Description:
The number of packets of this Flow dropped by packet treatment since the Metering Process (re-)initialization for this Observation Point.

Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 135
Status: current
Units: packets

5.9.15. postMCastPacketDeltaCount

Description:
The number of outgoing multicast packets since the previous report (if any) sent for packets of this Flow by a multicast daemon within the Observation Domain. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 19
Status: current
Units: packets

5.9.16. postMCastOctetDeltaCount

Description:
The number of octets since the previous report (if any) in outgoing multicast packets sent for packets of this Flow by a multicast daemon within the Observation Domain. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: deltaCounter
ElementId: 20
Status: current
Units: octets

5.9.17. postMCastPacketTotalCount

Description:
The total number of outgoing multicast packets sent for packets of this Flow by a multicast daemon within the Observation Domain since the Metering Process (re-)initialization. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means.
Abstract Data Type: unsigned64

Data Type Semantics: totalCounter
ElementId: 174
Status: current
Units: packets

5.9.18. postMCastOctetTotalCount

Description:
The total number of octets in outgoing multicast packets sent for packets of this Flow by a multicast daemon in the Observation Domain since the Metering Process (re-)initialization. This property cannot necessarily be observed at the Observation Point, but may be retrieved by other means. The number of octets include IP header(s) and IP payload.
Abstract Data Type: unsigned64
Data Type Semantics: totalCounter
ElementId: 175
Status: current
Units: octets

5.10. Miscellaneous Flow Properties

Information Elements in this section describe properties of Flows that are related to Flow start, Flow duration and Flow termination, but they are no time stamps as Information Elements in section 5.8.

ID	Name	ID	Name
36	flowActiveTimeOut	161	flowDurationMilliSeconds
37	flowInactiveTimeout	162	flowDurationMicroSeconds
136	flowEndReason		

5.10.1. flowActiveTimeOut

Description:
The number of seconds after which an active Flow is timed out anyway, even if there is still a continuous flow of packets.
Abstract Data Type: unsigned16
ElementId: 36
Status: current
Units: seconds

5.10.2. flowInactiveTimeout

Description:
A Flow is considered to be timed out if no packets belonging to the Flow have been observed for the number of seconds specified by this field.
Abstract Data Type: unsigned16
ElementId: 37
Status: current
Units: seconds

5.10.3. flowEndReason

Description:
The reason for Flow termination. The range of values includes

0x01: idle timeout
The flow was terminated because it was considered to be idle.
0x02: active timeout
The flow was terminated for reporting purposes while it was still active, for example, after the maximum lifetime of unreported flows was reached.
0x03: end of Flow detected
The flow was terminated because the Metering Process detected signals indicating the end of the flow, for example, the TCP FIN flag.
0x04: forced end
The flow was terminated because of some external event, for example, a shut down of the Metering Process initiated by a network management application.
0x05: cache full
the flow was terminated because of lack of resources available to the Metering Process and/or the Exporting Process

Abstract Data Type: octet
Data Type Semantics: identifier
ElementId: 136
Status: current

5.10.4. flowDurationMilliSeconds

Description: The difference between in time between the observation of the first packet of this Flow and the observation of the last packet of this Flow.

Abstract Data Type: unsigned32
ElementId: 161
Status: current
Units: milliseconds

5.10.5. flowDurationMicroSeconds

Description: The difference between in time between the observation of the first packet of this Flow and the observation of the last packet of this Flow.
Abstract Data Type: unsigned32
ElementId: 162
Status: current
Units: microseconds

5.11. Padding

This section contains a single Information Element only, that can be used for padding of Flow Records.

IPFIX Implementations may wish to align Information Elements within Data Records or to align entire Data Records to 4 octet or 8 octet boundaries. This can be achieved by including one or more paddingOctets Information Elements in a Data Record.

ID	Name	ID	Name
210	paddingOctets		

5.11.1. paddingOctets

Description:
The value of this Information Element is always 0.
Abstract Data Type: octetArray
ElementId: 210
Status: current

6. Extending the Information Model

A key requirement for IPFIX is to allow for extending the set of Information Elements which are reported. This section defines the mechanism for extending this set.

Extension is done by defining new Information Elements. Each new Information Element MUST be assigned a unique Information Element

identifier as part of its definition. These unique Information Element identifiers are the connection between the record structure communicated by the protocol using templates and a consuming application. For generally applicable Information Elements using IETF and IANA mechanisms for extending the information model is recommended.

Names of new Information Elements SHOULD be chosen according to the naming conventions given in section 2.3.

For extensions, the type space defined in section 3 can be used. If required, new data types can be added. New data types SHOULD be defined in IETF standards track documents.

Enterprises may wish to define Information Elements without registering them with IANA. IPFIX explicitly supports enterprise-specific Information Elements. Enterprise-specific Information Elements as described in sections 2.1 and 4.

However, before creating enterprise-specific Information Elements, the general applicability of such Information Elements should be considered. IPFIX does not support enterprise-specific data types.

7. IANA Considerations

This documents defines an initial set of IPFIX Information Elements. For extending them in the future, IANA needs to create a new registry for IPFIX Information Element identifiers.

New assignments for IPFIX Information Elements will be administered by IANA, on a First Come First Served basis [RFC2434], subject to Expert Review [RFC2434], i.e. review by one of a group of experts designated by an IETF Operations and Management Area Director. The group of experts must double check the Information Elements definitions with already defined Information Elements for completeness, accuracy, redundancy, and correct naming following the naming conventions in section 2.3. Those experts will initially be drawn from the Working Group Chairs and document editors of the IPFIX and PSAMP Working Groups.

Appendix B defines an XML schema which may be used to create consistent machine readable extensions to the IPFIX information model. This schema introduces a new namespace, which will be assigned by IANA according to RFC 3688. Currently the name space for this schema is identified as http://www.ietf.org/ipfix.

8. Security Considerations

The IPFIX information model itself does not directly introduce security issues. Rather it defines a set of attributes which may for privacy or business issues be considered sensitive information.

The underlying protocol used to exchange the information described here must therefore apply appropriate procedures to guarantee the integrity and confidentiality of the exported information. Such protocols are defined in separate documents, specifically the IPFIX protocol document [I-D.ietf-ipfix-protocol].

9. Acknowledgements

The editors thank Paul Callato for creating the initial version of this document, Thomas Dietz for developing the XSLT scripts that generate large portions of the text part of this document from the XML appendices, and Paul Aitken for a very detailed review that helped improving the document significantly.

10. References

10.1. Normative References

[I-D.ietf-ipfix-protocol]
Claise, B., "IPFIX Protocol Specification", draft-ietf-ipfix-protocol-12 (work in progress), April 2005.

10.2. Informative References

[I-D.ietf-ipfix-architecture]
Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", draft-ietf-ipfix-architecture-07 (work in progress), March 2005.

[I-D.ietf-ipfix-as]
Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IPFIX Applicability", draft-ietf-ipfix-as-04 (work in progress), February 2005.

[IEEE.754.1985]
Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.

[IEEE.802-11.1999]
"Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Standard 802.11, 1999, <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>.

[IEEE.802-3.2002]
"Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications"", IEEE Standard 802.3, September 2002.

[IEEE.P802-1Q.2003]
Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", IEEE Standard 802.1Q, March 2003.

[ISO.10646-1.1993]
International Organization for Standardization, "Information Technology - Universal Multiple-octet coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", ISO Standard 10646-1, May 1993.

[ISO.646.1991]
International Organization for Standardization, "Information technology - ISO 7-bit coded character set for information interchange", ISO Standard 646, 1991.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, March 1996.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

[RFC2547] Rosen, E. and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, March 1999.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[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.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3260] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260, April 2002.

[RFC3667] Bradner, S., "IETF Rights in Contributions", RFC 3667, February 2004.

[RFC3668] Bradner, S., "Intellectual Property Rights in IETF Technology", RFC 3668, February 2004.

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

[RFC3954] Claise, B., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.

Appendix A. Formal Specification of IPFIX Information Element

This appendix contains a formal description of the IPFIX information model XML document. Note that this appendix is of informational nature, while the text in section 4 generated from this appendix is normative.

Using a formal and machine readable syntax for the Information model enables the creation of IPFIX aware tools which can automatically adapt to extensions to the information model, by simply reading updated information model specifications.

The wide availability of XML aware tools and libraries for client devices is a primary consideration for this choice. In particular libraries for parsing XML documents are readily available. Also mechanisms such as the Extensible Stylesheet Language (XSL) allow for transforming a source XML document into other documents. This draft was authored in XML and transformed according to RFC2629.

It should be noted that the use of XML in Exporters, Collectors or other tools is not mandatory for the deployment of IPFIX. In particular Exporting Processes do not produce or consume XML as part of their operation. It is expected that IPFIX Collectors MAY take advantage of the machine readability of the Information Model vs. hard coding their behavior or inventing proprietary means for

accommodating extensions.

```
<fieldDefinitions>

  <field name="ipVersion" dataType="octet"
         group="IpHeader"
         dataTypeSemantics="identifier"
         fieldId="60" applicability="all" status="current">
    <description>
      The IP version field in the IP packet header.
    </description>
    <reference>
      <paragraph>
      See RFC 791 for a definition of the version field in the
      IPv4 packet header.
      See RFC 2460 for a definition of the version field in the
      IPv6 packet header.
      Additional information on defined version numbers
      can be found at
      http://www.iana.org/assignments/version-numbers.
      </paragraph>
    </reference>
  </field>

  <field name="sourceIPv4Address" dataType="ipv4Address"
         group="IpHeader"
         dataTypeSemantics="identifier"
         fieldId="8" applicability="all" status="current">
    <description>
      <paragraph>
      The IPv4 source address in the IP packet header.
      </paragraph>
    </description>
    <reference>
      See RFC 791 for the definition of the IPv4 source address
      field.
    </reference>
  </field>

  <field name="sourceIPv6Address" dataType="ipv6Address"
         group="IpHeader"
         dataTypeSemantics="identifier"
         fieldId="27" applicability="all" status="current">
    <description>
      <paragraph>
```

```
        The IPv6 source address in the IP packet header.
        </paragraph>
      </description>
    </field>

    <field name="sourceIPv4Mask" dataType="octet"
           group="IpHeader"
           fieldId="9" applicability="option" status="current">
      <description>
        <paragraph>
        The number of contiguous bits that are relevant in the
        sourceIPv4Prefix Information Element.
        </paragraph>
      </description>
      <units>bits</units>
      <range>0-32</range>
    </field>

    <field name="sourceIPv6Mask" dataType="octet"
           group="IpHeader"
           fieldId="29" applicability="option" status="current">
      <description>
        <paragraph>
        The number of contiguous bits that are relevant in the
        sourceIPv6Prefix Information Element.
        </paragraph>
      </description>
      <units>bits</units>
      <range>0-128</range>
    </field>

    <field name="sourceIPv4Prefix" dataType="ipv4Address"
           group="IpHeader"
           fieldId="44" applicability="data" status="current">
      <description>
        <paragraph>
        IPv4 source address prefix.
        </paragraph>
      </description>
    </field>

    <field name="sourceIPv6Prefix" dataType="ipv6Address"
           group="IpHeader"
           fieldId="170" applicability="data" status="current">
      <description>
        <paragraph>
        IPv6 source address prefix.
        </paragraph>
```

</description>
</field>

<field name="destinationIPv4Address" dataType="ipv4Address"
group="IpHeader"
dataTypeSemantics="identifier"
fieldId="12" applicability="all" status="current">
<description>
<paragraph>
The IPv4 destination address in the IP packet header.
</paragraph>
</description>
<reference>
See RFC 791 for the definition of the IPv4 destination address
field.
</reference>
</field>

<field name="destinationIPv6Address" dataType="ipv6Address"
group="IpHeader"
dataTypeSemantics="identifier"
fieldId="28" applicability="all" status="current">
<description>
<paragraph>
The IPv6 destination address in the IP packet header.
</paragraph>
</description>
</field>

<field name="destinationIPv4Mask" dataType="octet"
group="IpHeader"
fieldId="13" applicability="option" status="current">
<description>
<paragraph>
The number of contiguous bits that are relevant in the
destinationIPv4Prefix Information Element.
</paragraph>
</description>
<units>bits</units>
<range>0-32</range>
</field>

<field name="destinationIPv6Mask" dataType="octet"
group="IpHeader"
fieldId="30" applicability="option" status="current">
<description>
<paragraph>
The number of contiguous bits that are relevant in the

```
        destinationIPv6Prefix Information Element.
        </paragraph>
      </description>
      <units>bits</units>
      <range>0-128</range>
    </field>

    <field name="destinationIPv4Prefix" dataType="ipv4Address"
           group="IpHeader"
           fieldId="45" applicability="data" status="current">
      <description>
        <paragraph> IPv4 destination address prefix. </paragraph>
      </description>
    </field>

    <field name="destinationIPv6Prefix" dataType="ipv6Address"
           group="IpHeader"
           fieldId="169" applicability="data" status="current">
      <description>
        <paragraph> IPv6 destination address prefix. </paragraph>
      </description>
    </field>

    <field name="ipTimeToLive" dataType="octet"
           group="IpHeader"
           fieldId="192" applicability="all" status="current">
      <description>
        <paragraph>
        For IPv4, the value of the Information Element matches
        the value of the Time to Live field in the IPv4 packet
        header.  For IPv6, the value of the Information Element
        matches the value of the Hop Limit field in the IPv6
        packet header.
        </paragraph>
      </description>
      <reference>
        See RFC 791 for the definition of the IPv4 Time to Live
        field.
        See RFC 2460 for the definition of the IPv6 Hop Limit
        field.
      </reference>
      <units>hops</units>
    </field>

    <field name="protocolIdentifier" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="4" applicability="all" status="current">
```

```
        <description>
          <paragraph>
          The value of the protocol number in the IP packet header.
          The protocol number identifies the IP packet payload type.
          Protocol numbers are defined in the IANA Protocol Numbers
          registry.</paragraph>

          <paragraph>
          In Internet Protocol version 4 (IPv4) this is carried in the
          "Protocol" field. In Internet Protocol version 6 (IPv6) this
          is carried in the "Next Header" field in the last extension
          header of the packet.</paragraph>
        </description>
        <reference>
          <paragraph>
          See RFC 791 for the specification of the IPv4 protocol field.
          See RFC 2460 for the specification of the IPv6 protocol field.
          See the list of protocol numbers assigned by IANA at
          http://www.iana.org/assignments/protocol-numbers.
          </paragraph>
        </reference>
      </field>

      <field name="nextHeaderIPv6" dataType="octet"
             group="IpHeader"
             fieldId="193" applicability="all" status="current">
        <description>
          <paragraph>
          The value of the Next Header field of the IPv6 header.
          The value identifies the type of the following IPv6
          extension header or of the following IP payload.
          Valid values are defined in the IANA
          Protocol Numbers registry.
          </paragraph>
        </description>
        <reference>
          See RFC 2460 for the definition of the IPv6 Next Header field.
          See the list of protocol numbers assigned by IANA at
          http://www.iana.org/assignments/protocol-numbers.
        </reference>
      </field>

      <field name="ipClassOfService" dataType="octet"
             group="IpHeader"
             dataTypeSemantics="identifier"
             fieldId="194" applicability="all" status="current">
        <description>
          <paragraph>
```

```
        For IPv4 packets, this is the value of the TOS field in
        the IPv4 packet header. For IPv6 packets, this is the
        value of the Traffic Class field in the IPv6 packet header.
        </paragraph>
      </description>
      <reference>
        See section 5.3.2 of RFC 1812 and RFC 791 for the
        definition of the IPv4 TOS field.
        See RFC 2460 for the definition of the IPv6 Traffic Class
        field.
      </reference>
    </field>

    <field name="ipDiffServCodePoint" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="195" applicability="all" status="current">
      <description>
        <paragraph>
        The value of a Differentiated Services Code Point (DSCP)
        encoded in the Differentiated Services Field. The
        Differentiated Services Field spans the most significant
        6 bits of the IPv4 TOS field or the IPv6 Traffic class
        field, respectively.
        </paragraph>
        <paragraph>
        This Information Element encodes only the 6 bits of the
        Differentiated Services field. Therefore its value may
        range from 0 to 63.
        </paragraph>
      </description>
      <range>0-63</range>
      <reference>
        See RFC 3260 for the definition of the Differentiated
        Services Field.
        See section 5.3.2 of RFC 1812 and RFC 791 for the
        definition of the IPv4 TOS field.
        See RFC 2460 for the definition of the IPv6 Traffic Class
        field.
      </reference>
    </field>

    <field name="ipPrecedence" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="196" applicability="all" status="current">
      <description>
        <paragraph>
```

```
        The value of the IP Precedence.  The IP Precedence value
        is encoded in the first 3 bits of the IPv4 TOS field
        or the IPv6 Traffic class field, respectively.
        </paragraph>
        <paragraph>
        This Information Element encodes only the 3 bits of the
        Differentiated Services field. Therefore its value may
        range from 0 to 7.
        </paragraph>
      </description>
      <range>0-7</range>
      <reference>
        See section 5.3.3 of RFC 1812 and RFC 791 for the
        definition of the IP Precedence.
        See section 5.3.2 of RFC 1812 and RFC 791 for the
        definition of the IPv4 TOS field.
        See RFC 2460 for the definition of the IPv6 Traffic Class
        field.
      </reference>
    </field>

    <field name="classOfServiceIPv4" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="5" applicability="all" status="current">
      <description>
        <paragraph>
        The value of the TOS field in the IPv4 packet header.
        </paragraph>
      </description>
      <reference>
        See RFC 791 for the definition of the IPv4 TOS field.
      </reference>
    </field>

    <field name="postClassOfServiceIPv4" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="55" applicability="all" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'classOfServiceIPv4', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
```

```
      <reference>
        See RFC 791 for the definition of the IPv4 TOS field.
        See RFC 3234 for the definition of middleboxes.
      </reference>
    </field>

    <field name="classOfServiceIPv6" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="137" applicability="data" status="current">
      <description>
        <paragraph>
        The value of the Traffic Class field in the
        IPv6 packet header.
        </paragraph>
      </description>
      <reference>
        See RFC 2460 for the definition of the IPv6 Traffic Class
        field.
      </reference>
    </field>

    <field name="postClassOfServiceIPv6" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="138" applicability="data" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'classOfServiceIPv6', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
      <reference>
        See RFC 2460 for the definition of the IPv6 traffic class
        field.
      </reference>
    </field>

    <field name="flowLabelIPv6" dataType="unsigned32"
           group="IpHeader"
           dataTypeSemantics="identifier"
           fieldId="31" applicability="all" status="current">
      <description>
        <paragraph>
        The value of the IPv6 Flow Label field in the IP packet header.
```

```
        </paragraph>
      </description>
      <reference>
        See RFC 2460 for a definition of the flow label field in the
        IPv6 packet header.
      </reference>
    </field>

    <field name="isMulticast" dataType="octet"
           group="IpHeader"
           dataTypeSemantics="flags"
           fieldId="206" applicability="data" status="current">
      <description>
        <paragraph>
          If the IP destination address is a reserved multicast
          address then the value of this Information Element is
          not equal to zero.  Otherwise, the value of all bits
          of the octet is zero.
        </paragraph>
        <paragraph>
          The first bit of this octet is set to 1 if the Version
          field of the IP header has the value 4 and if the
          Destination Address field contains a reserved multicast
          address in the range from 224.0.0.0 to 239.255.255.255.
          Otherwise, this bit is set to 0.
        </paragraph>
        <paragraph>
          The second and third bit of this octet are reserved for
          future use.
        </paragraph>
        <paragraph>
          The remaining bits of the octet are only set to values
          other than zero if th IP Destination Address is a
          reserved IPv6 multicast address. Then the fourth bit
          of the octet is set to the value of the T flag in the
          IPv6 multicast address and the remaining four bits are
          set to the value of the scope field in the IPv6
          multicast address.
        </paragraph>
        <artwork>
            0      1      2      3      4      5      6      7
         +------+------+------+------+------+------+------+------+
         | MCv4 | RES. | RES. |  T   |   IPv6 multicast scope    |
         +------+------+------+------+------+------+------+------+

         Bit 0:    set to 1 if IPv4 multicast
         Bit 1-2:  reserved for future use
         Bit 4:    set to value of T-flag, if IPv6 multicast
```

```
          Bit 4-7:  set to value of multicast scope if IPv6 multicast
         </artwork>
       </description>
       <reference>
         See RFC 1112 for the specification of reserved IPv4 multicast
         addresses.
         See RFC 3513 for the specification of reserved IPv6 multicast
         addresses and the definition of the T-flag and the IPv6
         multicast scope.
       </reference>
     </field>

     <field name="identificationIPv4" dataType="unsigned16"
            group="IpHeader"
            dataTypeSemantics="identifier"
            fieldId="54" applicability="data" status="current">
       <description>
         <paragraph>
         The value of the IPv4 packet identification field
         in the IP packet header.
         </paragraph>
       </description>
       <reference>
         See RFC 791 for the definition of the IPv4 identification
         field.
       </reference>
     </field>

       <field name="fragmentOffsetIPv4" dataType="unsigned16"
            group="IpHeader"
            dataTypeSemantics="identifier"
            fieldId="88" applicability="all" status="current">
       <description>
         <paragraph>
         The value of the IPv4 fragment offset field
         in the IP packet header.
         </paragraph>
       </description>
       <reference>
         <paragraph>
         See RFC 791 for the specification of the IPv4 fragment offset.
         </paragraph>
       </reference>
     </field>

       <field name="fragmentFlagsIPv4" dataType="octet"
            group="IpHeader"
            dataTypeSemantics="flags"
```

```
          fieldId="197" applicability="all" status="current">
    <description>
      <paragraph>
      The value of the fragmentation bits
      in the IPv4 packet header.
      </paragraph>
      <artwork>
      Bit 0:    reserved, must be zero.
      Bit 1:    (DF) 0 = May Fragment,  1 = Don't Fragment.
      Bit 2:    (MF) 0 = Last Fragment, 1 = More Fragments.
      Bits 3-7: (DC) Don't Care, value is irrelevant.

           0   1   2   3   4   5   6   7
         +---+---+---+---+---+---+---+---+
         |   | D | M | D | D | D | D | D |
         | 0 | F | F | C | C | C | C | C |
         +---+---+---+---+---+---+---+---+
      </artwork>
    </description>
    <reference>
      <paragraph>
      See RFC 791 for the specification of the IPv4 fragment flags.
      </paragraph>
    </reference>
  </field>

    <field name="ipHeaderLength" dataType="octet"
         group="IpHeader"
         fieldId="189" applicability="all" status="current">
    <description>
      <paragraph>
      The length of the IP header.  For IPv6, the value of this
      Information Element is 40.
      </paragraph>
    </description>
    <reference>
      <paragraph>
      See RFC 791 for the specification of the IPv4 header.
      See RFC 2460 for the specification of the IPv6 header.
      </paragraph>
    </reference>
    <units>octets</units>
  </field>

    <field name="headerLengthIPv4" dataType="octet"
         group="IpHeader"
         fieldId="213" applicability="all" status="current">
    <description>
```

```
        <paragraph>
        The length of the IPv4 header.
        </paragraph>
      </description>
      <reference>
        <paragraph>
        See RFC 791 for the specification of the IPv4 header.
        </paragraph>
      </reference>
      <units>octets</units>
    </field>

      <field name="internetHeaderLengthIPv4" dataType="octet"
          group="IpHeader"
          fieldId="207" applicability="all" status="current">
      <description>
        <paragraph>
        The value of the Internet Header Length (IHL) field in
        the IPv4 header. It specifies the length of the header
        in units of 4 octets.
        </paragraph>
      </description>
      <reference>
        <paragraph>
        See RFC 791 for the specification of the IPv4 header.
        </paragraph>
      </reference>
      <units>4 octets</units>
    </field>

      <field name="totalLengthIPv4" dataType="unsigned16"
          group="IpHeader"
          fieldId="190" applicability="all" status="current">
      <description>
        <paragraph>
        The total length of the IPv4 packet.
        </paragraph>
      </description>
      <reference>
        <paragraph>
        See RFC 791 for the specification of the IPv4 total length.
        </paragraph>
      </reference>
      <units>octets</units>
    </field>

      <field name="payloadLengthIPv6" dataType="unsigned32"
          group="IpHeader"
```

```
          fieldId="191" applicability="all" status="current">
      <description>
        <paragraph>
        The length of the IPv6 payload, i.e., the rest of the
        packet following the IPv6 header, in octets.  Note that any
        extension headers present are considered part
        of the payload, i.e., included in the length count.
        </paragraph>
        <paragraph>
        This Information Element reports the value of the Payload
        Length field in the IPv6 header except in the case that
        the value of this field is zero and that there is a valid
        jumbo payload option. Then the value of the Jumbo Payload
        Length field in the jumbo payload option is reported.
        </paragraph>
      </description>
      <reference>
        <paragraph>
        See RFC 2460 for the specification of the IPv6 payload length.
        See RFC 2675 for the specification of the IPv6 jumbo payload
        length.
        </paragraph>
      </reference>
    </field>

      <field name="ipPayloadLength" dataType="unsigned64"
          group="IpHeader"
          fieldId="204" applicability="all" status="current">
      <description>
        <paragraph>
        The effective length of the IP payload.
        </paragraph>
        <paragraph>
        For IPv4 packets the value of this Information Element is
        the difference between the total length of the IPv4 packet
        (as reported by Information Element totalLengthIPv4) and the
        length of the IPv4 header (as reported by Information Element
        headerLengthIPv4).
        </paragraph>
        <paragraph>
        For IPv6, the value of the Payload Length field
        in the IPv6 header is reported except in the case that
        the value of this field is zero and that there is a valid
        jumbo payload option. In this case the value of the
        Jumbo Payload Length field in the jumbo payload option
        is reported.
        </paragraph>
      </description>
```

```
       <reference>
         <paragraph>
         See RFC 791 for the specification of IPv4 packets.
         See RFC 2460 for the specification of the IPv6 payload length.
         See RFC 2675 for the specification of the IPv6 jumbo payload
         length.
         </paragraph>
       </reference>
     </field>

     <field name="sourceTransportPort" dataType="unsigned16"
            group="transportHeader"
            dataTypeSemantics="identifier"
            fieldId="7" applicability="all" status="current">
       <description>
         <paragraph>
         The source port identifier in the transport header.
         For the transport
         protocols UDP, TCP and SCTP this is the source port number
         given in the respective header.  This field MAY also be used
         for future transport protocols that have 16 bit source port
         identifiers.
         </paragraph>
       </description>
       <reference>
         <paragraph>
         See RFC 768 for the definition of the UDP source port field.
         See RFC 793 for the definition of the TCP source port field.
         See RFC 2960 for the definition of SCTP.</paragraph>
         <paragraph>
         Additional information on defined UDP and TCP port numbers can
         be found at http://www.iana.org/assignments/port-numbers.
         </paragraph>
       </reference>
     </field>

     <field name="destinationTransportPort" dataType="unsigned16"
            group="transportHeader"
            dataTypeSemantics="identifier"
            fieldId="11" applicability="all" status="current">
       <description>
         <paragraph>
         The destination port identifier in the transport header.
         For the transport protocols UDP, TCP and SCTP this is the
         destination port number given in the respective header.
         This field MAY also be used for future transport protocols
         that have 16 bit destination port identifiers.
         </paragraph>
```

```
    </description>
    <reference>
      <paragraph>
      See RFC 768 for the definition of the UDP source port field.
      See RFC 793 for the definition of the TCP source port field.
      See RFC 2960 for the definition of SCTP.  </paragraph>

      <paragraph>
      Additional information on defined UDP and TCP port numbers can
      be found at http://www.iana.org/assignments/port-numbers.
      </paragraph>
    </reference>
  </field>

  <field name="udpSourcePort" dataType="unsigned16"
         group="transportHeader"
         dataTypeSemantics="identifier"
         fieldId="180" applicability="all" status="current">
    <description>
      The source port identifier in the UDP header.
    </description>
    <reference>
      See RFC 768 for the definition of the UDP source port field.
      Additional information on defined UDP port numbers can
      be found at http://www.iana.org/assignments/port-numbers.
    </reference>
  </field>

  <field name="udpDestinationPort" dataType="unsigned16"
         group="transportHeader"
         dataTypeSemantics="identifier"
         fieldId="181" applicability="all" status="current">
    <description>
      The destination port identifier in the UDP header.
    </description>
    <reference>
      See RFC 768 for the definition of the UDP source port field.
      Additional information on defined UDP port numbers can
      be found at http://www.iana.org/assignments/port-numbers.
    </reference>
  </field>

  <field name="udpMessageLength" dataType="unsigned16"
         group="transportHeader"
         fieldId="205" applicability="all" status="current">
    <description>
      The value of the Length field in the UDP header.
    </description>
```

```
        <reference>
          See RFC 768 for the specification of the UDP header.
        </reference>
        <units>octets</units>
      </field>

      <field name="tcpSourcePort" dataType="unsigned16"
             group="transportHeader"
             dataTypeSemantics="identifier"
             fieldId="182" applicability="all" status="current">
        <description>
          The source port identifier in the TCP header.
        </description>
        <reference>
          See RFC 793 for the definition of the TCP source port field.
          Additional information on defined TCP port numbers can
          be found at http://www.iana.org/assignments/port-numbers.
        </reference>
      </field>

      <field name="tcpDestinationPort" dataType="unsigned16"
             group="transportHeader"
             dataTypeSemantics="identifier"
             fieldId="183" applicability="all" status="current">
        <description>
          The destination port identifier in the TCP header.
        </description>
        <reference>
          See RFC 793 for the definition of the TCP source port field.
          Additional information on defined TCP port numbers can
          be found at http://www.iana.org/assignments/port-numbers.
        </reference>
      </field>

      <field name="tcpSequenceNumber" dataType="unsigned32"
             group="transportHeader"
             fieldId="184" applicability="all" status="current">
        <description>
          The sequence number in the TCP header.
        </description>
        <reference>
          See RFC 793 for the definition of the TCP sequence number.
        </reference>
      </field>

      <field name="tcpAcknowledgementNumber" dataType="unsigned32"
             group="transportHeader"
             fieldId="185" applicability="all" status="current">
```

```
    <description>
      The acknowledgement number in the TCP header.
    </description>
    <reference>
      See RFC 793 for the definition of the TCP acknowledgement
      number.
    </reference>
  </field>

  <field name="tcpWindowSize" dataType="unsigned16"
         group="transportHeader"
         fieldId="186" applicability="all" status="current">
    <description>
      The window field in the TCP header.
    </description>
    <reference>
      See RFC 793 for the definition of the TCP window field.
    </reference>
  </field>

  <field name="tcpUrgentPointer" dataType="unsigned16"
         group="transportHeader"
         fieldId="187" applicability="all" status="current">
    <description>
      The urgent pointer in the TCP header.
    </description>
    <reference>
      See RFC 793 for the definition of the TCP urgent pointer.
    </reference>
  </field>

  <field name="tcpHeaderLength" dataType="unsigned16"
         group="transportHeader"
         fieldId="188" applicability="all" status="current">
    <description>
      The length of the TCP header.
    </description>
    <reference>
      See RFC 793 for the definition of the TCP header.
    </reference>
    <units>octets</units>
  </field>

  <field name="icmpTypeCodeIPv4" dataType="unsigned16"
         group="transportHeader"
         dataTypeSemantics="identifier"
         fieldId="32" applicability="all" status="current">
    <description>
```

```
        <paragraph>
        Type and Code of the IPv4 ICMP message. The combination of
        both values is reported as (ICMP type * 256) + ICMP code.
        </paragraph>
      </description>
      <reference>
        See RFC 792 for a definition of the IPv4 ICMP type and code
        fields.
      </reference>
    </field>

    <field name="icmpTypeIPv4" dataType="octet"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="176" applicability="all" status="current">
      <description>
        <paragraph>
        Type of the IPv4 ICMP message.
        </paragraph>
      </description>
      <reference>
        See RFC 792 for a definition of the IPv4 ICMP type field.
      </reference>
    </field>

    <field name="icmpCodeIPv4" dataType="octet"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="177" applicability="all" status="current">
      <description>
        <paragraph>
        Code of the IPv4 ICMP message.
        </paragraph>
      </description>
      <reference>
        See RFC 792 for a definition of the IPv4 ICMP code field.
      </reference>
    </field>

    <field name="icmpTypeCodeIPv6" dataType="unsigned16"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="139" applicability="all" status="current">
      <description>
        <paragraph>
        Type and Code of the IPv6 ICMP message. The combination of
        both values is reported as (ICMP type * 256) + ICMP code.
        </paragraph>
```

```
      </description>
      <reference>
        See RFC 2463 for a definition of the IPv6 ICMP type and code
        fields.
      </reference>
    </field>

    <field name="icmpTypeIPv6" dataType="octet"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="178" applicability="all" status="current">
      <description>
        <paragraph>
        Type of the IPv6 ICMP message.
        </paragraph>
      </description>
      <reference>
        See RFC 2463 for a definition of the IPv6 ICMP type field.
      </reference>
    </field>

    <field name="icmpCodeIPv6" dataType="octet"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="179" applicability="all" status="current">
      <description>
        <paragraph>
        Code of the IPv6 ICMP message.
        </paragraph>
      </description>
      <reference>
        See RFC 2463 for a definition of the IPv6 ICMP code field.
      </reference>
    </field>

    <field name="igmpType" dataType="octet"
           group="transportHeader"
           dataTypeSemantics="identifier"
           fieldId="33" applicability="all" status="current">
      <description>
        The type field of the IGMP message.
      </description>
      <reference>
        See RFC 2236 for a definition of the IGMP type field.
      </reference>
    </field>

    <field name="sourceMacAddress" dataType="macAddress"
```

```
              group="subIpHeader"
              dataTypeSemantics="identifier"
              fieldId="56" applicability="data" status="current">
         <description>
           <paragraph>
             The IEEE 802 source MAC address field.
           </paragraph>
         </description>
         <reference>
           See IEEE.802-3.2002.
         </reference>
       </field>

       <field name="postSourceMacAddress" dataType="macAddress"
              group="subIpHeader"
              dataTypeSemantics="identifier"
              fieldId="81" applicability="data" status="current">
         <description>
           <paragraph>
           The definition of this Information Element is identical
           to the definition of Information Element
           'sourceMacAddress', except that it reports a
           potentially modified value caused by a middlebox
           function after the packet passed the observation point.
           </paragraph>
         </description>
         <reference>
           See IEEE.802-3.2002.
         </reference>
       </field>

       <field name="vlanId" dataType="unsigned16"
              group="subIpHeader"
              dataTypeSemantics="identifier"
              fieldId="58" applicability="data" status="current">
         <description>
           <paragraph>
             The IEEE 802.1Q VLAN identifier (VID) extracted from the Tag
             Control Information field that was attached to the IP packet.
           </paragraph>
         </description>
         <reference>
           See IEEE.802-1Q.2003.
         </reference>
       </field>

       <field name="postVlanId" dataType="unsigned16"
              group="subIpHeader"
```

```
            dataTypeSemantics="identifier"
            fieldId="59" applicability="data" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'vlanId', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
      <reference>
        See IEEE.802-1Q.2003.
      </reference>
    </field>

    <field name="destinationMacAddress" dataType="macAddress"
            group="subIpHeader"
            dataTypeSemantics="identifier"
            fieldId="80" applicability="data" status="current">
      <description>
        <paragraph>
          The IEEE 802 destination MAC address field.
        </paragraph>
      </description>
      <reference>
        See IEEE.802-3.2002.
      </reference>
    </field>

    <field name="postDestinationMacAddr" dataType="macAddress"
            group="subIpHeader"
            dataTypeSemantics="identifier"
            fieldId="57" applicability="data" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'destinationMacAddress', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
      <reference>
        See IEEE.802-3.2002.
      </reference>
    </field>
```

```
<field name="wlanChannelId" dataType="octet"
       group="subIpHeader"
       dataTypeSemantics="identifier"
       fieldId="146" applicability="data" status="current">
  <description>
    <paragraph>
      The identifier of the 802.11 (Wi-Fi) channel used.
    </paragraph>
  </description>
  <reference>
    See IEEE.802-11.1999.
  </reference>
</field>

<field name="wlanSsid" dataType="string"
       group="subIpHeader"
       fieldId="147" applicability="data" status="current">
  <description>
    <paragraph>
      The Service Set IDentifier (SSID) identifying an 802.11
      (Wi-Fi) network used.  According to IEEE.802-11.1999 the
      SSID is encoded into a string of up to 32 characters.
    </paragraph>
  </description>
  <reference>
    See IEEE.802-11.1999.
  </reference>
</field>

<field name="mplsTopLabelTtl" dataType="unsigned32"
       group="subIpHeader"
       fieldId="200" applicability="all" status="current">
  <description>
    The TTL field from the top MPLS label stack entry,
    i.e. the last label that was pushed.
  </description>
  <reference>
    See RFC 3032 for the specification of the TTL field.
  </reference>
</field>

<field name="mplsTopLabelExp" dataType="octet"
       group="subIpHeader"
       dataTypeSemantics="flags"
       fieldId="203" applicability="all" status="current">
  <description>
    <paragraph>
    The Exp field from the top MPLS label stack entry,
```

```
         i.e. the last label that was pushed.
         </paragraph>
         <artwork>
         Bit 0-4:  Don't Care, value is irrelevant.
         Bit 5-7:  MPLS Exp field

             0   1   2   3   4   5   6   7
           +---+---+---+---+---+---+---+---+
           |     don't care    |    Exp    |
           +---+---+---+---+---+---+---+---+
         </artwork>
       </description>
       <reference>
         See RFC 3032 for the specification of the Exp field.
         See RFC 3270 for usage of the Exp field.
       </reference>
     </field>

     <field name="mplsLabelStackDepth" dataType="unsigned32"
            group="subIpHeader"
            fieldId="202" applicability="all" status="current">
       <description>
         The number of labels in the MPLS label stack.
       </description>
       <reference>
         See RFC 3032 for the specification of the MPLS label
         stack.
       </reference>
       <units>label stack entries</units>
     </field>

     <field name="mplsLabelStackLength" dataType="unsigned32"
            group="subIpHeader"
            fieldId="201" applicability="all" status="current">
       <description>
         The length of the MPLS label stack in units of octets.
       </description>
       <reference>
         See RFC 3032 for the specification of the MPLS label
         stack.
       </reference>
       <units>octets</units>
     </field>

     <field name="mplsPayloadLength" dataType="unsigned32"
            group="subIpHeader"
            fieldId="214" applicability="all" status="current">
       <description>
```

```
          The size of the MPLS packet without the label stack.
        </description>
        <reference>
          See RFC 3031 for the specification of MPLS packets.
          See RFC 3032 for the specification of the MPLS label
          stack.
        </reference>
        <units>octets</units>
      </field>

      <field name="mplsTopLabelStackEntry" dataType="unsigned32"
             group="subIpHeader"
             dataTypeSemantics="identifier"
             fieldId="70" applicability="all" status="current">
        <description>
          <paragraph>
          The label, exp and s fields from the top MPLS label
          stack entry, i.e. the last label that was pushed.
          </paragraph>
          <artwork>
        0                   1                   2
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                Label                  | Exp |S|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

       Label:  Label Value, 20 bits
       Exp:    Experimental Use, 3 bits
       S:      Bottom of Stack, 1 bit
          </artwork>
        </description>
        <reference>
          See RFC 3032.
        </reference>
      </field>

      <field name="mplsLabelStackEntry2" dataType="unsigned32"
             group="subIpHeader"
             dataTypeSemantics="identifier"
             fieldId="71" applicability="all" status="current">
        <description>
          <paragraph>
          The label, exp, and s fields from the label stack entry that
          was pushed immediately before the label stack entry that would
          be reported by mplsTopLabelStackEntry. See the definition of
          mplsTopLabelStackEntry for further details.
          </paragraph>
        </description>
```

```
      <reference>
        See RFC 3032.
      </reference>
    </field>

    <field name="mplsLabelStackEntry3" dataType="unsigned32"
           group="subIpHeader"
           dataTypeSemantics="identifier"
           fieldId="72" applicability="all" status="current">
      <description>
        <paragraph>
        The label, exp, and s fields from the label stack entry that
        was pushed immediately before the label stack entry that would
        be reported by mplsLabelStackEntry2. See the definition of
        mplsTopLabelStackEntry for further details.
        </paragraph>
      </description>
      <reference>
        See RFC 3032.
      </reference>
    </field>

    <field name="mplsLabelStackEntry4" dataType="unsigned32"
           group="subIpHeader"
           dataTypeSemantics="identifier"
           fieldId="73" applicability="all" status="current">
      <description>
        <paragraph>
        The label, exp, and s fields from the label stack entry that
        was pushed immediately before the label stack entry that would
        be reported by mplsLabelStackEntry3. See the definition of
        mplsTopLabelStackEntry for further details.
        </paragraph>
      </description>
      <reference>
        See RFC 3032.
      </reference>
    </field>

    <field name="mplsLabelStackEntry5" dataType="unsigned32"
           group="subIpHeader"
           dataTypeSemantics="identifier"
           fieldId="74" applicability="all" status="current">
      <description>
        <paragraph>
        The label, exp, and s fields from the label stack entry that
        was pushed immediately before the label stack entry that would
        be reported by mplsLabelStackEntry4. See the definition of
```

```
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="mplsLabelStackEntry6" dataType="unsigned32"
         group="subIpHeader"
         dataTypeSemantics="identifier"
         fieldId="75" applicability="all" status="current">
    <description>
      <paragraph>
      The label, exp, and s fields from the label stack entry that
      was pushed immediately before the label stack entry that would
      be reported by mplsLabelStackEntry5. See the definition of
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="mplsLabelStackEntry7" dataType="unsigned32"
         group="subIpHeader"
         dataTypeSemantics="identifier"
         fieldId="76" applicability="all" status="current">
    <description>
      <paragraph>
      The label, exp, and s fields from the label stack entry that
      was pushed immediately before the label stack entry that would
      be reported by mplsLabelStackEntry6. See the definition of
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="mplsLabelStackEntry8" dataType="unsigned32"
         group="subIpHeader"
         dataTypeSemantics="identifier"
         fieldId="77" applicability="all" status="current">
    <description>
      <paragraph>
```

```
      The label, exp, and s fields from the label stack entry that
      was pushed immediately before the label stack entry that would
      be reported by mplsLabelStackEntry7. See the definition of
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="mplsLabelStackEntry9" dataType="unsigned32"
         group="subIpHeader"
         dataTypeSemantics="identifier"
         fieldId="78" applicability="all" status="current">
    <description>
      <paragraph>
      The label, exp, and s fields from the label stack entry that
      was pushed immediately before the label stack entry that would
      be reported by mplsLabelStackEntry8. See the definition of
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="mplsLabelStackEntry10" dataType="unsigned32"
         group="subIpHeader"
         dataTypeSemantics="identifier"
         fieldId="79" applicability="all" status="current">
    <description>
      <paragraph>
      The label, exp, and s fields from the label stack entry that
      was pushed immediately before the label stack entry that would
      be reported by mplsLabelStackEntry9. See the definition of
      mplsTopLabelStackEntry for further details.
      </paragraph>
    </description>
    <reference>
      See RFC 3032.
    </reference>
  </field>

  <field name="ipNextHopIPv4Address" dataType="ipv4Address"
         group="derived"
         dataTypeSemantics="identifier"
```

```
           fieldId="15" applicability="data" status="current">
      <description>
        <paragraph>
        The IPv4 address of the next IPv4 hop.
        </paragraph>
      </description>
    </field>

    <field name="ipNextHopIPv6Address" dataType="ipv6Address"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="62" applicability="data" status="current">
      <description>
        <paragraph>
        The IPv6 address of the next IPv6 hop.
        </paragraph>
      </description>
    </field>

  <!--
    <field name="ipNextHopAsNumber" dataType="unsigned16"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="162" applicability="all" status="current">
      <description>
        <paragraph>
        The autonomous system (AS) number of the next IP hop.
        </paragraph>
      </description>
      <reference>
        See RFC 1771 for a description of BGP-4 and
        see RFC 1930 for a definition of the AS number.
      </reference>
    </field>
  -->

    <field name="bgpSourceAsNumber" dataType="unsigned16"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="16" applicability="all" status="current">
      <description>
        <paragraph>
        The autonomous system (AS) number of the source IP address.
        If AS path information for this Flow is only available as
        unordered AS set (and not as ordered AS sequence),
        then the value of this Information Element is 0.
        </paragraph>
      </description>
```

```
        <reference>
          See RFC 1771 for a description of BGP-4 and
          see RFC 1930 for a definition of the AS number.
        </reference>
      </field>

      <field name="bgpDestinationAsNumber" dataType="unsigned16"
             group="derived"
             dataTypeSemantics="identifier"
             fieldId="17" applicability="all" status="current">
        <description>
          <paragraph>
          The autonomous system (AS) number of the destination IP
          address.  If AS path information for this Flow is only
          available as unordered AS set (and not as ordered AS
          sequence), then the value of this Information Element is 0.
          </paragraph>
        </description>
        <reference>
          See RFC 1771 for a description of BGP-4 and
          see RFC 1930 for a definition of the AS number.
        </reference>
      </field>

      <field name="bgpNextAdjacentAsNumber" dataType="unsigned16"
             group="derived"
             dataTypeSemantics="identifier"
             fieldId="128" applicability="all" status="current">
        <description>
          <paragraph>
          The autonomous system (AS) number of the first AS in the AS
          path to the destination IP address.  The path is deduced
          by looking up the destination IP address of the Flow in the
          BGP routing information base.  If AS path information for
          this Flow is only available as unordered AS set (and not as
          ordered AS sequence),
          then the value of this Information Element is 0.
          </paragraph>
        </description>
        <reference>
          See RFC 1771 for a description of BGP-4 and
          see RFC 1930 for a definition of the AS number.
        </reference>
      </field>

      <field name="bgpPrevAdjacentAsNumber" dataType="unsigned16"
             group="derived"
             dataTypeSemantics="identifier"
```

```
         fieldId="129" applicability="all" status="current">
  <description>
    <paragraph>
    The autonomous system (AS) number of the last AS in the AS
    path from the source IP address.  The path is deduced
    by looking up the source IP address of the Flow in the BGP
    routing information base.  If AS path information for this
    Flow is only available as unordered AS set (and not as
    ordered AS sequence), then the value of this Information
    Element is 0.  In case of BGP asymmetry, the
    bgpPrevAdjacentAsNumber might not be
    able to report the correct value.
    </paragraph>
  </description>
  <reference>
    See RFC 1771 for a description of BGP-4 and
    see RFC 1930 for a definition of the AS number.
  </reference>
</field>

<field name="bgpNextHopIPv4Address" dataType="ipv4Address"
       group="derived"
       dataTypeSemantics="identifier"
       fieldId="18" applicability="all" status="current">
  <description>
    <paragraph>
    The IPv4 address of the next (adjacent) BGP hop.
    </paragraph>
  </description>
  <reference>
    See RFC 1771 for a description of BGP-4 and
  </reference>
</field>

<field name="bgpNextHopIPv6Address" dataType="ipv6Address"
       group="derived"
       dataTypeSemantics="identifier"
       fieldId="63" applicability="all" status="current">
  <description>
    <paragraph>
    The IPv6 address of the next (adjacent) BGP hop.
    </paragraph>
  </description>
  <reference>
    See RFC 1771 for a description of BGP-4.
  </reference>
</field>
```

```
    <field name="mplsTopLabelType" dataType="octet"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="46" applicability="data" status="current">
      <description>
        <paragraph>
        This field identifies the control protocol that allocated the
        top of stack label.  Defined values for this field include:
        <artwork>
       - 0x01 TE-MIDPT: Any TE tunnel mid-point or tail label
       - 0x02 Pseudowire: Any PWE3 or Cisco AToM based label
       - 0x03 VPN: Any label associated with VPN
       - 0x04 BGP: Any label associated with BGP or BGP routing
       - 0x05 LDP: Any label associated with dynamically assigned
                   labels using LDP
        </artwork>
        </paragraph>
      </description>
      <reference>
        See RFC 3031 for the MPLS label structure.
        See RFC 2547 for the association of MPLS labels with VPNs.
        See RFC 1771 for BGP and BGP routing.
        See RFC 3036 for LDP.
        and IP addresses.
      </reference>
    </field>

    <field name="mplsTopLabelIPv4Address" dataType="ipv4Address"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="47" applicability="data" status="current">
      <description>
        <paragraph>
          The IPv4 address of the system that the MPLS top label will
          cause this Flow to be forwarded to.
        </paragraph>
      </description>
      <reference>
        See RFC 3031 for the association between MPLS labels
        and IP addresses.
      </reference>
    </field>

    <field name="mplsTopLabelIPv6Address" dataType="ipv6Address"
           group="derived"
           dataTypeSemantics="identifier"
           fieldId="140" applicability="data" status="current">
      <description>
```

```
        <paragraph>
          The IPv6 address of the system that the MPLS top label will
          cause this Flow to be forwarded to.
        </paragraph>
      </description>
      <reference>
        See RFC 3031 for the association between MPLS labels
        and IP addresses.
      </reference>
    </field>

    <field name="exporterIPv4Address" dataType="ipv4Address"
           dataTypeSemantics="identifier"
           group="config"
           fieldId="130" applicability="all" status="current">
      <description>
        <paragraph>
        The IPv4 address used by the Exporting Process.  This is used
        by the Collector to identify the Exporter in cases where the
        identity of the Exporter may have been obscured by the use of
        a proxy.
        </paragraph>
      </description>
    </field>

    <field name="exporterIPv6Address" dataType="ipv6Address"
           dataTypeSemantics="identifier"
           group="config"
           fieldId="131" applicability="all" status="current">
      <description>
        <paragraph>
        The IPv6 address used by the Exporting Process.  This is used
        by the Collector to identify the Exporter in cases where the
        identity of the Exporter may have been obscured by the use of
        a proxy.
        </paragraph>
      </description>
    </field>

    <field name="minimumPacketLength" dataType="unsigned16"
           group="minMax"
           fieldId="25" applicability="all" status="current">
      <description>
        <paragraph>
        Length of the smallest packet observed for this Flow.
        </paragraph>
      </description>
      <units>octets</units>
```

```
     </field>

     <field name="maximumPacketLength" dataType="unsigned16"
            group="minMax"
            fieldId="26" applicability="all" status="current">
       <description>
         <paragraph>
         Length of the largest packet observed for this Flow.
         </paragraph>
       </description>
       <units>octets</units>
     </field>

     <field name="minimumTtl" dataType="octet"
            group="minMax"
            fieldId="52" applicability="data" status="current">
       <description>
         <paragraph>
           Minimum TTL value observed for any packet in this Flow.
         </paragraph>
       </description>
     </field>

     <field name="maximumTtl" dataType="octet"
            group="minMax"
            fieldId="53" applicability="data" status="current">
       <description>
         <paragraph>
           Maximum TTL value observed for any packet in this Flow.
         </paragraph>
       </description>
     </field>

     <field name="ipv4Options" dataType="unsigned32"
            dataTypeSemantics="flags"
            group="minMax"
            fieldId="208" applicability="all" status="current">
       <description>
         <paragraph>
         IPv4 options in packets of this Flow.
         The information is encoded in a set of bit fields.  For
         each valid IPv4 option type there is a bit in this set.
         The bit is set to 1 if any observed packet of this Flow
         contains the corresponding IPv4 option type.  Otherwise,
         if no observed packet of this Flow contained the
         respective IPv4 option type, the value of the
         corresponding bit is 0.
         </paragraph>
```

```
<paragraph>
The list of valid IPv4 options is maintained by IANA.
Note that for identifying an option not just the 5-bit
Option Number, but all 8 bits of the Option Type need to
match one of the IPv4 options specified at
http://www.iana.org/assignments/ip-parameters.
</paragraph>
<paragraph>
Options are mapped to bits according to their option numbers.
Option number X is mapped to bit X.
The mapping is illustrated by the figure below.
</paragraph>
<artwork>
      0      1      2      3      4      5      6      7
  +------+------+------+------+------+------+------+------+
  | EOOL | NOP  | SEC  | LSR  |  TS  |E-SEC |CIPSO |  RR  | ...
  +------+------+------+------+------+------+------+------+

      8      9     10     11     12     13     14     15
  +------+------+------+------+------+------+------+------+
... | SID  | SSR  | ZSU  | MTUP | MTUR | FINN | VISA |ENCODE| ...
  +------+------+------+------+------+------+------+------+

     16     17     18     19     20     21     22     23
  +------+------+------+------+------+------+------+------+
... |IMITD | EIP  |  TR  |ADDEXT|RTRALT| SDB  |NSAPA | DPS  | ...
  +------+------+------+------+------+------+------+------+

     24     25     26     27     28     29     30     31
  +------+------+------+------+------+------+------+------+
... | UMP  |               to be assigned by IANA            |
  +------+------+------+------+------+------+------+------+


      Type   Option
  Bit Value  Name    Reference
  ---+-----+-------+------------------------------------
   0     0   EOOL    End of Options List, RFC 791
   1     1   NOP     No Operation, RFC 791
   2   130   SEC     Security, RFC 1108
   3   131   LSR     Loose Source Route, RFC 791
   4    68   TS      Time Stamp, RFC 791
   5   133   E-SEC   Extended Security, RFC 1108
   6   134   CIPSO   Commercial Security
   7     7   RR      Record Route, RFC 791
   8   136   SID     Stream ID, RFC 791
   9   137   SSR     Strict Source Route, RFC 791
  10    10   ZSU     Experimental Measurement
```

```
       11    11   MTUP    (obsoleted) MTU Probe, RFC 1191
       12    12   MTUR    (obsoleted) MTU Reply, RFC 1191
       13   205   FINN    Experimental Flow Control
       14   142   VISA    Experimental Access Control
       15    15   ENDOCE
       16   144   IMITD   IMI Traffic Descriptor
       17   145   EIP     Extended Internet Protocol, RFC 1385
       18    82   TR      Traceroute, RFC 3193
       19   147   ADDEXT  Address Extension
       20   148   RTRALT  Router Alert, RFC 2113
       21   149   SDB     Selective Directed Broadcast
       22   150   NSAPA   NSAP Address
       23   151   DPS     Dynamic Packet State
       24   152   UMP     Upstream Multicast Pkt.
      ...   ...   ...     Further options numbers
                          may be assigned by IANA
         </artwork>
       </description>
       <reference>
         See RFC 791 for the definition of IPv4 options.
         See the list of IPv4 option numbers assigned by IANA
         at http://www.iana.org/assignments/ip-parameters.
       </reference>
     </field>

     <field name="ipv6ExtensionHeaders" dataType="unsigned32"
            dataTypeSemantics="flags"
            group="minMax"
            fieldId="64" applicability="all" status="current">
       <description>
         <paragraph>
        IPv6 extension headers observed in packets of this Flow.
        The information is encoded in a set of bit fields.  For
        each IPv6 option header there is a bit in this set.
        The bit is set to 1 if any observed packet of this Flow
        contains the corresponding IPv6 extension header.
        Otherwise, if no observed packet of this Flow contained
        the respective IPv6 extension header, the value of the
        corresponding bit is 0.
         </paragraph>
         <artwork>
             0     1     2     3     4     5     6     7
          +-----+-----+-----+-----+-----+-----+-----+-----+
          | Res | FRA1| ROU | FRA0| UNK | Res | HOP | DST |  ...
          +-----+-----+-----+-----+-----+-----+-----+-----+

             8     9    10    11    12    13    14    15
          +-----+-----+-----+-----+-----+-----+-----+-----+
```

```
      ... | PAY | AUT | ENC |         Reserved            | ...
          +-----+-----+-----+-----+-----+-----+-----+-----+

             16    17    18    19    20    21    22    23
          +-----+-----+-----+-----+-----+-----+-----+-----+
      ... |                  Reserved                     | ...
          +-----+-----+-----+-----+-----+-----+-----+-----+

             24    25    26    27    28    29    30    31
          +-----+-----+-----+-----+-----+-----+-----+-----+
      ... |                  Reserved                     |
          +-----+-----+-----+-----+-----+-----+-----+-----+

        Bit    IPv6 Option   Description

       0, Res                Reserved
       1, FRA1     44        Fragmentation header - not first fragment
       2, ROU      43        Routing header
       3, FRA0     44        Fragment header - first fragment
       4, UNK                Unknown Layer 4 header
                             (compressed, encrypted, not supported)
       5, Res                Reserved
       6, HOP       0        Hop-by-hop option header
       7, DST      60        Destination option header
       8, PAY     108        Payload compression header
       9, AUT      51        Authentication Header
      10, ENC      50        Encrypted security payload
      11 to 31               Reserved
         </artwork>
       </description>
       <reference>
        See RFC 2460 for the general definition of IPv6 extensions
        headers and for the specification of the hop-by-hop options
        header, the routing header, the fragment header, and the
        destination options header.
        See RFC 2402 for the specification of the authentication
        header.
        See RFC 2406 for the specification of the encapsulating
        security payload.
       </reference>
     </field>

     <field name="tcpControlBits" dataType="octet"
            dataTypeSemantics="flags"
            group="minMax"
            fieldId="6" applicability="all" status="current">
       <description>
         <paragraph>
```

```
        TCP control bits observed for packets of this Flow.
        The information is encoded in a set of bit fields.
        For each TCP control bit there is a bit in this
        set.  A bit is set to 1 if any observed packet of this
        Flow has the corresponding TCP control bit set to 1.
        A value of 0 for a bit indicates that the corresponding
        bit was not set in any of the observed packets
        of this Flow.
        </paragraph>
        <artwork>
         0     1     2     3     4     5     6     7
     +-----+-----+-----+-----+-----+-----+-----+-----+
     |  Reserved | URG | ACK | PSH | RST | SYN | FIN |
     +-----+-----+-----+-----+-----+-----+-----+-----+

     Reserved:  Reserved for future use by TCP.  Must be zero.
          URG:  Urgent Pointer field significant
          ACK:  Acknowledgment field significant
          PSH:  Push Function
          RST:  Reset the connection
          SYN:  Synchronize sequence numbers
          FIN:  No more data from sender
        </artwork>
      </description>
      <reference>See RFC 793 for a definition of the TCP control bits
      in the TCP header.</reference>
    </field>

    <field name="tcpOptions" dataType="unsigned64"
           dataTypeSemantics="flags"
           group="minMax"
           fieldId="209" applicability="all" status="current">
      <description>
        <paragraph>
       TCP options in packets of this Flow.
       The information is encoded in a set of bit fields.  For
       each TCP option there is a bit in this set.
       The bit is set to 1 if any observed packet of this Flow
       contains the corresponding TCP option.
       Otherwise, if no observed packet of this Flow contained
       the respective TCP option, the value of the
       corresponding bit is 0.
        </paragraph>
        <paragraph>
        Options are mapped to bits according to their option
        numbers.  Option number X is mapped to bit X.
        TCP option numbers are maintained by IANA.
        </paragraph>
```

```
        </description>
        <reference>
          See RFC 793 for the definition of TCP options.
          See the list of TCP option numbers assigned by IANA
          at http://www.iana.org/assignments/tcp-parameters.
        </reference>
      </field>

      <field name="flowStartSeconds" dataType="dateTimeSeconds"
             group="timestamp"
             fieldId="150" applicability="data" status="current">
        <description>
          The absolute timestamp of the first packet of this Flow.
        </description>
        <units>seconds</units>
      </field>

      <field name="flowEndSeconds" dataType="dateTimeSeconds"
             group="timestamp"
             fieldId="151" applicability="data" status="current">
        <description>
          The absolute timestamp of the last packet of this Flow.
        </description>
        <units>seconds</units>
      </field>

      <field name="flowStartMilliSeconds" dataType="dateTimeMilliSeconds"
             group="timestamp"
             fieldId="152" applicability="data" status="current">
        <description>
          The absolute timestamp of the first packet of this Flow.
        </description>
        <units>milliseconds</units>
      </field>

      <field name="flowEndMilliSeconds" dataType="dateTimeMilliSeconds"
             group="timestamp"
             fieldId="153" applicability="data" status="current">
        <description>
          The absolute timestamp of the last packet of this Flow.
        </description>
        <units>milliseconds</units>
      </field>

      <field name="flowStartMicroSeconds" dataType="dateTimeMicroSeconds"
             group="timestamp"
             fieldId="154" applicability="data" status="current">
        <description>
```

```
        The absolute timestamp of the first packet of this Flow.
      </description>
      <units>microseconds</units>
    </field>

    <field name="flowEndMicroSeconds" dataType="dateTimeMicroSeconds"
           group="timestamp"
           fieldId="155" applicability="data" status="current">
      <description>
        The absolute timestamp of the last packet of this Flow.
      </description>
      <units>microseconds</units>
    </field>

    <field name="flowStartNanoSeconds" dataType="dateTimeNanoSeconds"
           group="timestamp"
           fieldId="156" applicability="data" status="current">
      <description>
        The absolute timestamp of the first packet of this Flow.
      </description>
      <units>nanoseconds</units>
    </field>

    <field name="flowEndNanoSeconds" dataType="dateTimeNanoSeconds"
           group="timestamp"
           fieldId="157" applicability="data" status="current">
      <description>
        The absolute timestamp of the last packet of this Flow.
      </description>
      <units>nanoseconds</units>
    </field>

    <field name="flowStartDeltaMicroSeconds" dataType="unsigned32"
           group="timestamp"
           fieldId="158" applicability="data" status="current">
      <description>
        This is a relative time stamp only valid within the scope
        of a single IPFIX Message.  It contains the negative time
        offset of the first observed packet of this Flow relative
        to the export time specified in the IPFIX Message header.
      </description>
      <reference>
        See [I-D.ietf-ipfix-protocol] for the definition of the IPFIX
        Message header.
      </reference>
      <units>microseconds</units>
    </field>
```

```
<field name="flowEndDeltaMicroSeconds" dataType="unsigned32"
       group="timestamp"
       fieldId="159" applicability="data" status="current">
  <description>
    This is a relative time stamp only valid within the scope
    of a single IPFIX Message.  It contains the negative time
    offset of the last observed packet of this Flow relative
    to the export time specified in the IPFIX Message header.
  </description>
  <reference>
    See [I-D.ietf-ipfix-protocol] for the definition of the IPFIX
    Message header.
  </reference>
  <units>microseconds</units>
</field>

<field name="systemInitTimeMilliSeconds"
       dataType="dateTimeMilliSeconds"
       group="timestamp"
       fieldId="160" applicability="data" status="current">
  <description>
    The absolute timestamp of the last (re-)initialization of the
    IPFIX Device.
  </description>
  <units>milliseconds</units>
</field>

<field name="flowStartSysUpTime" dataType="unsigned32"
       group="timestamp"
       fieldId="22" applicability="data" status="current">
  <description>
    The relative timestamp of the first packet of this Flow.
    It indicates the number of milliseconds since the
    last (re-)initialization of the IPFIX Device (sysUpTime).
  </description>
  <units>milliseconds</units>
</field>

<field name="flowEndSysUpTime" dataType="unsigned32"
       group="timestamp"
       fieldId="21" applicability="data" status="current">
  <description>
    The relative timestamp of the last packet of this Flow.
    It indicates the number of milliseconds since the
    last (re-)initialization of the IPFIX Device (sysUpTime).
  </description>
  <units>milliseconds</units>
</field>
```

```
     <field name="flowActiveTimeOut" dataType="unsigned16"
            group="misc"
            fieldId="36" applicability="all" status="current">
       <description>
         <paragraph>
         The number of seconds after which an active Flow is timed out
         anyway, even if there is still a continuous flow of packets.
         </paragraph>
       </description>
       <units>seconds</units>
     </field>

     <field name="flowInactiveTimeout" dataType="unsigned16"
            group="misc"
            fieldId="37" applicability="all" status="current">
       <description>
         <paragraph>
          A Flow is considered to be timed out if no packets belonging
          to the Flow have been observed for the number of seconds
          specified by this field.
         </paragraph>
       </description>
       <units>seconds</units>
     </field>

     <field name="flowEndReason" dataType="octet"
            group="misc"
            dataTypeSemantics="identifier"
            fieldId="136" applicability="data" status="current">
       <description>
         <paragraph>
         The reason for Flow termination. The range of values includes
         <artwork>
      0x01: idle timeout
            The flow was terminated because it was considered to be
            idle.
      0x02: active timeout
            The flow was terminated for reporting purposes while it was
            still active, for example, after the maximum lifetime of
            unreported flows was reached.
      0x03: end of Flow detected
            The flow was terminated because the Metering Process
            detected signals indicating the end of the flow,
            for example, the TCP FIN flag.
      0x04: forced end
            The flow was terminated because of some external event,
            for example, a shut down of the Metering Process initiated
            by a network management application.
```

```
      0x05: cache full
            the flow was terminated because of lack of resources
            available to the Metering Process and/or the Exporting
            Process
         </artwork>
         </paragraph>
       </description>
     </field>

     <field name="flowDurationMilliSeconds" dataType="unsigned32"
            group="misc"
            fieldId="161" applicability="data" status="current">
       <description>
         The difference between in time between the observation of the
         first packet of this Flow and the observation of the last
         packet of this Flow.
       </description>
       <units>milliseconds</units>
     </field>

     <field name="flowDurationMicroSeconds" dataType="unsigned32"
            group="misc"
            fieldId="162" applicability="data" status="current">
       <description>
         The difference between in time between the observation of the
         first packet of this Flow and the observation of the last
         packet of this Flow.
       </description>
       <units>microseconds</units>
     </field>

     <field name="octetDeltaCount" dataType="unsigned64"
            dataTypeSemantics="deltaCounter"
            group="flowCounter"
            fieldId="1" applicability="data" status="current">
       <description>
         <paragraph>
         The number of octets since the previous report (if any)
         in incoming packets for this Flow at the Observation Point.
         The number of octets include IP header(s) and IP payload.
         </paragraph>
       </description>
       <units>octets</units>
     </field>

     <field name="postOctetDeltaCount" dataType="unsigned64"
            dataTypeSemantics="deltaCounter"
            group="flowCounter"
```

```
          fieldId="23" applicability="data" status="current">
    &lt;description&gt;
      &lt;paragraph&gt;
      The definition of this Information Element is identical
      to the definition of Information Element
      'octetDeltaCount', except that it reports a
      potentially modified value caused by a middlebox
      function after the packet passed the observation point.
      &lt;/paragraph&gt;
    &lt;/description&gt;
    &lt;units&gt;octets&lt;/units&gt;
  &lt;/field&gt;

  &lt;field name="octetDeltaSumOfSquares" dataType="unsigned64"
          group="flowCounter"
          fieldId="198" applicability="data" status="current"&gt;
    &lt;description&gt;
      &lt;paragraph&gt;
      The sum of the squared numbers of octets per incoming
      packet since the previous report (if any) for this
      Flow at the Observation Point.
      The number of octets include IP header(s) and IP payload.
      &lt;/paragraph&gt;
    &lt;/description&gt;
  &lt;/field&gt;

  &lt;field name="octetTotalCount" dataType="unsigned64"
          dataTypeSemantics="totalCounter"
          group="flowCounter"
          fieldId="85" applicability="all" status="current"&gt;
    &lt;description&gt;
      &lt;paragraph&gt;
      The total number of octets in incoming packets
      for this Flow at the Observation Point since the Metering
      Process (re-)initialization for this Observation Point. The
      number of octets include IP header(s) and IP payload.
      &lt;/paragraph&gt;
    &lt;/description&gt;
    &lt;units&gt;octets&lt;/units&gt;
  &lt;/field&gt;

  &lt;field name="postOctetTotalCount" dataType="unsigned64"
          dataTypeSemantics="totalCounter"
          group="flowCounter"
          fieldId="171" applicability="all" status="current"&gt;
    &lt;description&gt;
      &lt;paragraph&gt;
      The definition of this Information Element is identical
```

```
        to the definition of Information Element
        'octetTotalCount', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
      <units>octets</units>
    </field>

    <field name="octetTotalSumOfSquares" dataType="unsigned64"
           group="flowCounter"
           fieldId="199" applicability="all" status="current">
      <description>
        <paragraph>
        The total sum of the squared numbers of octets in incoming
        packets for this Flow at the Observation Point since the
        Metering Process (re-)initialization for this Observation
        Point. The number of octets include IP header(s) and IP
        payload.
        </paragraph>
      </description>
      <units>octets</units>
    </field>

    <field name="packetDeltaCount" dataType="unsigned64"
           dataTypeSemantics="deltaCounter"
           group="flowCounter"
           fieldId="2" applicability="data" status="current">
      <description>
        <paragraph>
        The number of incoming packets since the previous report
        (if any) for this Flow at the Observation Point.
        </paragraph>
      </description>
      <units>packets</units>
    </field>

    <field name="postPacketDeltaCount" dataType="unsigned64"
           dataTypeSemantics="deltaCounter"
           group="flowCounter"
           fieldId="24" applicability="data" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'packetDeltaCount', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
```

```
        </paragraph>
      </description>
      <units>packets</units>
    </field>

    <field name="packetTotalCount" dataType="unsigned64"
           dataTypeSemantics="totalCounter"
           group="flowCounter"
           fieldId="86" applicability="all" status="current">
      <description>
        <paragraph>
        The total number of incoming packets for this Flow
        at the Observation Point since the Metering Process
        (re-)initialization for this Observation Point.
        </paragraph>
      </description>
      <units>packets</units>
    </field>

    <field name="postPacketTotalCount" dataType="unsigned64"
           dataTypeSemantics="totalCounter"
           group="flowCounter"
           fieldId="172" applicability="all" status="current">
      <description>
        <paragraph>
        The definition of this Information Element is identical
        to the definition of Information Element
        'packetTotalCount', except that it reports a
        potentially modified value caused by a middlebox
        function after the packet passed the observation point.
        </paragraph>
      </description>
      <units>packets</units>
    </field>

    <field name="droppedOctetDeltaCount" dataType="unsigned64"
           dataTypeSemantics="deltaCounter"
           group="flowCounter"
           fieldId="132" applicability="data" status="current">
      <description>
        <paragraph>
        The number of octets since the previous report (if any)
        in packets of this Flow dropped by packet treatment.
        The number of octets include IP header(s) and IP payload.
        </paragraph>
      </description>
      <units>octets</units>
    </field>
```

```
      <field name="droppedPacketDeltaCount" dataType="unsigned64"
             dataTypeSemantics="deltaCounter"
             group="flowCounter"
             fieldId="133" applicability="data" status="current">
        <description>
          <paragraph>
          The number of packets since the previous report (if any)
          of this Flow dropped by packet treatment.
          </paragraph>
        </description>
        <units>packets</units>
      </field>

      <field name="droppedOctetTotalCount" dataType="unsigned64"
             dataTypeSemantics="totalCounter"
             group="flowCounter"
             fieldId="134" applicability="data" status="current">
        <description>
          <paragraph>
          The total number of octets in packets of this Flow dropped
          by packet treatment since the Metering Process
          (re-)initialization for this Observation Point.
          The number of octets include IP header(s) and IP payload.
          </paragraph>
        </description>
        <units>octets</units>
      </field>

      <field name="droppedPacketTotalCount" dataType="unsigned64"
             dataTypeSemantics="totalCounter"
             group="flowCounter"
             fieldId="135" applicability="data" status="current">
        <description>
          <paragraph>
          The number of packets of this Flow dropped by packet
          treatment since the Metering Process
          (re-)initialization for this Observation Point.
          </paragraph>
        </description>
        <units>packets</units>
      </field>

      <field name="postMCastPacketDeltaCount" dataType="unsigned64"
             dataTypeSemantics="deltaCounter"
             group="flowCounter"
             fieldId="19" applicability="data" status="current">
        <description>
          <paragraph>
```

```
          The number of outgoing multicast packets since the
          previous report (if any) sent for packets of this Flow
          by a multicast daemon within the Observation Domain.
          This property cannot necessarily be observed at the
          Observation Point, but may be retrieved by other means.
          </paragraph>
        </description>
        <units>packets</units>
      </field>

      <field name="postMCastOctetDeltaCount" dataType="unsigned64"
             dataTypeSemantics="deltaCounter"
             group="flowCounter"
             fieldId="20" applicability="data" status="current">
        <description>
          <paragraph>
          The number of octets since the previous report (if any)
          in outgoing multicast packets sent for packets of this
          Flow by a multicast daemon within the Observation Domain.
          This property cannot necessarily be observed at the
          Observation Point, but may be retrieved by other means.
          The number of octets include IP header(s) and IP payload.
          </paragraph>
        </description>
        <units>octets</units>
      </field>

      <field name="postMCastPacketTotalCount" dataType="unsigned64"
             dataTypeSemantics="totalCounter"
             group="flowCounter"
             fieldId="174" applicability="data" status="current">
        <description>
          <paragraph>
          The total number of outgoing multicast packets sent for
          packets of this Flow by a multicast daemon within the
          Observation Domain since the Metering Process
          (re-)initialization.  This property cannot necessarily
          be observed at the Observation Point, but may be retrieved
          by other means.
          </paragraph>
        </description>
        <units>packets</units>
      </field>

      <field name="postMCastOctetTotalCount" dataType="unsigned64"
             dataTypeSemantics="totalCounter"
             group="flowCounter"
             fieldId="175" applicability="data" status="current">
```

```
      <description>
        <paragraph>
        The total number of octets in outgoing multicast packets
        sent for packets of this Flow by a multicast daemon in the
        Observation Domain since the Metering Process
        (re-)initialization.  This property cannot necessarily be
        observed at the Observation Point, but may be retrieved by
        other means.
        The number of octets include IP header(s) and IP payload.
        </paragraph>
      </description>
      <units>octets</units>
    </field>

    <field name="exportedMessageTotalCount" dataType="unsigned64"
           dataTypeSemantics="totalCounter"
           group="processCounter"
           fieldId="41" applicability="data" status="current">
      <description>
        <paragraph>
        The total number of IPFIX Messages that the Exporting Process
        successfully sent since the Exporting Process
        (re-)initialization to the Collecting Process receiving a
        report that contains this Information Element.
        </paragraph>
      </description>
      <units>messages</units>
    </field>

    <field name="exportedOctetTotalCount" dataType="unsigned64"
           dataTypeSemantics="totalCounter"
           group="processCounter"
           fieldId="40" applicability="data" status="current">
      <description>
        <paragraph>
           The total number of octets that the Exporting Process
           successfully sent since the Exporting Process
           (re-)initialization to the Collecting Process receiving a
           report that contains this Information Element.  The value
           of this Information Element is calculated by summing up
           the IPFIX Message header length values of all IPFIX Messages
           that were successfully sent to the Collecting Process
           receiving a report that contains this Information Element.
        </paragraph>
      </description>
      <units>octets</units>
    </field>
```

```
<field name="exportedFlowTotalCount" dataType="unsigned64"
       group="processCounter"
       dataTypeSemantics="totalCounter"
       fieldId="42" applicability="data" status="current">
  <description>
    <paragraph>
    The total number of Flows Records that the Exporting
    Process successfully sent as Data Records since the Exporting
    Process (re-)initialization to the Collecting Process receiving
    a report that contains this Information Element.
    </paragraph>
  </description>
  <units>Flows</units>
</field>

<field name="observedFlowTotalCount" dataType="unsigned64"
       dataTypeSemantics="totalCounter"
       group="processCounter"
       fieldId="163" applicability="data" status="current">
  <description>
    <paragraph>
    The total number of Flows observed in the Observation Domain
    since the Metering Process (re-)initialization for this
    Observation Point.
    </paragraph>
  </description>
  <units>Flows</units>
</field>

<field name="ignoredPacketTotalCount" dataType="unsigned64"
       dataTypeSemantics="totalCounter"
       group="processCounter"
       fieldId="164" applicability="data" status="current">
  <description>
    <paragraph>
       The total number of observed IP packets that the
       Metering Process did not process since the
       (re-)initialization of the Metering Process.
    </paragraph>
  </description>
  <units>packets</units>
</field>

<field name="ignoredOctetTotalCount" dataType="unsigned64"
       dataTypeSemantics="totalCounter"
       group="processCounter"
       fieldId="165" applicability="data" status="current">
  <description>
```

```
      <paragraph>
         The total number of octets in observed IP packets
         that the Metering Process did not process since the
         (re-)initialization of the Metering Process.
      </paragraph>
    </description>
    <units>octets</units>
  </field>

  <field name="notSentFlowTotalCount" dataType="unsigned64"
         dataTypeSemantics="totalCounter"
         group="processCounter"
         fieldId="166" applicability="data" status="current">
    <description>
      <paragraph>
         The total number of Flow Records that were generated by the
         Metering Process and but dropped by the Metering Process or
         by the Exporting Process
         instead of sending it to the Collecting Process.
         There are several potential reasons for this including
         resource shortage and special Flow export policies.
      </paragraph>
    </description>
    <units>Flows</units>
  </field>

  <field name="notSentPacketTotalCount" dataType="unsigned64"
         dataTypeSemantics="totalCounter"
         group="processCounter"
         fieldId="167" applicability="data" status="current">
    <description>
      <paragraph>
         The total number of packets in Flow Records that were
         generated by the Metering Process and but dropped
         by the Metering Process or by the Exporting Process
         instead of sending it to the Collecting Process.
         There are several potential reasons for this including
         resource shortage and special Flow export policies.
      </paragraph>
    </description>
    <units>packets</units>
  </field>

  <field name="notSentOctetTotalCount" dataType="unsigned64"
         dataTypeSemantics="totalCounter"
         group="processCounter"
         fieldId="168" applicability="data" status="current">
    <description>
```

```
        <paragraph>
           The total number of octets in packets in Flow Records
           that were generated by the Metering Process and but
           dropped by the Metering Process or by the Exporting
           Process instead of sending it to the Collecting Process.
           There are several potential reasons for this including
           resource shortage and special Flow export policies.
        </paragraph>
      </description>
      <units>octets</units>
    </field>

    <field name="flowKeyIndicator" dataType="unsigned64"
           dataTypeSemantics="flags"
           group="processCounter"
           fieldId="173" applicability="all" status="current">
      <description>
        <paragraph>
        This set of bit fields is used for marking the Information
        Elements of a Data Record that serve as Flow Key.  Each bit
        represents an Information Element in the Data Record with
        the n-th bit representing the n-th Information Element.
        A set bit with value 1 indicates that the corresponding
        Information element is a Flow Key of the reported Flow.
        A value of 0 indicates that this is not the case.  If the
        Data Record contains more than 64 Information Elements, the
        corresponding Template SHOULD be designed such that all Flow
        Keys are among the first 64 Information Elements, because the
        flowKeyIndicator only contains 64 bits.  If the Data Record
        contains less than 64 Information Elements, then the bits in
        the flowKeyIndicator for which no corresponding Information
        Element exists SHOULD have the value 0.
        </paragraph>
      </description>
    </field>

    <field name="lineCardId" dataType="unsigned32"
           group="scope"
           dataTypeSemantics="identifier"
           fieldId="141" applicability="option" status="current">
      <description>
        <paragraph>
          A locally unique identifier of a line card at an IPFIX
          Device hosting an Observation Point.  Typically, this
          Information Element is used for limiting the scope
          of other Information Elements.
        </paragraph>
      </description>
```

```
</field>

<field name="portId" dataType="unsigned32"
       group="scope"
       dataTypeSemantics="identifier"
       fieldId="142" applicability="option" status="current">
  <description>
    <paragraph>
      A locally unique identifier of a line port at an IPFIX
      Device hosting an Observation Point.  Typically, this
      Information Element is used for limiting the scope
      of other Information Elements.
    </paragraph>
  </description>
</field>

<field name="ingressInterface" dataType="unsigned32"
       group="scope"
       dataTypeSemantics="identifier"
       fieldId="10" applicability="all" status="current">
  <description>
    <paragraph>
      The index of the IP interface (ifIndex) where packets of
      this Flow are being received.
    </paragraph>
  </description>
  <reference>
    See RFC 2863 for the definition of the ifIndex object.
  </reference>
</field>

<field name="egressInterface" dataType="unsigned32"
       group="scope"
       dataTypeSemantics="identifier"
       fieldId="14" applicability="all" status="current">
  <description>
    <paragraph>
      The index of the IP interface (ifIndex) where packets of
      this Flow are being sent.
    </paragraph>
  </description>
  <reference>
    See RFC 2863 for the definition of the ifIndex object.
  </reference>
</field>

<field name="meteringProcessId" dataType="unsigned32"
       group="scope"
```

```
            dataTypeSemantics="identifier"
            fieldId="143" applicability="option" status="current">
       <description>
         <paragraph>
           A locally unique identifier of a Metering Process
           at an IPFIX Device.  Typically, this
           Information Element is used for limiting the scope
           of other Information Elements.
         </paragraph>
       </description>
     </field>

     <field name="exportingProcessId" dataType="unsigned32"
            group="scope"
            dataTypeSemantics="identifier"
            fieldId="144" applicability="option" status="current">
       <description>
         <paragraph>
           A locally unique identifier of an Exporting Process
           at an IPFIX Device.  Typically, this
           Information Element is used for limiting the scope
           of other Information Elements.
         </paragraph>
       </description>
     </field>

     <field name="flowId" dataType="unsigned32"
            group="scope"
            dataTypeSemantics="identifier"
            fieldId="148" applicability="option" status="current">
       <description>
         <paragraph>
           An identifier of a Flow that is unique within an Observation
           Domain. This Information Element can be used to distinguish
           between different Flows if Flow Keys such as IP addresses and
           port numbers are not reported or reported in separate
           records.
         </paragraph>
       </description>
     </field>

     <field name="templateId" dataType="unsigned16"
            group="scope"
            dataTypeSemantics="identifier"
            fieldId="145" applicability="option" status="current">
       <description>
         <paragraph>
           An identifier of a Template that is locally unique to an
```

```
          Exporting Process.  Typically, this Information Element is
          used for limiting the scope of other Information Elements.
        </paragraph>
      </description>
    </field>

    <field name="sourceId" dataType="unsigned32"
           group="scope"
           dataTypeSemantics="identifier"
           fieldId="149" applicability="option" status="current">
      <description>
        <paragraph>
          An identifier of an Observation Domain that is locally
          unique to an Exporting Process.  Typically, this Information
          Element is used for limiting the scope of other Information
          Elements.
        </paragraph>
      </description>
    </field>

    <field name="paddingOctets" dataType="octetArray"
           group="padding"
           fieldId="210" applicability="option" status="current">
      <description>
        <paragraph>
          The value of this Information Element is always 0.
        </paragraph>
      </description>
    </field>

  </fieldDefinitions>
```

Appendix B. Formal Specification of Abstract Data Types

This appendix containfs a formal description of the abstract data types to be used for IPFIX Information Elements and a formal description of the template used for defining IPFIX Information Elements. Note that this appendix is of informational nature, while the text in sections 2 and 3 generated from this appendix is normative.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
   <schema>
   <!-- <schema elementFormDefault="qualified"
           targetNamespace="http://www.ietf.org/ipfix"
           xmlns="http://www.w3.org/2001/XMLSchema"
           xmlns:ipfix="http://www.ietf.org/ipfix">  -->

     <simpleType name="dataType">
       <restriction base="string">
         <enumeration value="octet">
           <annotation>
             <documentation>The type "octet" represents a
               non-negative integer value in the range of 0 to 255.
             </documentation>
           </annotation>
         </enumeration>

         <enumeration value="unsigned16">
           <annotation>
             <documentation>The type "unsigned16" represents a
               non-negative integer value in the range of 0 to 65535.
             </documentation>
           </annotation>
         </enumeration>

         <enumeration value="unsigned32">
           <annotation>
             <documentation>The type "unsigned32" represents a
                non-negative integer value in the range of 0 to
                4294967295.
             </documentation>
           </annotation>
         </enumeration>

         <enumeration value="unsigned64">
           <annotation>
             <documentation>The type "unsigned64" represents a
               non-negative integer value in the range of 0 to
               18446744073709551615.
             </documentation>
           </annotation>
         </enumeration>

         <enumeration value="float32">
           <annotation>
             <documentation>The type "float32" corresponds to an IEEE
               single-precision 32-bit floating point type as defined
               in [IEEE.754.1985].
             </documentation>
```

```
      </annotation>
    </enumeration>

    <enumeration value="boolean">
      <annotation>
        <documentation>The type "boolean" represents a binary
          value.  The only allowed values are "true" and false.
        </documentation>
      </annotation>
    </enumeration>

    <enumeration value="macAddress">
      <annotation>
        <documentation>The type "macAddress" represents a
          string of 6 octets.
        </documentation>
      </annotation>
    </enumeration>

    <enumeration value="octetArray">
      <annotation>
        <documentation>The type "octetArray" represents a finite
        length string of octets.
        </documentation>
      </annotation>
    </enumeration>

    <enumeration value="string">
      <annotation>
        <documentation>
          The type "string" represents a finite length string
          of valid characters from the Unicode character encoding
          set [ISO.10646-1.1993].  Unicode allows for ASCII
          [ISO.646.1991] and many other international character
          sets to be used.  It is expected that strings will be
          encoded in UTF-8 format, which is identical in encoding
          for ASCII characters, but also accommodates other Unicode
          multi-byte characters.
        </documentation>
      </annotation>
    </enumeration>

    <enumeration value="dateTimeSeconds">
      <annotation>
        <documentation>
          The type "dateTimeSeconds" represents a time value
          in units of seconds normalized to the
          GMT time zone.
```

```
          </documentation>
        </annotation>
      </enumeration>

      <enumeration value="dateTimeMilliSeconds">
        <annotation>
          <documentation>The type "dateTimeMilliSeconds" represents
            a time value in units of milliseconds
            normalized to the GMT time zone.
          </documentation>
        </annotation>
      </enumeration>

      <enumeration value="dateTimeMicroSeconds">
        <annotation>
          <documentation>The type "dateTimeMicroSeconds" represents
            a time value in units of microseconds
            normalized to the GMT time zone.
          </documentation>
        </annotation>
      </enumeration>

      <enumeration value="dateTimeNanoSeconds">
        <annotation>
          <documentation>The type "dateTimeNanoSeconds" represents
            a time value in units of nanoseconds
            normalized to the GMT time zone.
          </documentation>
        </annotation>
      </enumeration>

      <enumeration value="ipv4Address">
        <annotation>
          <documentation>The type "ipv4Address" represents a value
            of an IPv4 address.
          </documentation>
        </annotation>
      </enumeration>

      <enumeration value="ipv6Address">
        <annotation>
          <documentation>The type "ipv6Address" represents a value
            of an IPv6 address.
          </documentation>
        </annotation>
      </enumeration>
    </restriction>
  </simpleType>
```

```
      <simpleType name="dataTypeSemantics">
        <restriction base="string">
          <enumeration value="quantity">
            <annotation>
              <documentation>
                A quantity value represents a discrete
                measured value pertaining to the record. This is
                distinguished from counters which represent an ongoing
                measured value whose "odometer" reading is captured as
                part of a given record. If no semantic qualifier is
                given, the Information Elements that have an integral
                data type should behave as a quantity.
              </documentation>
            </annotation>
          </enumeration>

          <enumeration value="totalCounter">
            <annotation>
              <documentation>
                An integral value reporting the value of a counter.
                Basically the same semantics as counters in SNMP.
                Counters are unsigned and wrap back to zero after
                reaching the limit of the type. For example, an
                unsigned64 with counter semantics will continue to
                increment until reaching the value of 2**64 - 1.  At
                this point the next increment will wrap its value to
                zero and continue counting from zero.  A running counter
                counts independently of the export of its value.
              </documentation>
            </annotation>
          </enumeration>

          <enumeration value="deltaCounter">
            <annotation>
              <documentation>
                An integral value reporting the value of a counter.
                Basically the same semantics as counters in SNMP.
                Counters are unsigned and wrap back to zero after
                reaching the limit of the type. For example, an
                unsigned64 with counter semantics will continue to
                increment until reaching the value of 2**64 - 1.  At
                this point the next increment will wrap its value to
                zero and continue counting from zero.  A delta counter
                is reset to zero each time its value is exported.
              </documentation>
            </annotation>
          </enumeration>
```

```
        <enumeration value="identifier">
          <annotation>
            <documentation>
              An integral value which serves as an identifier.
              Specifically mathematical operations on two
              identifiers (aside from the equality operation) are
              meaningless. For example, Autonomous System ID 1 *
              Autonomous System ID 2 is meaningless.
            </documentation>
          </annotation>
        </enumeration>

        <enumeration value="flags">
          <annotation>
            <documentation>
              An integral value which actually represents a set of
              bit fields.  Logical operations are appropriate on
              such values, but not other mathematical operations.
              Flags should always be of an unsigned type.
            </documentation>
          </annotation>
        </enumeration>
      </restriction>
    </simpleType>

    <simpleType name="applicability">
      <restriction base="string">
        <enumeration value="data">
          <annotation>
            <documentation>
              Used for Information Elements that are applicable to
              Flow Records only.
            </documentation>
          </annotation>
        </enumeration>

        <enumeration value="option">
          <annotation>
            <documentation>
              Used for Information Elements that are applicable to
              option records only.
            </documentation>
          </annotation>
        </enumeration>

        <enumeration value="all">
          <annotation>
            <documentation>
```

```
              Used for Information Elements that are applicable to
              Flow Records as well as to option records.
            </documentation>
          </annotation>
        </enumeration>
      </restriction>
    </simpleType>

    <simpleType name="status">
      <restriction base="string">
        <enumeration value="current">
          <annotation>
            <documentation>
              Indicates that the Information Element definition
              is that the definition is current and valid.
            </documentation>
          </annotation>
        </enumeration>

        <enumeration value="deprecated">
          <annotation>
            <documentation>
              Indicates that the Information Element definition is
              obsolete, but it permits new/continued implementation
              in order to foster interoperability with older/existing
              implementations.
            </documentation>
          </annotation>
        </enumeration>

        <enumeration value="obsolete">
          <annotation>
            <documentation>
              Indicates that the Information Element definition is
              obsolete and should not be implemented and/or can be
              removed if previously implemented.
            </documentation>
          </annotation>
        </enumeration>
      </restriction>
    </simpleType>

  <!--
    <simpleType name="enumRange">
      <restriction base="string"/>
    </simpleType>
  -->
```

```
<simpleType name="range">
  <restriction base="string"/>
</simpleType>

<complexType name="descriptionList">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="1"
             name="item" type="string">
      <annotation>
        <documentation>to be done ...</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="text" mixed="true">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0"
             name="paragraph" type="string">
      <annotation>
        <documentation>to be done ...</documentation>
      </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0"
             name="list" type="ipfix:descriptionList">
      <annotation>
        <documentation>to be done ...</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<element name="fieldDefinitions">
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" minOccurs="1" name="field">
        <complexType>
          <sequence>
            <element maxOccurs="1" minOccurs="1" name="description"
                     type="ipfix:text">
              <annotation>
                <documentation>
                  The semantics of this Information Element.
                  Describes how this Information Element is
                  derived from the Flow or other information
                  available to the observer.
                </documentation>
              </annotation>
```

```
                 </element>
   <!--
                 <element maxOccurs="1" minOccurs="0" name="usage"
                          type="ipfix:text">
                   <annotation>
                     <documentation>to be done ...</documentation>
                   </annotation>
                 </element>
   -->
                 <element maxOccurs="1" minOccurs="0" name="units"
                          type="string">
                   <annotation>
                     <documentation>
                       If the Information Element is a measure of some
                       kind, the units identify what the measure is.
                     </documentation>
                   </annotation>
                 </element>

                 <element maxOccurs="1" minOccurs="0" name="reference"
                          type="ipfix:text">
                   <annotation>
                     <documentation>
                       Identifies additional specifications which more
                       precisely define this item or provide additional
                       context for its use.
                     </documentation>
                   </annotation>
                 </element>

   <!--
                 <element maxOccurs="1" minOccurs="0"
                          name="enumeratedRange" type="ipfix:enumRange">
                   <annotation>
                     <documentation>
                       Some items may have a specific set of numeric
                       identifiers associated with a set of discrete
                       values this Information Element may take. The
                       meaning of each discrete value and a human
                       readable name should be assigned.
                     </documentation>
                   </annotation>
                 </element>
   -->

                 <element maxOccurs="1" minOccurs="0" name="range"
                          type="ipfix:range">
                   <annotation>
```

```
          <documentation>
            Some Information Elements may only be able to
            take on a restricted set of values which can be
            expressed as a range (e.g. 0 through 511
            inclusive). If this is the case, the valid
            inclusive range should be specified.
          </documentation>
        </annotation>
      </element>
    </sequence>

    <attribute name="name" type="string" use="required">
      <annotation>
        <documentation>
          A unique and meaningful name for the Information
          Element.
        </documentation>
      </annotation>
    </attribute>

    <attribute name="dataType" type="ipfix:dataType"
               use="required">
      <annotation>
        <documentation>
          One of the types listed in section 3.1 of this
          document or in a future extension of the
          information model. The type space for attributes
          is constrained to facilitate implementation.  The
          existing type space does however encompass most
          basic types used in modern programming languages,
          as well as some derived types (such as ipv4Address)
          which are common to this domain and useful
          to distinguish.
        </documentation>
      </annotation>
    </attribute>

    <attribute name="dataTypeSemantics"
               type="ipfix:dataTypeSemantics" use="optional">
      <annotation>
        <documentation>
          The integral types may be qualified by additional
          semantic details. Valid values for the data type
          semantics are specified in section 3.2 of this
          document or in a future extension of the
          information model.
        </documentation>
      </annotation>
```

```
               </attribute>

               <attribute name="elementId" type="nonNegativeInteger"
                          use="required">
                 <annotation>
                   <documentation>
                     A numeric identifier of the Information Element.
                     If this identifier is used without an enterprise
                     identifier (see below), then it is globally unique
                     and the list of allowed values is administered by
                     IANA.  It is used for compact identification of an
                     Information Element when encoding templates in the
                     protocol.
                   </documentation>
                 </annotation>
               </attribute>

               <attribute name="enterpriseId" type="nonNegativeInteger"
                          use="optional">
                 <annotation>
                   <documentation>
                     Enterprises may wish to define Information Elements
                     without registering them with IANA, for example for
                     enterprise-internal purposes. For such Information
                     Elements the Information Element identifier
                     described above is not sufficient when the
                     Information Element is used outside the enterprise.
                     If specifications of enterprise-specific
                     Information Elements are made public and/or if
                     enterprise-specific identifiers are used by the
                     IPFIX protocol outside the enterprise, then the
                     enterprise-specific identifier MUST be made
                     globally unique by combining it with an enterprise
                     identifier.  Valid values for the enterpriseId are
                     defined by IANA as SMI network management private
                     enterprise codes.  They are defined at
                     http://www.iana.org/assignments/enterprise-numbers.
                   </documentation>
                 </annotation>
               </attribute>

               <attribute name="applicability"
                          type="ipfix:applicability" use="required">
                 <annotation>
                   <documentation>This propoerty of an Information
                   Element indicates in which kind of records the
                   Information Element can be used.
                   Allowed values for this property are 'data',
```

```
                   'option', and 'all'.</documentation>
                 </annotation>
               </attribute>

               <attribute name="status" type="ipfix:status"
                          use="required">
                 <annotation>
                   <documentation>
                     The status of the specification of this
                     Information Element.  Allowed values are 'current',
                     'deprecated', and 'obsolete'.
                   </documentation>
                 </annotation>
               </attribute>

               <attribute name="group" type="string"
                          use="required">
                 <annotation>
                   <documentation>to be done ...</documentation>
                 </annotation>
               </attribute>

             </complexType>
           </element>
         </sequence>
       </complexType>

       <unique name="infoElementIdUnique">
         <selector xpath="field"/>

         <field xpath="infoElementId"/>
       </unique>
     </element>
   </schema>
```

Authors' Addresses

Juergen Quittek
NEC
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 6221 90511-15
Email: quittek@netlab.nec.de
URI: http://www.netlab.nec.de/

Stewart Bryant
Cisco Systems
250, Longwater, Green Park
Reading RG2 6GB
United Kingdom

Email: stbryant@cisco.com

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

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

Jeff Meyer
PayPal
2211 N. First St.
San Jose, CA 95131-2021
US

Phone: +1 408 976-9149
Email: jemeyer@paypal.com
URI: http://www.paypal.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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