Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 22, 2012

L. Braun
C. Schmitt
TU Muenchen
B. Claise
Cisco Systems, Inc.
G. Carle
TU Muenchen

September 21, 2011

Compressed IPFIX for smart meters in constrained networks
<draft-braun-core-compressed-ipfix-03>

Abstract

This document specifies the Compressed IPFIX protocol that serves for transmitting smart metering data in 6LoWPAN networks [RFC4944]. Compressed IPFIX is derived from IPFIX [RFC5101] and adopted to the needs of constrained networks. This documents specifies how the Compressed IPFIX Data and Template Records are transmitted in 6LoWPAN networks and how Compressed IPFIX data can be converted into uncompressed IPFIX data in a proxy device.

Table of Contents

1. Introduction

Smart meters that form a constrained wireless network need an application layer protocol that allows the efficient transmission of metering data from the devices to some kind of central analysis device. The meters used to build such networks are usually equipped with low-cost and low-power hardware. This leads to constraints in computational capacities, available memory and networking resources.

The devices are often battery powered and are expected to run for a long time without having the possibility to re-charge themselves. In order to save energy, smart meters often power off their wireless networking device. Hence, they don't have a steady network connection, but are only part of the wireless network as needed when there is data that needs to be exported. A push protocol like Compressed IPFIX, where data is transmitted autonomically from the meters to one or more collectors, is suitable for reporting metering data in such networks.

Compressed IPFIX is derived from IPFIX [RFC5101] and therefore inherits most of its properties. One of these properties is the separation of data and its data description by encoding the former in Data Sets and the latter in Template Sets.

Transforming Compressed IPFIX to IPFIX as per [RFC5101] is very simple and can be done on the border between the constrained network and the more general network. The transformation between one form of IPFIX data into another is known as IPFIX Mediation [RFC5982]. Hence, smart metering networks that are based on Compressed IPFIX can be easily integrated into an existing IPFIX measurement infrastructure.

1.1. Document structure

Section 2 introduces the used terminology in this draft. Afterwards, hardware and software constraints in constrained networks, which will motivate our modifications to the IPFIX protocol, are discussed in Section 3. Section 4 describes the application scenarios and Section 5 describes the architecture for Compressed IPFIX. Section 6 defines the Compressed IPFIX protocol itself and discusses the differences between Compressed IPFIX and IPFIX. The Mediation Process from Compressed IPFIX to IPFIX is described in Section 7. Section 8 defines the process of Template Management on the Exporter and the Collector. Section 9 and Section 10 discuss the security and IANA considerations for Compressed IPFIX. Section 11 lists the open issues that need to be addressed in further versions of this draft.

2. Terminology

The term smart meter is used to refer to constrained devices like wireless senor nodes, motes or any other kind of small constraint device that can be part of a network that is based on IEEE802.15.4 and 6LoWPAN [RFC4944].

Most of the terms used in this draft are defined in [RFC5101]. All these terms are written with their first letter being capitalized. Most of the terms that are defined for IPFIX can be used to describe Compressed IPFIX. The term "Compressed" is used in front of the IPFIX term to distinguish between the IPFIX version and the Compressed IPFIX version. This draft uses the term IPFIX to refer to IPFIX as per RFC 5101 and the term Compressed IPFIX for the IPFIX version defined in this draft.

The terms IPFIX Message, IPFIX Device, Set, Data Set, Template Set, Data Record, Template Record, Collecting Process, Collector, Exporting Process and Exporter are defined as in [RFC5101]. The term IPFIX Mediator is defined in [RFC5982]. The terms Intermediate Process, IPFIX Proxy, IPFIX Concentrator are defined in [I-D.ietf-ipfix-mediators-framework].

All these terms above have been adapted from the IPFIX definitions. As they keep a similar notion but in a different context of constrained networks, the term "Compressed" now complements the defined terms.

Compressed Exporting Process

The Compressed Exporting Process is a process that exports Compressed Records.

Compressed Exporter

A Compressed Exporter is a smart metering device that contains at least one Compressed Exporting Process.

Compressed Collecting Process

The Compressed Collecting Process is a process inside a device that is able to receive and process Compressed Records.

Compressed IPFIX Collector

A Compressed Collector is a device that contains at least one Compressed Collecting Process.

Compressed IPFIX Device

A Compressed IPFIX Device is a device that contains one or more Compressed Collector or one or more Compressed Exporter.

Compressed IPFIX Smart Meter

A Compressed IPFIX Smart Meter is a device that contains the functionality of an Compressed IPFIX device. It is usually equipped with one or more sensors that meter a physical quantity, like power consumption, temperature, or physical tempering with the device. Every Compressed IPFIX Smart Meter MUST at least contain an Compressed Exporting Process. It MAY contain a Compressed Collecting Process in order to work as a Compressed IPFIX Proxy or Concentrator.

Compressed IPFIX Message

The Compressed IPFIX Message is a message originated by an Compressed IPFIX Exporter. It is composed of a Compressed Message Header and one or more Compressed Sets. The Compressed IPFIX Message Format is defined in Section 6.

Compressed Data Record

A Compressed Data Record equals a Data Record in [RFC5101]. The term is used to distinguish between IPFIX and Compressed IPFIX throughout the document.

Compressed Template Record

A Compressed Template Record is similar to a Template Record. The Template Record Header is substituted with a Compressed Template Record Header and is otherwise equal to a Template Record. Section 6.3.

Compressed Set

The Compressed Set is a group of Compressed Data Records or Compressed Template Records with a Compressed Set Header. Its format is defined in Section 6.2.

Compressed Data Set

The Compressed Data Set is a Compressed Set that contains Compressed Data Records. in Section 6.2.

Compressed Template Set

A Compressed Template Set is a Compressed Set that contains Compressed Template Records.

Compressed Intermediate Process

A Compressed Intermediate Process is an Intermediate Process that can handle Compressed IPFIX Messages.

Compressed IPFIX Proxy

A Compressed IPFIX Proxy is an IPFIX Proxy that can handle Compressed IPFIX Messages.

Compressed IPFIX Concentrator

A Compressed IPFIX Concentrator is an IPFIX Concentrator that can handle Compressed IPFIX Messages.

A Compressed IPFIX Transport Session is defined by the communication between a Compressed Exporter (identified by an 6LowPAN-Address, the Transport Protocol, and the Transport Port) and a Compressed Collector (identified by the same properties).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Hard- and Software constraints

3.1. Hardware constraints

The target devices for Compressed IPFIX are usually equipped with low-cost hardware and therefore face several constraints concerning CPU and memory [Schmitt09]. For example, the IRIS mote from Crossbow Technologies Inc. has a size of 58 x 32 x 7 mm (without a battery pack) [Crossbow]. Thus, there is little space for micro controller, flash memory (128 kb) and radio frequency transceiver, which are located on the board.

Network protocols used on such hardware need to respect these constraints. They must be simple to implement using little code and little run time memory and should produce little overhead when encoding the application payload.

3.2. Energy constraints

Smart meters that are battery powered have hard energy constraints [Schmitt09]. By power supply of two 2 AA 2,800-mAh batteries this means approximately 30,240J. If they run out of power, their battery Has to be changed, which means physical manipulation to the device is necessary. Using as little energy as possible for network communication is therefore desired.

A smart metering device can save a lot of energy, if it powers down its radio frequency transceiver. Such devices do not have permanent network connectivity but are only part of the network as needed. A push protocol, where only one side is sending data, is suitable for transmitting application data under such circumstances. As the communication is unidirectional, a meter can completely power down its radio frequency transceivers as long as it does not have any data to sent. If the metering device is able to keep a few measurements in memory, and if real time metering is not a requirement, the Compressed Data Records can be pushed less frequently. Therefore, saving some more energy on the radio frequency transceivers.

3.3. Packet size constraints

Compressed IPFIX is mainly targeted for the use in 6LoWPAN networks, which are based on IEEE 802.15.4 [RFC4944]. However, the protocol can also be used to transmit data in other networks. IEEE 802.15.4 defines a maximum frame size of 127 octets, which usually leaves 102 octets for user data. IPv6 on the other hand defines a minimum MTU of 1280 octets. Hence, fragmentation has to be implemented in order to transmit such large packets. While fragmentation allows the transmission of large messages, its use is problematic in networks with high packet loss because the complete message has to be discarded if only a single fragment gets lost.

Compressed IPFIX enhances IPFIX by a header compression scheme, which allows to reduce the overhead from header sizes significantly. Additionally, the overall Compressed IPFIX Message size is reduced, which reduces the need for fragmentation.

3.4. Transport protocol constraints

The IPFIX standard [RFC5101] defines several transport protocol bindings for the transmission of IPFIX Messages. SCTP support is REQUIRED for any IPFIX Device to achieve standard conformance [RFC5101], and its use is highly recommended. However, sending IPFIX over UDP and TCP MAY also be implemented.

This transport protocol recommendation is not suitable for Compressed IPFIX. A header compression scheme that allows to compress an IPv6

header from 40 octets down to 2 octets is defined in 6LoWPAN. There is a similar compression scheme for UDP, but there is no such compression for TCP or SCTP headers. If header compression can be employed, more space for application payload is available.

Using UDP on the transport layer for transmitting IPFIX Messages is therefore highly recommended. Furthermore, TCP or SCTP are currently not supported on some platforms, like on TinyOS [Harvan08]. Hence, UDP may be the only option.

Every Compressed IPFIX Exporter and Collector MUST implement UDP transport layer support for transmitting data in a constrained network environment. It MAY also offer TCP or SCTP support. However, using these protocols is NOT RECOMMENDED as their use will consume more power and reduces the available size of application payload compared to the use of UDP. If Compressed IPFIX is transmitted over a non-constrained network, using SCTP as a transport layer protocol is RECOMMENDED.

4. Application scenarios for Compressed IPFIX

Compressed IPFIX is derived from IPFIX [RFC5101] and is therefore a unidirectional push protocol. This means all communication that employs Compressed IPFIX is unidirectional from an Exporting Process to a Collecting Process. Hence, Compressed IPFIX only fits for application scenarios where meters transmit data to one or more Collectors.

If Compressed IPFIX is used over UDP, as recommended, packet loss can occur. Furthermore, if an initial Template Message gets lost, and is therefore unknown to the Collector, all Data Sets that reference this Template cannot be decoded. Hence, all these Messages are lost if they are not cached by the Collector. It should be clear to an application developer, that Compressed IPFIX can only be used over UDP if these Compressed IPFIX Message losses are not a problem.

Compressed IPFIX over UDP is especially not a suitable protocol for applications where sensor data trigger policy decisions or configuration updates for which packet loss is not tolerable.

Applications that use smart sensors for accounting purposes for long time measurements can benefit from the use of Compressed IPFIX. One application for IPFIX can be long term monitoring of large physical volumes. In [Tolle05], Tolle et al. built a system for monitoring a "70-meter tall redwood tree, at a density interval of 5 minutes in time and 2 meters in space". The sensor node infrastructure was deployed to measure the air temperature, relative humidity and

photosynthetically active solar radiation over a long time period.

Deploying Compressed IPFIX in such scenarios seems to be a good fit. The sensors of the IPFIX Smart Meter can be queried over several 5 minute time intervals and the query results can be aggregated into a single Compressed IPFIX Message. As soon as enough query results are stored in the Compressed IPFIX Message, e.g. if the Compressed IPFIX Message size fills the available payload in a single IEEE 802.15.4 packet, the wireless transceiver can be activated and the Compressed IPFIX Message can be exported to a Compressed IPFIX Collector.

Similar sensor networks have been built to monitor the habitat of animals, e.g. in the "Great Duck Island Project" [GreatDuck], [SMPC04]. The purpose of the sensor network was to monitor the birds by deploying sensors in and around their burrows. The measured sensor data was collected and stored in a database for offline analysis and visualization. Again, the sensors can perform their measurements periodically, aggregate the sensor data and export them to a Compressed IPFIX Collector.

Other application scenarios for Compressed IPFIX could be applications where sensor networks are used for long term structural health monitoring in order to investigate long term weather conditions on the structure of a building. For example, a smart metering network has been built to monitor the structural health of the Golden Gate Bridge [Kim07]. If a sensor network is deployed to perform a long term measurement of the structural integrity, Compressed IPFIX can be used to collect the sensor measurement data.

If an application developer wants to decide whether to use Compressed IPFIX for transmitting data from smart meters, he must take the following considerations into account:

1. The application must require a push protocol. It is not possible to request data from a smart meter. The IPFIX Smart Meter decides for itself when to send its metering data.
2. The property above allows a IPFIX Smart Meter to turn off its wireless device in order to save energy, as it does not have to receive any data.
3. If real-time is not required, the application might benefit from accumulated several measurements into a single Compressed IPFIX Message. Compressed IPFIX easily allows the aggregation of several into a single Compressed IPFIX Message (or a single packet). This aggregation can happen on the IPFIX Smart Meter that aggregates several of its own measurements. Or it can happen within a multi-hop wireless network where one IPFIX Proxy aggregates several Compressed IPFIX Messages into a single Compressed IPFIX Message before forwarding them.

4. The application must accept potential packet loss. Compressed IPFIX only fits for applications where metering data is stored for accounting purposes and not for applications where the sensor data triggers configuration changes or policy decisions (except: if Message loss is acceptable for some reason).

5. Architecture for Compressed IPFIX

The Compressed IPFIX architecture is similar to the IPFIX architecture which is described in [RFC5470]. The most common deployment of IPFIX Smart Meters is shown in Figure 1.

```
                            +-----------------+     +-----------------+
                            |[*Application 1]| ... |[*Application n]|
                            +--------+--------+     +-------+---------+
                                    ^                      ^
                                    |                      |
                                    + = = = = -+- = = = = +
                                               ^
                                               |
+-------------------------+ Compressed +---------+-------------------+
| Compressed IPFIX S.M.   |   IPFIX    | Compressed IPFIX Collector |
|   [Exporting Process]   |----------->|  [Collecting Process(es)]  |
+-------------------------+            +----------------------------+
```

Figure 1: Direct transmission between sensors and applications

An IPFIX Smart Meter (S.M.) queries its internal sensors to retrieve the sensor data. It then encodes the results into a Compressed IPFIX Message and exports this Compressed IPFIX Message to one or more Compressed IPFIX Collectors. The Compressed IPFIX Collector runs one or more applications that process the collected sensor data. The Compressed IPFIX Collector can be deployed on non-constrained devices at the constrained network border.

A second way to deploy IPFIX Smart Meter can employ aggregation on Compressed IPFIX Messages during their journey through the constrained network as shown in Figure 2. This aggregation can be performed by special IPFIX Smart Meter that act as Compressed IPFIX Concentrators. Such devices must have enough resources to perform the aggregation.

```
 +-------------------------+                  +------------------------+
 | Compressed IPFIX S.M.   | Compressed IPFIX |Comp. IPFIX Concentrator|
 |   [Exporting Process]   |----------------->|  [Collecting  Process] |
 +-------------------------+        +-------->|  [Exporting Process]   |
                                    |         +------------------------+
 +-------------------------+        |                     |
 | Compressed IPFIX S.M.   |        |     Compressed IPFIX|
 |   [Exporting Process]   |--------+                     |
 +-------------------------+                              v
                                            +-------+------------------+
                                            |      Collector(1)        |
                                            | [Collecting Process(es)] |
                                            +--------------------------+
```

Figure 2: Aggregation on Compressed IPFIX

IPFIX Smart Meters send their data to Compressed IPFIX Concentrator which needs to have enough storage space to store the incoming data. It may also aggregate the incoming data with its own measurement data. The aggregated data can then be re-exported again to one or more Collectors.

The last deployment, shown in Figure 3, employs another Compressed IPFIX Mediation process.

```
  +------------------------+                  +------------------------+
  | Compressed IPFIX S.M   | Compressed IPFIX | Compressed IPFIX Proxy |
  |  [Exporting Process]   |----------------->|  [Collecting Process]  |
  +------------------------+                  |  [Exporting Process]   |
                                              +------------------------+
                                                         |
                                                 IPFIX   |
                                                         |
                                                         v
                                            +-------+------------------+
                                            |    IPFIX Collector(1)    |
                                            | [Collecting Process(es)] |
                                            +--------------------------+
```

Figure 3: Aggregation on Compressed IPFIX

The IPFIX Smart Meters transmit their Compressed IPFIX Messages to one node, e.g. the base station, which translates the Compressed IPFIX Messages to IPFIX Messages. The IPFIX Messages can then be exported into an existing IPFIX infrastructure. The Mediation

process from Compressed IPFIX to IPFIX is described in Section 7.

6. Compressed IPFIX Message Format

A Compressed IFPIX Message starts with a Compressed Message Header, followed by one or more Compressed Sets. The Compressed Sets can be any of the possible two types: Compressed Template Set and Compressed Data Set. An Compressed IPFIX Message MUST only contain one type of Compressed Set. The format of the Compressed IPFIX Message is shown in Figure 4

```
+----------------------------------------------------+
| Compressed Message Header                          |
+----------------------------------------------------+
| Compressed Set                                     |
+----------------------------------------------------+
| Compressed Set                                     |
+----------------------------------------------------+
  ...
+----------------------------------------------------+
| Compressed Set                                     |
+----------------------------------------------------+
```

Figure 4: Compressed IPFIX Message Format

6.1. Compressed IPFIX Message Header

The Compressed IPFIX Message Header is derived from the IPFIX Message Header, with some optimization using field compression. The IPFIX Message Header from [RFC5101] is shown in Figure 5.

```
 0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version Number              |         Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Export Time                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sequence Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Observation ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: IPFIX Message Header

The length of the IPFIX Message Header is 16 octets and every IPFIX Message has to be started with it. The Compressed IPFIX Message Header needs to be smaller due to the packet size constraints discussed in Section 3.3. Compressed IPFIX introduces a Compressed IPFIX Message Header that has a smaller size. The Compressed header consists of a fixed part of two octets and a variable length "Remaining Header" as shown in Figure 6.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|ETC|SNC|     Length      |
|Number |   |   |                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Remaining Header          |
|                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Format of the Compressed IPFIX Message header

The first part has a fixed length of two octets and consists of the "Version Field" (4 bit), the "Export Time Compression" (ETC) field (2 bit), the "Sequence Number Compression" (SNC) field (2 bit) and the "Length" field (8 bit). The second part (the "Remaining Header") has a variable length. Its length is defined by the ETC and SNC fields in the fixed header.

The fixed header has a length of two octets which equals the length of the version field of the IPFIX Message Header. Hence, Compressed IPFIX Messages can be read and identified by an IPFIX Collector. This is important for building an IPFIX Mediator by extending an IPFIX Collector (Section 7).

The fixed header fields are defined as follows:

Version number

The Compressed IPFIX version field MUST have the most significant bit set to one and the other bits set to zero. The remaining bits of the version field are reserved for future versions of Compressed IPFIX. Note that IPFIX has the version 0x000a, hence an IPFIX Collector can distinguish between IPFIX and Compressed IPFIX by checking the first bit of the version field.

ETC

The ETC field defines the compression level of the "Export Time" field of the IPFIX Messages Header. Its value defines the length as follows. A bit sequence of "00" denotes that the "Export Time" field is omitted. A sequence of "01" denotes that the "Export Time" field has a size of one octet. A sequence of "10" denotes that the "Export Time" field has a size of two octets. Finally, a sequence of "11" denotes that the "Export Time" field has the original length of four octets.

SNC

The SNC field defines the compression level of the "Sequence Number" field of the IPFIX Messages Header. Its value defines the length as follows. A bit sequence of "00" denotes that the "Sequence Number" field is omitted. A sequence of "01" denotes that the "Sequence Number" field has a size of one octet. A sequence of "10" denotes that the "Sequence Number" field has a size of two octets. Finally, a sequence of "11" denotes that the "Sequence Number" field has the original length of four octets.

Length

The length field has a fixed length of one octet. Compressed IPFIX Messages therefore have a maximum length of 255 octets.

An application SHOULD never send a Compressed IPFIX that is bigger than 102 octets to avoid fragmentation. If the "Export Time" field is not omitted, it is placed directly behind the length field. If the Export Time field has a size of four octets, it MUST contain the time in seconds since 0000 UTC Jan 1, 1970, at which the Compressed IPFIX Message Header leaves the Exporter. This complies with the "Export Time" field in IPFIX.

Afterwards, the "Sequence Number" field is attached (if not omitted). If the field has a length of four bytes, it must contain the number of records sent since the start of the Exporter module 2^32 at the end of this Compressed IPFIX Message. If the field is Compressed to one or two bytes, it must contain the number of IPFIX messages sent by the Exporter since its start modulo 2^8 or 2^16.

6.2. Compressed Set

A Compressed Set is a set of Compressed Template or Compressed Data Records. Depending on the Compressed Record type, the Compressed Set can either be a Compressed Template Set or a Compressed Data Set. Every Compressed Set is started with an Compressed Set Header and is

followed by one or more Compressed Records.

The IPFIX Set Header consists of an two octet "Set ID" field and a two octet "Length" field. These two fields are compressed to one octet each for the Compressed Set Header. The format of the Compressed Set Header is shown in Figure 7.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Comp. Set ID  |     Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Compressed Set Header

The two fields are defined as follows:

Compressed Set ID

The "Compressed Set ID" (Comp. Set ID) identifies the type of data that is transported in the Compressed Set. A Compressed Template Set is identified by Compressed Set ID 2. This corresponds to the Set IDs that are used by Sets in IPFIX. Compressed Set ID number 3 MUST NOT be used. All values from 4 to 127 are reserved for future use. Values above 127 are used for Compressed Data Sets.

Length

The "Length" Field contains the total length of the Compressed Set, including the Compressed Set Header.

6.3. Compressed Template Record Format

The format of the Compressed Template Records is shown in Figure 8. The Compressed Template Record starts with an Compressed Template Record Header and is followed by one or more Field Specifiers. The Field Specifier format is defined as in Section 6.4 and is identical to the Field Specifier definition in [RFC5101].

```
+--------------------------------------------------+
| Compressed Template Record Header                |
+--------------------------------------------------+
| Field Specifier                                  |
+--------------------------------------------------+
| Field Specifier                                  |
+--------------------------------------------------+
 ...
+--------------------------------------------------+
| Field Specifier                                  |
+--------------------------------------------------+
```

Figure 8: Compressed Template Format

The format of the Compressed Template Record Header is shown in Figure 9.

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Comp. Temp ID |  Field Count  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Compressed Template Header

Compressed Template ID

Each Compressed Template Record must have a unique Compressed Template ID (Comp. Temp ID) between 128 and 255. The Compressed Template ID must be unique for the given Compressed Transport Session.

Field Count

The number of fields placed in the Compressed Template Record.

6.4. Field Specifier Format

The type and length of the transmitted data is encoded in Field Specifiers within Compressed Template Records. The Field Specifier is shown in Figure 10 and is identical with the Field Specifier that was defined for IPFIX [RFC5101]. The following text has been copied from [RFC5101] for completeness.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E|  Information Element ident. |        Field Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Enterprise Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Compressed Template Header

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. If this bit is zero, the Information Element Identifier identifies an IETF-specified Information Element, and the four-octet Enterprise Number field MUST NOT be present. If this bit is one, the Information Element Identifier identifies an enterprise-specific Information Element, and the Enterprise Number field MUST be present.

Information Element Identifier

A numeric value that represents the type of Information Element.

Field Length

The length of the corresponding encoded Information Element, in octets. Refer to [RFC5102]. The value 65535 is illegal as there are no variable size encoded elements as they are defined in IPFIX.

Enterprise Number

IANA [IANA] enterprise number of the authority defining the Information Element identifier in this Template Record.

Vendors can easily define their own data model by registering a Enterprise ID with IANA. Using their own Enterprise ID, they can use any ID in the way they want them to use.

6.5. Compressed Data Record Format

The Data Records are sent in Compressed Data Sets. The format of the Data Records is shown in Figure 11 and matches the Data Record format from IPFIX.

```
      +---------------------------------------------------+
      | Field Value                                       |
      +---------------------------------------------------+
      | Field Value                                       |
      +---------------------------------------------------+
       ...
      +---------------------------------------------------+
      | Field Value                                       |
      +---------------------------------------------------+
```

Figure 11: Data Record Format

7. Compressed IPFIX Mediation

There are two types of Compressed IPFIX Intermediate Processes. The first one can occur on the transition between a constraint 6LoWPAN and the non-constrained network. This mediation changes the network and transport protocol from 6LowPAN/UDP to IP/(SCTP|TCP|UDP) and is shown in Figure 12.

```
   +------------------------+Compressed IPFIX+-------------------------+
   |Compressed IPFIX S.M.   |     6LoWPAN/UDP |Compressed IPFIX mediator|
   | [Exporting Process]    |---------------->|   [Collecting Process]  |
   +------------------------+                 |   [Exporting Process]   |
                                              +-------------------------+
                                                           |
                                   Compressed IPFIX        |
                                   IP/(UDP/SCTP|TCP)       |
                                                           v
                                             +-------+-----------------+
                                             |      Collector(1)       |
                                             | [Collecting Process(es)] |
                                             +-------------------------+
```

Figure 12: Translation from Compressed IPFIX over 6LowPAN/UDP to Compressed IPFIX over IP/(SCTP|TCP|UDP)

The mediator removes the Compressed IPFIX Messages from the 6LowPAN/UDP packets and wraps them into the new network and transport protocols. Templates MUST be managed the same way as in the constraint environment after the translation to IP/(SCTP|UDP|TCP) (see Section 8).

The second type of mediation transforms Compressed IPFIX into IPFIX.

This process MUST be combined with the transport protocol mediation as shown in Figure 13.

```
+-------------------------+ Compressed IPFIX+------------------------+
| Compressed IPFIX S.M.  |    6LoWPAN/UDP  |    IPFIX mediator      |
|[Exporting Processes]   |---------------->| [Collecting Process]  |
+-------------------------+                 | [Exporting Process]   |
                                            +------------------------+
                                                       |
                                         IPFIX         |
                                   IP/(UDP/SCTP|TCP)   |
                                                       v
                                         +-------+-----------------+
                                         |      Collector(1)       |
                                         | [Collecting Process(es)] |
                                         +--------------------------+
```

Figure 13: Transformation from Compressed IPFIX to IPFIX

This mediation can also be performed by an IPFIX Collector before parsing the IPFIX message as shown in Figure 14. There is no need for a Compressed IPFIX parser if such a mediation process can be employed in front of an already existing IPFIX collector.

```
+-------------------------+ Compressed IPFIX +-----------------------+
| Compressed IPFIX S.M.  |    6LoWPAN/UDP    |     IPFIX Mediator   |
| [Exporting Processes]  |------------------>| [Collecting Process] |
+-------------------------+                  |  [Exporting Process] |
                                             |         |            |
                                             |         |IPFIX       |
                                             |         |            |
                                             |         v            |
                                             |   Collector(1)       |
                                             | [Collecting Process] |
                                             +-----------------------+
```

Figure 14: Transformation from Compressed IPFIX to IPFIX

The Compressed Mediation Process has to translate the Compressed IPFIX Message Header, the Compressed Set Headers and the Compressed Template Record Header into their counterparts in IPFIX Afterwards, the new IPFIX Message Length needs to be calculated and inserted into the IPFIX Message header.

7.1. Expanding the Message header

The fields of the IPFIX Message Header that are shown in Figure 5 can be determined as follows:

Version

This is always 0x000a.

Length

The IPFIX Message Length can only be calculated after the complete Compressed IPFIX Message has been translated. The new length can be calculated by adding the length of the IPFIX Message Header, which is 16 octets, and the length of all Sets that are contained in the IPFIX Message.

Export Time

If the "Export Time" in the Compressed IPFIX Message Header has a length of 4 octets, the value from the Compressed Message Header MUST be used for the IPFIX Message Header. If it was omitted, the "Export Time" MUST be generated by the Mediator. If the IPFIX Message is exported again, the "Export Time" field MUST contain the time in seconds since 0000 UTC Jan 1, 1970, at which the IPFIX Message leaves the Exporter. If the Message is passed to an IPFIX Collector for decoding directly, the "Export Time" field is the time in seconds since 0000 UTC Jan 1 1970 at which the Compressed IPFIX Message has been received by the Compressed IPFIX Exporter.

Sequence Number

If the Compressed Sequence Number has a length of 4 octets, the original value MUST be used for the IPFIX Message. If the Compressed Sequence Number has a size of one or two octets, the Compressed IPFIX Mediator MUST expand the Compressed Sequence Number into a four octet field. If the Compressed Sequence Number was omitted, the Mediator needs to calculate the Sequence Number as per RFC 5101 [RFC5101].

Observation Domain ID

This is always 0 indicating to the IPFIX Collector, that the Observation Domain ID is not relevant.

7.2. Translating the Set Headers

Both fields in the Compressed Set Header have a size of one octet and need to be expanded:

Set ID

The field needs to be expanded from one octet to two octets. If the Set ID is below 128, no recalculation needs to be performed. This is because all IDs below 128 are reserved for special messages and match the IDs used in IPFIX. The Compressed Set IDs starting with 128 identify Data Sets. Therefore, every Compressed Set ID above 127 needs to be incremented by 128 because IPFIX Data Set IDs are located above 255.

Set Length

The field needs to be expanded from one octet to two octets. It needs to be recalculated by adding a value of 2 octet to match the additional size of the Set Header. For each Compressed Template Record that is contained in the Compressed Set, 2 more octets need to be added to the length.

7.3. Expanding the Template Record Header

Both fields in the Compressed Template Record Header have a length of one octet and therefore need translation:

Template ID

The field needs to be expanded from one octet to two octets. The Template ID needs to be increased by a value of 128.

Field Count

The field needs to be expanded from one octet to two octets.

8. Template Management

The way Compressed Templates have to be managed depends on the usd transport protocol. If TCP or SCTP is used, it can be ensured that Compressed Templates are delivered reliably. Template loss can occur on UDP on the other hand. If a Template is lost on its way to the Collector, all following Compressed Data Records that refer to this Compressed Template cannot be decoded.

8.1. TCP / SCTP

If TCP or SCTP is an option and can be used for the transmission of Compressed IPFIX, Template Management MUST be performed as defined in [RFC5101] for IPFIX.

8.2. UDP

All specifications for Template management from [RFC5101] apply unless specified otherwise in this document.

Compressed Templates MUST be sent by an Compressed Exporter before any Compressed Data Set that refers to the Compressed Template is transmitted. Compressed Templates are not expected to change over time in Compressed IPFIX. Hence, a Compressed Template that has been sent once MAY NOT be withdrawn and MUST NOT expire. If an IPFIX Smart Meter wants to use another Compressed Template it MUST use a new Compressed Template ID for the Compressed Template.

As UDP is used, reliable transport of Compressed Templates cannot be guaranteed and Compressed Templates can be lost. A Compressed Exporter MUST expect Compressed Template loss. It MUST therefore re-send its Compressed Templates periodically. A Compressed Template MUST be re-send after a fixed number of N Compressed IPFIX Messages that contained Compressed Data Sets that referred to this Compressed Template. The number N MUST be configured by the application developer.

9. Security considerations

The same security considerations as for the IPFIX Protocol [RFC5101] apply.

10. IANA Considerations

The same IANA considerations as for the IPFIX Protocol [RFC5101] apply.

11. Open Issues

1. Export Time field value if the field is compressed to one or two bytes is unclear.
2. It is unclear how reserved IPFIX Set IDs above 127 can be handled, if they are standardized some day

3. Translating the one or two byte long Sequence Number from Compressed IPFIX to IPFIX has some pitfalls when packet loss occurs.
4. Option Templates need to be defined (they are forbidden right now)
5. Section on the Collector side (as in RFC 5101) is needed.

12. References

12.1. Norminative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

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

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

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.

[RFC5982] Kobayashi, A. and B. Claise, "IP Flow Information Export (IPFIX) Mediation: Problem Statement", RFC 5982, August 2010.

[I-D.ietf-ipfix-mediators-framework]
Kobayashi, A., Claise, B., Muenz, G., and K. Ishibashi, "IPFIX Mediation: Framework", draft-ietf-ipfix-mediators-framework-09 (work in progress), October 2010.

[I-D.shelby-core-coap]
Shelby, Z., Frank, B., and D. Sturek, "Constrained Application Protocol (CoAP)", draft-shelby-core-coap-01

(work in progress), May 2010.

12.2. Informative References

[IANA] "IANA Private Enterprise Numbers registry
http://www.iana.org/assignments/enterprise-numbers.".

[Schmitt09]
Schmitt, C. and G. Carle, "Applications for Wireless Sensor Networks", In Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications, Antonopoulos N.; Exarchakos G.; Li M.; Liotta A. (Eds.), Information Science Publishing. , 2010.

[Tolle05] Tolle, G., Polastre, J., Szewczyk, R., Turner, N., Tu, K., Buonadonna, P., Burgess, S., Gay, D., Hong, W., Dawnson, T., and D. Culler, "A macroscope in the redwoods", In the Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (Sensys 05), San Diego, ACM Press , November 2005.

[Kim07] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and M. Turon, "Health Monitoring of Civil Infrastructure Using Wireless Sensor Networks", In the Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN 2007), Cambridge, MA, ACM Press, pp. 254-263 , April 2007.

[SMPC04] Szewczyk, R., Mainwaring, A., Polastre, J., and D. Culler, "An analysis of a large scale habitat monitoring application", The Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 04) , November 2004.

[GreatDuck]
Habitat Monitoring on Great Duck Island, "http://www.greatduckisland.net", The Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 04) , November 2004.

[Harvan08]
Harvan, M. and J. Schoenwaelder, "TinyOS Motes on the Internet: IPv6 over 802.15.4 (6lowpan)", 2008.

[Crossbow]
Crossbow Technologies Inc., "http://www.xbow.com", 2010.

Authors' Addresses

Lothar Braun
Technische Universitaet Muenchen
Department of Informatics
Chair for Network Architectures and Services (I8)
Boltzmannstr. 3
Garching 85748
Germany

Email: braun@net.in.tum.de
URI: http://www.net.in.tum.de/~braun

Corinna Schmitt
Technische Universitaet Muenchen
Department of Informatics
Chair for Network Architectures and Services (I8)
Boltzmannstr. 3
Garching 85748
Germany

Email: schmitt@net.in.tum.de
URI: http://www.net.in.tum.de/~schmitt

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

Email: bclaise@cisco.com

Georg Carle
Technische Universitaet Muenchen
Department of Informatics
Chair for Network Architectures and Services (I8)
Boltzmannstr. 3
Garching 85748
Germany

Email: carle@net.in.tum.de
URI: http://www.net.in.tum.de/~carle