

Network Working Group
Internet-Draft
Expires: September 6, 2007

A. Fessi
G. Carle
University of Tuebingen
F. Dressler
University of Erlangen
J. Quittek
NEC
C. Kappler
H. Tschofenig
Siemens AG
March 5, 2007

NSLP for Metering Configuration Signaling
<[draft-dressler-nsis-metering-nslp-05.txt](#)>

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>.

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

This Internet-Draft will expire on September 6, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Monitoring, metering and accounting of packets are increasingly

important functionality that needs to be provided in the Internet. This document proposes the definition of a new NSIS Signaling Layer Protocol (NSLP), named Metering NSLP (M-NSLP), which allows the dynamic configuration of Metering Entities on the data path. An accompanying document [[I-D.fessi-nsis-m-nslp-framework](#)] provides a problem statement, describes scenarios where such a path-coupled configuration of Metering Entities is useful, elaborates requirements for a path-coupled protocol for the configuration of Metering Entities and discusses the applicability of NSIS for this purpose. This document defines a Metering NSLP protocol design and messages, and describes the protocol operation.

Table of Contents

1.	Introduction	5
2.	Terminology	5
3.	Protocol Overview	7
3.1.	Model of operation	7
3.2.	Metering NSLP Messages	9
3.2.1.	CONFIGURE	9
3.2.2.	REFRESH	9
3.2.3.	OPTIONS	9
3.2.4.	RESPONSE	10
3.2.5.	NOTIFY	10
4.	Design Decisions	10
4.1.	Soft State	10
4.2.	Message Sequencing	10
4.3.	Message Scoping	10
4.4.	Selection of MNEs	11
4.5.	Coordination of Metering Tasks	11
4.6.	Aggregation	11
4.7.	Reaction to Route Changes	11
4.8.	Metering Configuration Information	12
4.9.	Required State Information	12
5.	Metering NSLP Messages and Objects	13
5.1.	Metering NSLP Objects	13
5.1.1.	The 'Session Identifier' Object (SID)	13
5.1.2.	The 'Message Sequence Number' Object (MSN)	13
5.1.3.	The 'Selection of Metering Entities' Object (MNE_Selection)	13
5.1.4.	The 'Session Lifetime' Object (Session_LT)	14
5.1.5.	The 'MNSLP Hop Count' Object (MNSLP_Hop_Count)	14
5.1.6.	The 'Information Code' Object (INFO)	14

5.1.7.	MSPEC Objects	15
5.2.	Metering NSLP Messages	15
5.2.1.	CONFIGURE	15
5.2.2.	REFRESH	15
5.2.3.	OPTIONS	15
5.2.4.	RESPONSE	16
5.2.5.	NOTIFY	16
5.3.	MSPEC Objects	16
5.3.1.	IPFIX MSPEC Object	17
6.	Processing Rules	18
6.1.	Session State Machine	18
6.2.	Standard Message Processing Rules	21
6.2.1.	Message Parsing	21
6.2.2.	Authentication and Authorization	21
6.2.3.	Message Sequencing	21
6.2.4.	MNSLP Hop Counting	22
6.3.	Message Processing Rules	23
6.3.1.	CONFIGURE	23
6.3.2.	REFRESH	24
6.3.3.	OPTIONS	24
6.3.4.	RESPONSE	25
6.3.5.	NOTIFY	25
6.4.	Message Forwarding	26
6.5.	Interaction with GIST	26
7.	Examples of Operation	27
7.1.	Example with MNE_Selection is ANY	27
7.2.	Example with MNE_Selection is ALL	28
7.3.	Example with MNE_Selection is FIRST_and_LAST	29
7.4.	Example with a Route Changes	30
8.	Bit-Level Formats and Error Messages	30
8.1.	Metering NSLP Messages	30
8.2.	Metering NSLP Objects	31
8.2.1.	MSN	32
8.2.2.	Session_LT	32
8.2.3.	MNE_Selection	32
8.2.4.	INFO	33
8.2.5.	MSPEC Objects	33
9.	Mapping onto M-NSLP Requirements	34
10.	Security considerations	36
11.	Open Issues	36
12.	Acknowledgements	36

13.	References	36
13.1.	Normative References	36
13.2.	Informative References	37
	Authors' Addresses	38
	Intellectual Property and Copyright Statements	41

1. Introduction

Monitoring, metering and accounting of packets is an important functionality in many networks today. Several working groups have described mechanisms to collect and report usage data for resource consumption in a network by a particular entity. For example, the IPFIX WG defines a protocol for this purpose. The PSAMP WG defines a standard to sample subsets of packets by statistical and other methods. RADIUS (see [[RFC2865](#)] and [[RFC2866](#)]) and DIAMETER (see [[RFC3588](#)] and [[I-D.ietf-aaa-diameter-cc](#)]) are also protocols which provide information about consumed resources. The Meter MIB [[RFC2720](#)] is a MIB for collecting flow information.

However, it is also necessary to configure and coordinate the entities performing the metering. RADIUS, DIAMETER and SNMP are candidates for this configuration. Nevertheless, these protocols require some knowledge about the location of the appropriate Metering Entities to configure them.

In more complex network topologies and architectures, the appropriate entities to meter the properties of a given data flow can be discovered dynamically using path-coupled signaling. If more than one Metering Entity is required, all of them can potentially be configured and coordinated with a single message along the path.

Scenarios and requirements for Metering NSLP are described in [[I-D.fessi-nsis-m-nslp-framework](#)].

This draft defines the Metering NSLP for configuration and coordination of Metering Entities in a path-coupled fashion.

2. Terminology

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](#)].

Furthermore, the terminology defined by GIST [[I-D.ietf-nsis-ntlp](#)] applies to this draft.

Moreover, this document uses the following terms:

Metering Record

A Metering Record describes flow characteristics for a particular flow. Examples of such data are packet counter and time information.

Monitoring Probe

A Monitoring Probe is an entity that examines data flows in order to gather Metering Records.

Metering Entity

A Metering Entity is a node that is equipped with one or more Monitoring Probes. A Metering Entity MAY process the Metering Records, e.g. by aggregating them or by sampling them, before they are exported elsewhere, typically to a Collector. A Metering Entity MAY host other functions, for example, for processing of NSIS signaling.

Collector

A Collector receives Metering Records from one or multiple Metering Entities. These Metering Records can be aggregated and/or correlated by the Collector. The Collector is typically not co-located with a Metering Entity. Another protocol (e.g. IPFIX) is needed to transport the Metering Records from the Metering Entity to the Collector.

Metering Configuration State

A State used/kept by the Metering Manager to configure the Monitoring Probe.

Metering Manager

A unit co-located with the Monitoring Probe that communicates with M-NSLP processing. It holds Metering Configuration State which is used to configure the Monitoring Probe.

Metering NSIS Entity (MNE)

A Metering Entity which supports the Metering NSLP.

Metering NSIS Initiator (MNI)

The first node in the sequence of MNEs that issues a configuration message for a flow or aggregate.

Metering NSIS Responder (MNR)

The last node in the sequence of MNEs that receives a configuration message for a flow or aggregate.

Metering NSIS Forwarder (MNF)

A MNE that is neither MNI nor MNR.

3. Protocol Overview

The design for the Metering NSLP and the processing of M-NSLP messages is in some aspects similar to QoS NSLP [[I-D.ietf-nsis-qos-nslp](#)] and in other aspects to the NATFW NSLP [[I-D.ietf-nsis-nslp-natfw](#)]. One of the main differences compared to the QoS NSLP and the NATFW NSLP is that only a subset (in some cases even only one) of the Metering Entities in the signaling path will take part in the actual Metering. This fact has several consequences on the signaling itself and therefore on the protocol design.

3.1. Model of operation

Figure 1 shows an example logical model of the operation of the Metering NSLP and the associated metering mechanism in a MNE.

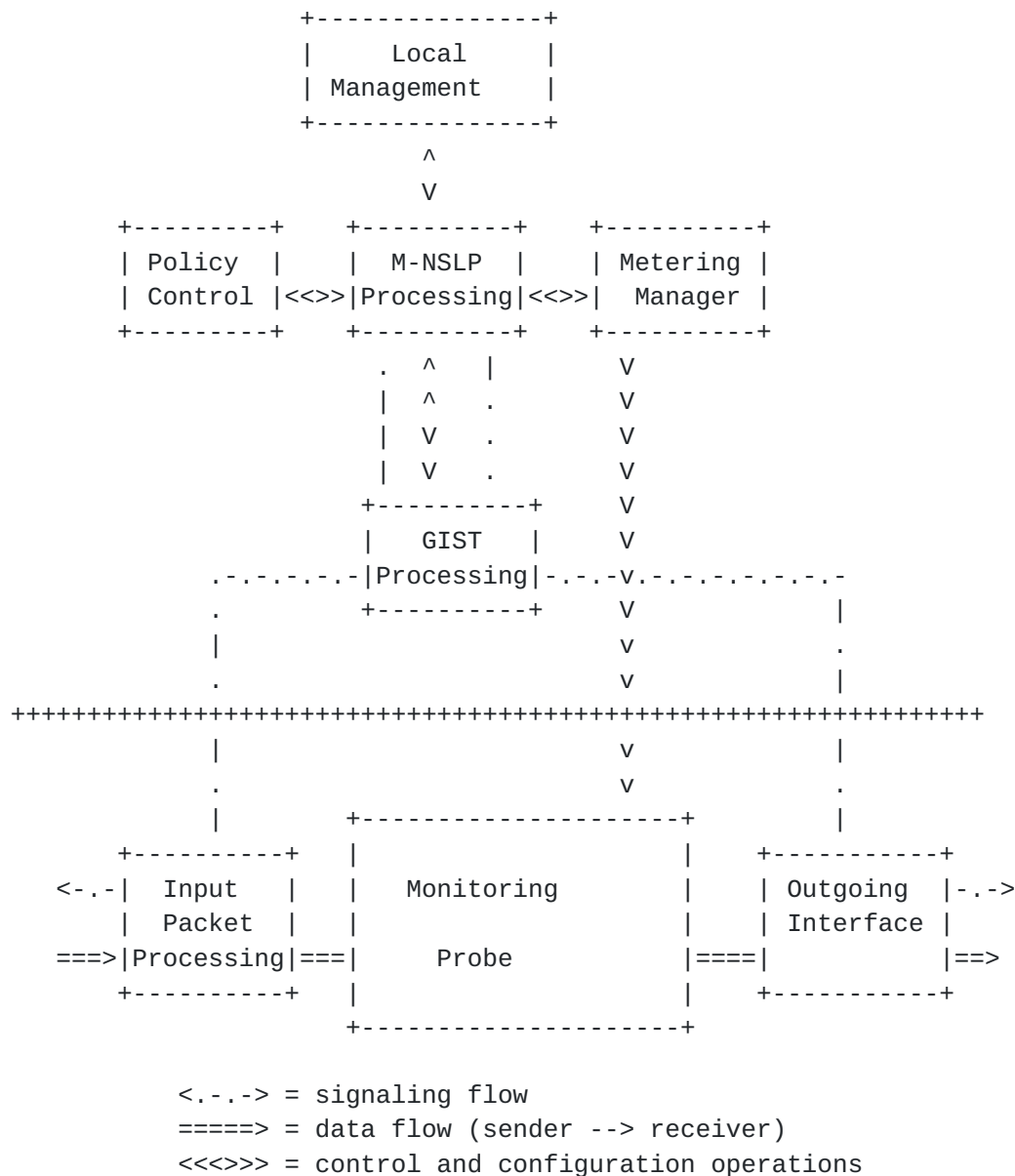


Figure 1: Metering-NSLP in Metering Entity

The Monitoring Probe collects metering data and processes them into Metering Records, which can be sent to the Collector. The Monitoring Probe is co-located with the Input Packet Processing and the Outgoing Interface.

A M-NSLP message called CONFIGURE transports:

- o Control information objects for the M-NSLP Processing, such as message sequence numbers and whether the MNE should actually take part in the metering process

- o Metering configuration information objects, also called MSPEC objects, that describe the actual configuration information. The MSPEC objects are interpreted by the Metering Manager and SHOULD be opaque to the M-NSLP Processing.
- o Policy information to authenticate and authorize the configuration request

The M-NSLP Processing decides if the current MNE is addressed by this configuring message. If yes, the MSPEC objects are extracted from the M-NSLP message and passed to the Metering Manager, where they are interpreted and used to install Metering Configuration State. The Metering Manager uses this state to configure the Monitoring Probe.

The Policy Control determines whether the sender of the M-NSLP message is authorized to configure the Monitoring Probe.

From the perspective of a single node, the request for configuration of Metering Entities may result from processing a local management request, or from processing an incoming M-NSLP message. The local management may in turn be triggered by a local application, e.g. when a video is requested from a video server, or by a physically separate network node.

3.2. Metering NSLP Messages

The Metering NSLP messages are classified into 3 categories:

- o request messages
- o response messages
- o asynchronous notifications

3.2.1. CONFIGURE

CONFIGURE is a M-NSLP request message. It is used to create Metering Configuration State in a Metering Entity.

3.2.2. REFRESH

REFRESH is a M-NSLP request message. It is used:

- o to extend the lifetime of an existing M-NSLP session.
- o for terminating a session (with session lifetime zero).
- o to detect route changes at the NSLP layer.

3.2.3. OPTIONS

OPTIONS is a M-NSLP request message. It is used to inquire the metering capabilities of MNEs along the signaling path.

3.2.4. RESPONSE

The RESPONSE message is used to provide information about the result of a previous M-NSLP request. This includes explicit confirmation of the state manipulation signaled in the CONFIGURE/REFRESH message or an error code.

3.2.5. NOTIFY

The NOTIFY message is an asynchronous notifications message used to convey information to a MNE. It differs from RESPONSE messages in that it is sent asynchronously and does not need to refer to any particular state or previously received message. The information conveyed by a NOTIFY message is typically related to error conditions. An example would be a notification to the MNI about state being torn down.

4. Design Decisions

4.1. Soft State

NSIS state is always soft state and needs to be refreshed. The control information in CONFIGURE/REFRESH messages carry an object that determines the lifetime of a M-NSLP session. The MNI suggests a lifetime for the session that it is being signaled for. Each MNE participating in the metering process MAY or MAY not accept the suggested lifetime or MAY start the metering with a reduced lifetime, depending on the local policies of the MNEs. This behavior is similar to the calculation of session lifetime in the NATFW-NSLP [[I-D.ietf-nsis-nslp-natfw](#)].

4.2. Message Sequencing

The order in which M-NSLP requests arrive influences the eventual configuration state that will be stored at a MNE. Therefore the M-NSLP supports the detection of message duplication and re-ordering using a Message Sequence Number (MSN) object. More Details on Message Sequencing are provided in [Section 6.2.3](#)

4.3. Message Scoping

Metering Entities at the edge of a signaling scope (for example, at the edge of the administrative domain of an ISP) MUST be marked accordingly. The Metering NSLP does not provide means by itself to discover dynamically the border of the signaling scope.

When a MNE recognize that it is at the edge of the signaling scope,

it MUST terminate the signaling and act as NSIS Responder (NR).

4.4. Selection of MNEs

An interesting feature of the Metering NSLP is that only a subset of MNEs on the data path might take part in the actual metering. Metering Entities taking part in the metering process are determined based, for example, on their type or position on the path.

When a CONFIGURE message is sent, each MNE on the data path needs to determine whether it will take part in the metering process. For this reason, the 'Selection of Metering Entities' object is included in CONFIGURE messages.

4.5. Coordination of Metering Tasks

The M-NSLP allocates different metering tasks to different Metering Entities on the signaling path depending on their position and metering capabilities. A 'Correlation ID' is included in the MSPEC and distributed to the Metering Entities during the configuration process. This 'Correlation ID' can be forwarded to the Collector, which can correlate Metering Records coming from different Metering Entities and belonging to the same session/measurement.

4.6. Aggregation

The metering configuration SHOULD allow aggregation of Metering Records belonging to the same user or application. The information required for the aggregation must be specified in the MSPEC. Such aggregation can be done in two ways.

- o A Monitoring Probe separately collects and reports data for each micro flow (e.g., given for all different combinations of port numbers) contained in the macro flow that is signaled by the NTLP.
- o A Monitoring Probe separately collects micro flows but reports all flows in a single record.

4.7. Reaction to Route Changes

M-NSLP automatically adapts to re-routing events. State along the old path times out automatically. When a new REFRESH message is initiated by the MNI and hits a MNE that is a new one on the signaling path, the MNE sends a negative RESPONSE with error code "Unknown Session" backwards to the MNI. The MNI MAY re-initiate the signaling along the new path.

Since metering tasks are allocated dynamically to MNEs on the path, when a MNE receives a CONFIGURE message for an existing session, old state for this session MUST be torn down and a new session is created

in order to avoid inconsistencies with an old configuration.

4.8. Metering Configuration Information

The actual configuration information for the Metering Entities is carried within the MSPEC objects. MSPEC objects are interpreted by the Metering Manager and SHOULD be opaque to M-NSLP Processing. Each MSPEC object contains a meter configuration. A single CONFIGURE message MAY contain multiple MSPEC objects describing different metering tasks, which MAY be allocated to different MNEs.

The Metering NSLP does not provide its own format for encoding the MSPEC objects. Instead, formats of existing protocols, such as IPFIX [[I-D.ietf-ipfix-protocol](#)], Diameter [[RFC3588](#)], or NetFlow [[RFC3954](#)], are re-used. The encoding of an MSPEC object depends on the metering and reporting technology. For example, the MSPEC object used for the configuration of an IPFIX device is encoded in IPFIX format. Technology-specific encoding avoids the problems of mapping the different semantics of information models for these technologies to each other.

4.9. Required State Information

For each M-NSLP session, the required state information at each MNE consists of:

NTLP state

NTLP state is required for each M-NSLP session at each MNE on the signaling path, even if the MNE is not taking part in the metering process, in order to avoid GIST re-discovering the MNE each time when it refreshes its routing state [[I-D.ietf-nsis-ntlp](#)].

M-NSLP state

M-NSLP state is required for each M-NSLP session at each MNE on the signaling path. Even if the MNE is not taking part in the metering process a minimal session state is required, which consists of the Session Identifier (SID) and the Session Lifetime (Session_LT).

Assume MNEs located on the signaling path, which are not taking part in the metering process, would not save M-NSLP state for this M-NSLP session. Then when a MNE receives a REFRESH message with an unknown Session ID, it would not be able to make the difference:

- * whether the MNE is new on the signaling path due to a route change

- * or whether the MNE has been already on the signaling path but is not taking part in the metering process.

Therefore, the MNE needs to keep M-NSLP state for this session in order to be aware of it. Implementations of the M-NSLP MAY include optimizations to reduce the amount of memory required if the session is known but the MNE is not taking part in the metering process.

Metering Configuration State (Optional)

Metering Configuration State is required for each M-NSLP only if the MNE is taking part in the metering process. It includes the MSPEC objects which describe the metering tasks that are allocated to the MNE.

5. Metering NSLP Messages and Objects

5.1. Metering NSLP Objects

5.1.1. The 'Session Identifier' Object (SID)

SID is globally statistically unique identifier for a MNSLP session. SID is not carried explicitly by MNSLP messages. It is rather carried by GIST.

SID MUST be generated for each new MNSLP session by the MNI and provided to GIST via the GIST API. All MNFs and the MNR receive the SID via the GIST API and MUST NOT modify it.

Note that this is conform to the specification of GIST [[I-D.ietf-nsis-ntlp](#)]: [Section 3.5](#). (Signaling Sessions): "Signaling applications provide the session identifier (SID) whenever they wish to send a message, and GIST reports the SID when a message is received."

5.1.2. The 'Message Sequence Number' Object (MSN)

MSN is used to detect duplicate and lost Metering NSLP messages. It is also used to to mach an incoming RESPONSE message to the appropriate request.

5.1.3. The 'Selection of Metering Entities' Object (MNE_Selection)

This object is required to determine which MNEs will actually take part in the metering. The value of MNE_Selection can be one of the following:

- o FIRST: the first MNE in the signaling path, i.e. the NI.
- o LAST: the last MNE signaling in the signaling path, i.e. the NR
- o 'FIRST_and_LAST': both the NI and the NR take part in the signaling.
- o ANY: any available MNE can perform the metering.
- o ALL: Each MNE capable of executing this metering request MUST perform it. "ALL" must be interpreted here as "as many as possible".
- o Enterprise-specific: Enterprises may wish to define their own methods to decide which Metering Entities should perform the metering. In this case, the parameter 'Selection of Metering Entities' needs to be combined with an enterprise-specific identifier. (See also <http://www.iana.org/assignments/enterprise-numbers>) If enterprises use their own defined methods for the MNE selection, some other M-NSLP objects not defined in this document may be required to take the decision which MNE will take part in the metering process.

5.1.4. The 'Session Lifetime' Object (Session_LT)

This object carries the requested lifetime for a M-NSLP session in a CONFIGURE / REFRESH message or the granted session lifetime in a RESPONSE message, in milliseconds. When a M-NSLP session expires, the Metering Manager MUST configure the Monitoring Probe to stop the Metering.

A value of zero for the 'Session Lifetime' object leads to immediate termination of the corresponding session.

5.1.5. The 'MNSLP Hop Count' Object (MNSLP_Hop_Count)

This object carries the number of previous MNEs on the signaling path for this session. This object is a integer which starts by zero at the MNI and MUST be incremented by one at each MNSLP hop on the signaling path.

5.1.6. The 'Information Code' Object (INFO)

This object carries the response code, which may be an indication for either a successful or failed request depending on the value of the 'response code' field (See [Section 8.2.4](#) for more details).

5.1.7. MSPEC Objects

The MSPEC objects describe the actual Configuration information. They are interpreted in the Metering Manager and SHOULD be opaque to M-NSLP Processing. MSPEC objects are described separately in [Section 5.3](#).

5.2. Metering NSLP Messages

This section defines which objects are carried in each Metering NSLP message. The description is provided in Augmented Backus-Naur Form (ABNF) specified in [[RFC4234](#)]. Message format at the bit-level is provided in [Section 8](#)

The ABNF implies an order for the objects in a message. However, in many (but not all) cases, object order makes no logical difference. An implementation should create messages with the objects in the order shown here, but accept the objects in any order, except for MSPEC object(s) which always appear last in the message, and whose mutual order matters.

All M-NSLP messages start with a common header 'HEADER' which is defined in [Section 8](#)

Note that the Session ID (SID) is not explicitly mentioned in this notation, since it is provided in GIST.

5.2.1. CONFIGURE

The format of a CONFIGURE message is as follows:

```
CONFIGURE = HEADER MSN Session_LT MNE_Selection [MNSLP_Hop_Count]
<1>*MSPEC
```

(According to [[RFC4234](#)] '<1>*MSPEC' means 'one or more MSPEC objects')

5.2.2. REFRESH

The format of a REFRESH message is as follows:

```
REFRESH = HEADER MSN Session_LT
```

5.2.3. OPTIONS

The format of a OPTIONS message is as follows:

```
OPTIONS = HEADER MSN
```


5.2.4. RESPONSE

The format of a RESPONSE message is as follows:

RESPONSE = HEADER MSN INFO Session_LT <0>*MSPEC

(According to [[RFC4234](#)] '<0>*MSPEC' means 'zero or more MSPEC objects')

The MSN MUST be the same as in the corresponding M-NSLP request in order to match a RESPONSE message to the appropriate request

Session_LT carries the granted session lifetime, which might be lower than the session lifetime requested by the MNI.

NO MSPEC object MUST be included if INFO indicates a successful confirmation of the request. The MSPEC objects are required only to report a failure in case the MSPEC objects list in the CONFIGURE message, or a sub-set of it, can not be processed.

5.2.5. NOTIFY

The format of a NOTIFY message is as follow:

NOTIFY = HEADER INFO

The INFO indicates the reason for the notification.

5.3. MSPEC Objects

As mentioned above, the MSPEC objects describe the actual configuration information. They are interpreted in the Metering Manager and SHOULD be opaque to M-NSLP Processing. Each MSPEC object contains a meter configuration. The MRI provides additional information to the MSPEC object on the flows/packets to be metered. The combination of the MRI and an MSPEC object contains a complete description of a requested metering task. A single CONFIGURE message may contain multiple MSPEC objects describing different metering tasks.

There is no common format or semantics for encoding metering tasks within an MSPEC object. The encoding of an MSPEC object depends on the requested metering and reporting technology, such as IPFIX, Diameter, or NetFlow.

There are different types of MSPEC objects depending on the metering and reporting technology. For example, there are IPFIX MSPEC objects, Diameter MSPEC objects, etc. The object type, which is

provided in the object header as described in [Section 8.2](#), implicits the requested metering and reporting technology.

Independently of the encoding, each MSPEC object MUST, in combination with the MRI, contain sufficient information for configuring an MNE to perform a metering task. Specifications of MSPEC objects MUST ensure that they provide sufficient means for configuring the MNE as described in sections [4.1.2](#) and [4.2.2](#) of [\[I-D.fessi-nsis-m-nslp-framework\]](#).

Below the MSPEC object for IPFIX metering tasks is specified. MSPEC objects for other meters may be defined in further standards documents. Proprietary MSPEC objects MAY be also used.

[5.3.1](#). IPFIX MSPEC Object

The IPFIX MSPEC object uses structure, semantics and encoding of IPFIX Messages as defined in [\[I-D.ietf-ipfix-protocol\]](#) and [\[I-D.ietf-ipfix-info\]](#). An IPFIX MSPEC object consists of a single IPFIX Message that contains three Sets, a Template Set, an Option Template Set, and a Data Set. Each set contains a single record only.

Fields of the header of the contained IPFIX Message are filled with values according to section 3.1 of [\[I-D.ietf-ipfix-protocol\]](#) except for fields "Export Time" and "Observation Domain ID" which must be both set to 0x00000000.

The Template Record in the Template Set defines the requested reporting. Except for the Template ID in the Template Record Header, the MNE MUST use exactly this Template Record for reporting metering results. The MNE also MUST use the Template ID received in the Template record for its metering reports. But if there is a collision caused by two different MSPEC objects (potentially from different NSIS sessions) that would violate the uniqueness of Template IDs as specified in section 3.4.1 of [\[I-D.ietf-ipfix-protocol\]](#), then the MNE MUST solve this conflict by modifying Template IDs. In order to keep the probability of such a collision low, the Template ID SHOULD be a randomly generated number.

The Option Template Record in the Option Template Set describes the format of the single Data Record in the Data Set. To the Template ID the same rules apply as for the Template Record. The Scope Field Count is always 0.

The Data Record in the Data Set contains the requested metering configuration. It is structured according to the Option Template Record in the Option Template Set. It MUST contain the following Information Elements that are specified in [\[I-D.ietf-ipfix-info\]](#).

flowKeyIndicator

This Information Element identifies which of the reported Flow properties are Flow Keys.

collectorIPv4Address or collectorIPv6Address, respectively

The IPv4 or IPv6 address, respectively, to which metering results are requested to be reported.

collectorProtocolVersion

The version of the IPFIX protocol to be used.

collectorTransportProtocol and collectorTransportPort

The transport protocol and transport port number to be used for reporting metering results

The Data record MAY contain further Information Elements for specifying the metering task, for example, the flowIdleTimeout.

If flow properties to be measured are specified in the Data record, they serve as filter. Only Flows that match the values of all these Information Element MUST be reported by the MNE. If multiple values are contained for the same property, i.e. it multiple Information Elements with the same Information Element identifier are contained, then it is sufficient if one of these values is matched by a Flow.

6. Processing Rules

6.1. Session State Machine

The hereby presented state machine of a M-NSLP session outlines the operation of the M-NSLP. It has three main states:

'closed':

At this state, the MNE does not keep any state for the session.

'pending':

At this state a CONFIGURE message has been received. The 'pending' state consists of 2 sub-states

pending.forward

The MNE is not taking part in the metering process. Instead, it is just expecting a confirmation to know that the session is active and some other MNEs will be metering.

pending.participating

The MNE is actively taking part in the metering process. The MNE has been configured. However, the configuration has not been activated yet, i.e. metering function has not been started.

'metering':

At this state the configuration request has been successfully confirmed with an appropriate RESPONSE message. The 'metering' state consists also of 2 sub-states.

metering.forward

The MNE is not actively taking part in the metering process. Instead, it is informed that the session is active and there are some other MNEs along the signaling path for this session that took over the metering process.

metering.participating

The MNE is actively taking part in the metering process. The MNE is metering according to a metering configuration received by a CONFIGURE message.

A session is uniquely identified by the SID. All sessions start in state 'closed'. Transitions between states are caused by events:

CONF:

This transition is triggered by a CONFIGURE message that is received and processed successfully by the MNE and that identifies a new session.

CONF-0:

This transition is triggered by a CONFIGURE message that is received and processed successfully by the MNE and that identifies an existing session in state 'pending' or 'metering'. This transition leads always to the state 'closed' in order to avoid

inconsistencies with old configurations. Furthermore, this transition MUST be immediately followed by a 'CONF' transition, i.e. after the old session is torn down, a new session is created.

REFR:

This transition is triggered by a REFRESH message that identifies an existing session in state 'pending' or 'metering'.

RESP-P:

This transition is triggered by a positive RESPONSE message that indicates that a metering configuration request or a refreshing request can be activated. This transaction always leads to state 'metering'.

RESP-N:

This transition is triggered by a negative RESPONSE message. This transaction always leads to state 'closed'.

T-OUT-1:

This transition is triggered by a timeout of a session in state 'pending'. The time interval for this timeout is typically large enough in order to tolerate network failures, network congestion and slow MNEs along the signaling path. This transition always leads to state 'closed'.

T-OUT-2:

This transition is triggered by a timeout of a session in state 'metering'. The time interval for this timeout is defined by the MNSLP session lifetime. This transition always leads to state 'closed'.

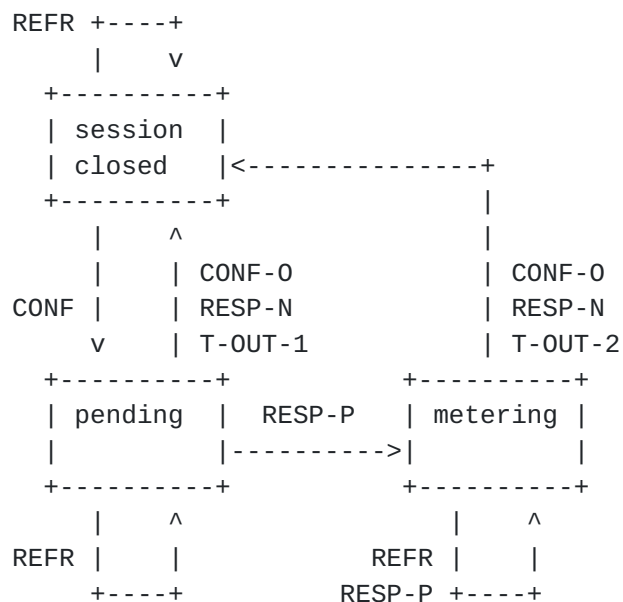


Figure 2: M-NSLP Session State Machine

6.2. Standard Message Processing Rules

6.2.1. Message Parsing

When a M-NSLP message is received, the MNE first checks the message format. In case of a malformed message or a mandatory object is missing, the MNE MUST NOT propagate the message any further. If the message has been a request the MNE MUST construct an RESPONSE message indicating the error condition and send it back to towards the MNI. If it has been a response, the message is discarded silently.

If a message contains an object of an unrecognized type, then the behavior depends on the object type value.

6.2.2. Authentication and Authorization

When a M-NSLP message is received, the MNE MUST determine whether the sender of the request is authenticated and authorized before any state changing actions are performed.

6.2.3. Message Sequencing

Message sequencing in the Metering NSLP is identical to message sequencing in the NATFW NSLP.

In more details:

M-NSLP messages need to carry an identifier so that all MNEs along the path can distinguish messages sent at different points in time. Messages can be lost along the path or duplicated. So all MNEs should be able to identify either old messages that have been received before (duplicated), or the case that messages have been lost before (loss). This requires every Metering message to carry a message sequence number (MSN), see also [Section 5.1.2](#).

The MSN MUST be set by the NI and MUST NOT be set or modified by any other MNE. The initial value for the MSN MUST be generated randomly and MUST be unique only within the session for which it is used. The NI MUST increment the MSN by one for every message sent. Once the MSN has reached the maximum value, the next value it takes is zero. All MNEs MUST use the algorithm defined in [[RFC1982](#)] to detect MSN wrap-arounds.

NSIS forwarders and the responder store the MSN from the initial CONFIGURE packet which creates the session as the start value for the session. NFs and NRs MUST include the received MSN value in the corresponding RESPONSE message that they generate.

When receiving a request message, a MNE uses the MSN given in the message to determine whether the state being requested is different to the state already installed. If the received MSN value is equal to or lower than the stored MSN value, this can indicate a duplicated or replayed message. In this case, the message MUST NOT have any impact the state of the MNE or the MNSLP session. However, the message MUST be forwarded towards the next MNSLP hop, since one or more of the MNEs on the rest of the signaling path may have not received this message yet.

If the received MSN value is greater than the already stored MSN value, the MNE MUST update its stored state accordingly, if permitted by all security checks, and stores the updated MSN value accordingly.

[6.2.4](#). MNSLP Hop Counting

Depending on the application scenario for the MNSLP, MNEs may need to know their position on the signaling path. In this case, a MNE needs to make the difference between:

- o the number of IP hops between the MNI and itself
- o the number of GIST hops between the MNI and itself
- o the number of MNSLP hops between the MNI and itself

The GIST-hop-count and IP-TTL can be provided by the GIST API as described in [[I-D.ietf-nsis-ntlp](#)] (Section B.1 and Section B.2).

The MNI MAY introduce an 'MNSLP Hop Count' object in the CONFIGURE

message. In cases it does, it MUST set the 'MNSLP Hop Count' to zero.

If a 'MNSLP Hop Count' object is included in a CONFIGURE message, MNFs and MNRs MUST increment the 'MNSLP Hop Count' by one before the message is forwarded to the next MNSLP hop.

6.3. Message Processing Rules

6.3.1. CONFIGURE

When a MNE receives a CONFIGURE message, and after successful message parsing and authentication/authorization are performed, the MNE performs a lookup in its M-NSLP session table. If the session already exists, the state for this session MUST be torn down and a new session MUST be created in order to avoid inconsistencies with earlier configurations due to the dynamic allocation of the MSPEC objects to the MNEs.

Now, regardless of whether the session is new or has been renewed, the MNE needs to figure out if it will take part in the metering process based on the MNE_Selection object, the MSPEC objects list and the capabilities of the MNE.

6.3.1.1. MNE_Selection is ALL

- o If the MNE is capable of metering the objects in the MSPEC objects list, then the MNE is taking part in the metering process. The MNE MUST change the current session state to 'pending.participating'.
- o If the MNE is NOT capable of metering the objects in the MSPEC objects list, then the MNE is NOT taking part in the metering process. The MNE MUST change the current session state to 'pending.forward'

6.3.1.2. MNE_Selection is FIRST_and_LAST

If MNE_Selection is FIRST_and_LAST and the MNE is the first or the last on the signaling path:

- o If the MNE is capable of metering the objects in the MSPEC objects list, then the MNE is taking part in the metering process. The MNE MUST change the current session state to 'pending.participating'.

- o If MSPEC objects can not be metered, i.e. the MNE is the first or the last on the signaling path but is not capable to measure the objects given in the MSPEC objects list, the MNE generates a negative RESPONSE message with the appropriate INFO object. The MNE changes the current session state to 'closed'.
- o EDITORIAL NOTE: The case FIRST_and_LAST is different from the cases ALL and ANY here.

If MNE_Selection is FIRST_and_LAST and the MNE is NOT the first NOR the last on the signaling path, the MNE MUST change the current session state to 'pending.forward'.

6.3.1.3. MNE_Selection is ANY

In case MNE_Selection is 'ANY', the MNE inspects the MSPEC objects list and removes those objects that it is capable to meter and adds them to its local MSPEC objects list for this session. This local MSPEC objects list represents the list of metering tasks allocated to this MNE for this session.

- o After inspecting the MSPEC objects list in the CONFIGURE message, if the local MSPEC objects list is empty, i.e. the MNE is not able to meter any of the given objects in the MSPEC objects list, the MNE MUST change the current session state to 'pending.forward'.
- o If the local MSPEC objects list is NOT empty, i.e. the MNE has been allocated some metering tasks from the MSPEC objects list, the MNE MUST change the current session state to 'pending.participating'. The local MSPEC objects is stored in the Metering Configuration State in order to use when the configuration will be activated.

6.3.2. REFRESH

When a MNE receives a REFRESH message, and after successful message parsing and authentication/authorization are performed, the MNE performs a lookup in its M-NSLP session table.

- o If the Session ID is not known, the MNE constructs a negative RESPONSE message with the appropriate error code and sends it upstream towards the MNI.
- o If the Session ID is found, the MNE remembers the requested lifetime in the REFRESH message and waits for a confirmation with a RESPONSE message. The current session state remains the same.

6.3.3. OPTIONS

tbd.

6.3.4. RESPONSE

When a MNE receives a RESPONSE message and after successful message parsing and authentication/authorization are performed, the remaining processing of the RESPONSE message depends on whether it is a negative RESPONSE or a positive RESPONSE. If it is a negative RESPONSE with the appropriate INFO object:

- o The MNE MUST change the current state for this session to 'closed'.
- o The MNE MAY keep state information for this session for optimization purposes, since the the MNI may re-initiate the signaling with different parameters soon.
- o The MNE MUST forward the message upstream towards the MNI as described in [Section 6.4](#).

When the MNI receives this message it MAY take the appropriate action based on the content of the INFO object.

If the RESPONSE message is a positive RESPONSE, the processing of the message depends on the type of the corresponding M-NSLP request:

RESPONSE to a CONFIGURE message

The pending configuration can be activated. If the session is currently in state 'pending.participating' the MNE MUST change the session state to 'metering.participating', and the Monitoring Probe MUST be configured to start the metering.

If the session is in state 'pending.forward' the MNE MUST change the session state to 'metering.forward'. No configuration action with the Monitoring Probe is taken.

RESPONSE to a REFRESH message

All existing timeouts for this session are reset. The session lifetime is extended with the lifetime given in the RESPONSE message. Note that this can be different from the session lifetime given in the previous REFRESH message.

In both cases, the MNE updates the session lifetime. If the session lifetime is zero, this implies that the session state is changed immediately to 'closed'.

6.3.5. NOTIFY

tbd.

6.4. Message Forwarding

CONFIGURE / REFRESH

After a M-NSLP request is successfully processed as described in [Section 6.3](#) (no error has been generated), then the session can be either in state 'pending' or 'metering'. Now, MNE needs to decide whether to forward the M-NSLP request towards the MNR or to construct a positive RESPONSE message. M-NSLP requests MUST be forwarded downstream except if:

- * the M-NSLP request has reached the MNR
- * the 'scoping' flag is set and the signaling scope has been reached.
- * all the involved MNEs in the metering process for this session have been reached.

The latter case can occur with a CONFIGURE request if MNE_Selection is 'ANY' and the MSPEC objects list has become empty, i.e. an appropriate MNE has been already found for each object in the MSPEC objects list and all the metering tasks have been allocated to some Metering Entities.

EDITORIAL NOTE: it has to verified whether this is the only case.

In this case, the MNE MUST act as a MNR for this session from now on. All the remaining requests MUST NOT be forwarded any further.

RESPONSE / NOTIFY

M-NSLP response messages and asynchronous notifications are forwarded upstream, i.e. the reverse direction of the data flow, until they reach the MNI.

6.5. Interaction with GIST

M-NSLP request messages (CONFIGURE or REFRESH) are initiated by the MNI and forwarded downstream, i.e. in the same direction as the data flow, towards the MNR, using the GIST API.

M-NSLP response messages can be initiated by the MNR or by any MNF on the signaling path and forwarded upstream, i.e. in the reverse direction as the data flow, towards the MNI, using the GIST API.

M-NSLP asynchronous notifications can be initiated by any MNE on the signaling path and forwarded upstream, i.e. in the reverse direction as the data flow, towards the MNI, using the GIST API.

All M-NSLP signaling is transported using GIST C-mode.

EDITORIAL NOTE: it should be verified whether D-mode might be also an option for initial configuration messages. However, authentication/authorization issues need to be considered.

7. Examples of Operation

7.1. Example with MNE_Selection is ANY

The MNI constructs a CONFIGURE message with the required MSPEC objects, and sends it towards the MNR. Assume 2 MSPEC objects are included, for example, MSPEC1 for 'counting bytes', and MSPEC2 for 'reporting the usage of the wireless link'. Assume further that MNI itself can perform MSPEC1 while and MNF2 can perform MSPEC2, for example, because it is an access router managing a range of WLAN networks. MNR can be, for example, the WLAN access point.

The CONFIGURE message is interpreted by the MNEs along the data path.

MNI removes MSPEC1 from the MSPEC objects list, changes its session state from 'closed' to 'pending.participating' and forwards the CONFIGURE message to MNF1.

MNF1 receives the CONFIGURE message with only MSPEC2 in the MSPEC objects list. However, MNF1 does not have the capabilities to meter and report the usage of the wireless link. MNF1 changes its session state from 'closed' to 'pending.forward' and forwards the CONFIGURE message to MNF2.

MNF2 interprets the CONFIGURE, removes MSPEC2 from the MSPEC objects list. Furthermore, it notices that the MSPEC object lists is now empty. Therefore, MNF2 replies with a positive RESPONSE message and becomes Responder for this session. Subsequent signaling for this session is stopped at MNF2 as shown in Figure 4. MNF2 starts the metering, changes the session state to 'metering.participating', constructs a positive RESPONSE message and sends it to MN1.

MNF1 receives the RESPONSE message and changes the session state to 'metering.forward'.

MNI receives the RESPONSE message, starts the metering and changes the session state to 'metering.participating'.

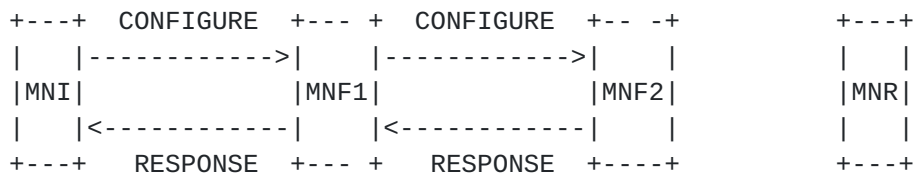


Figure 3: CONFIGURE-RESPONSE Message Flow with MNE_Selection 'ANY'

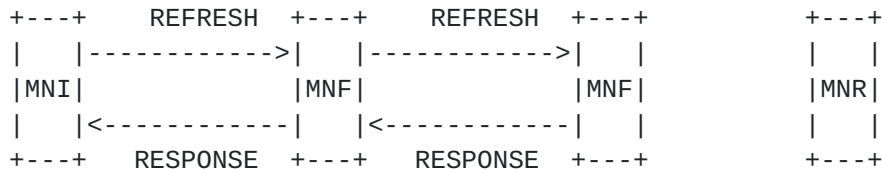


Figure 4: REFRESH-RESPONSE Message Flow with MNE_Selection 'ANY'

7.2. Example with MNE_Selection is ALL

The MNI constructs a CONFIGURE message with the required MSPEC objects, and sends it towards the MNR. The message is interpreted by each MNEs on the data path. Each MNE changes its state for this session to 'pending.participating' or 'pending.forward' depending on whether they are taking part of the metering process.

The MNR replies with a positive RESPONSE message. The RESPONSE message is interpreted by each MNE. Each MNE changes the session state to 'metering.participating' or 'metering.forward' depending on whether they are taking part of the metering process.

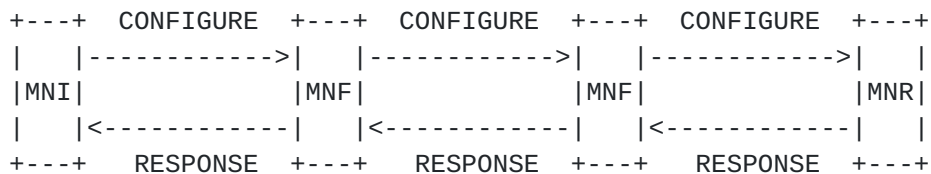


Figure 5: CONFIGURE-RESPONSE Message Flow with MNE_Selection 'ALL'

Similarly, a REFRESH message is initiated by MNI, travels towards the MNR where a RESPONSE is issued and sent back.

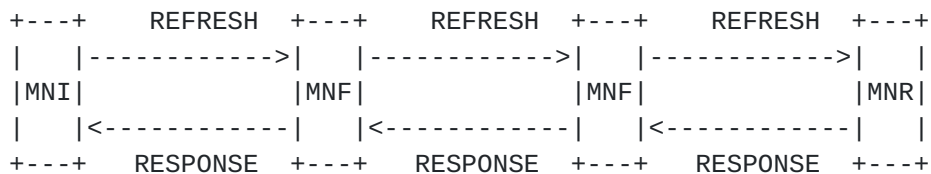


Figure 6: REFRESH-RESPONSE Message Flow with MNE_Selection 'ALL'

7.3. Example with MNE_Selection is FIRST_and_LAST

The MNI constructs a CONFIGURE message with the required MSPEC objects, and sends it towards the MNR. The message is interpreted by each MNEs on the data path. Each MNE changes its state for this session to 'pending.forward' except MNI itself and MNR, which perform a transition to the state 'pending.participating'.

The MNR replies with a positive RESPONSE message. The RESPONSE message is interpreted by each MNE. Each MNE changes the session state to 'metering.forward' except MNR and MNI, which perform a transition to the state 'metering.participating'.

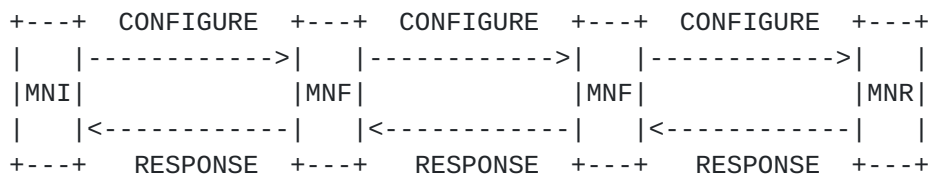


Figure 7: CONFIGURE-RESPONSE Message Flow with MNE_Selection 'FIRST_and_LAST'

Similarly, a REFRESH message is initiated by MNI, travels towards the MNR where a RESPONSE is issued and sent back.

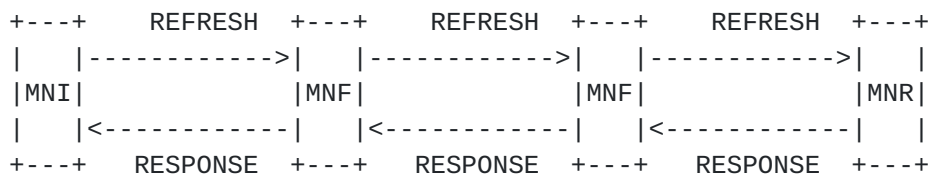


Figure 8: REFRESH-RESPONSE Message Flow with MNE_Selection 'FIRST_and_LAST'

7.4. Example with a Route Changes

Assume a topology as in Figure 9:

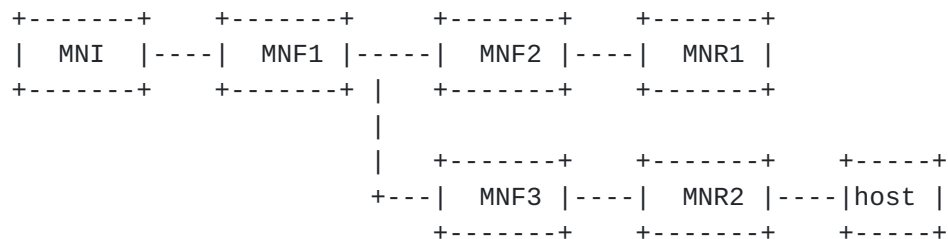


Figure 9: Example with a Route Change'

After a route change MNF2 and MNR1 are not on the signaling path anymore. MNF3 is new on the signaling path and receives a REFRESH message with an unknown Session ID. It sends a negative RESPONSE with error code "Unknown Session".

MNI receives the negative RESPONSE and re-initiates the signaling with a CONFIGURE message with the same Session ID.

When the MNE receives a CONFIGURE message with the same Session ID it re-starts the configuration process: the session is considered as a new session.

8. Bit-Level Formats and Error Messages

8.1. Metering NSLP Messages

All Metering NSLP messages start with a common header 'HEADER'. The Metering NSLP common header is similar to the QoS NSLP header. Note that it is not the same as the GIST Common Header.

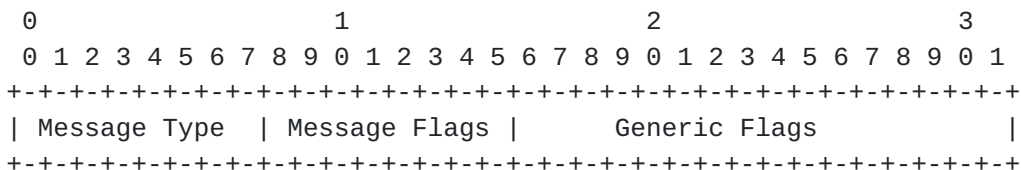


Figure 10: M-NSLP Message Common Header

The fields in the common header are as follows:

Message Type: 8 bits

- 1 = CONFIGURE
- 2 = RESPONSE
- 3 = NOTIFY

Message-specific flags: 8 bits

These flags are defined as part of the specification of individual messages.

Generic flags: 16 bits

There exists currently one generic flag, the Scoping bit (S). The use of the S-bit is described with each message that makes use of it.

The set of appropriate flags depends on the particular message being processed. Any bit not defined as a flag for a particular message MUST be set to zero on sending and MUST be ignored on receiving.

8.2. Metering NSLP Objects

The Metering NSLP uses the Type-Length-Value (TLV) object format defined by GIST [[I-D.ietf-nsis-ntlp](#)]. Every object consists of one or more 32-bit words with a one-word header. For convenience the standard object header is shown here:

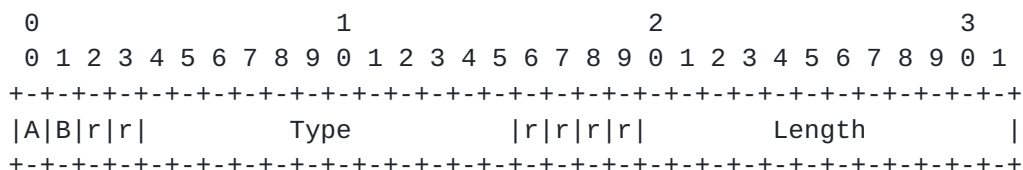


Figure 11: M-NSLP Object Common Header

The value for the Type field comes from Metering NSLP object type space, the various objects are presented in subsequent sections. The Length field is given in units of 32 bit words and measures the length of the Value component of the TLV object (i.e., it does not include the standard header).

The bits marked 'A' and 'B' are extensibility flags, and used to signal the desired treatment for objects whose treatment has not been defined in the protocol specification (i.e., whose Type field is unknown at the receiver).

EDITORIAL NOTE: include text here for the explanation of the meaning

of the A and B bits.

8.2.1. MSN

Type: MNSLP_MSN

Length: 1

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Message Sequence Number                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 12: Message Sequence Number Object

8.2.2. Session_LT

Type: MNSLP_Session_LT

Length: 1

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Session Lifetime                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 13: Session Lifetime Object

8.2.3. MNE_Selection

Type: MNSLP_MNE_Selection

Length: 1

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Selection of Metering Entities                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 14: Selection of Metering Entities Object

Possible values are:

- 1 = ALL
- 2 = ANY
- 3 = FIRST
- 4 = LAST

5 = FIRST_and_LAST
 greater than 1024 : Enterprise Specific

EDITORIAL NOTE: it still needs to be defined in the case Enterprise Specific, how MNSLP_SelectionMNEs is encoded.

8.2.4. INFO

This object carries the response code, which may be indications for either a successful request or failed request depending on the value of the 'response code' field.

Note that the format of this object is similar the INFO object for the NATFW NSLP.

Type: MNSLP_INFO

Length: 1

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Resv. | Class | Response Code |           Object Type           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 15: Information Code Object

The field 'Resv.' is reserved for future extensions and MUST be set to zero when generating such an object and MUST be ignored when receiving. The 'Object Type' field contains the type of the object causing the error. The value of 'Object Type' is set to 0, if no object is concerned. The 4 bit class field contains the severity class. The following classes are defined:

- o 0x1: Informational (NOTIFY only)
- o 0x2: Success
- o 0x3: Protocol error
- o 0x4: Transient failure
- o 0x5: Permanent failure
- o 0x6: Signaling session failures

EDITORIAL NOTE: Response Codes need to be defined here.

8.2.5. MSPEC Objects

Type:

Different types are possible.

- * MNSLP_IPFIX_MSPEC
- * MNSLP_DIAMETER_MSPEC
- * MNSLP_NETFLOW9

Note that this list is extensible.

Length: Variable.

9. Mapping onto M-NSLP Requirements

EDITORIAL NOTE: This section needs to be updated.

With the design described above, the requirements from [\[I-D.fessi-nsis-m-nslp-framework\]](#) are at this point satisfied as follows:

Extensibility:

The actual configuration information is encapsulated in the MSPEC. The MSPEC is designed such as it is interpreted by the Metering Manager and can be opaque to the M-NSLP processing. Furthermore, the MSPEC can be easily extended. Therefore, from the point of view of the configuration information, this requirement is fulfilled. Note however, that the Metering NSLP in its current design might show some lack of extensibility. For example, considering the selection of the MNEs, it might be useful for future application of the Metering NSLP to have the option "ANY N", where N is greater than one.

Interoperability:

Again, whether different metering solutions can interwork depends on how the MSPEC is designed. In QoS NSLP, the QSpec template design [\[I-D.ietf-nsis-qspec\]](#) aims at similar extensibility and interoperability. It needs to be studied whether or not the solutions chosen by the QSpec can also be applied to the MSPEC.

Flexible metering models:

As above, this is an issue of MSPEC design.

Distinguishing flows

The aggregation feature detailed in this requirement can be realized as described in [Section 4.6](#).

Flexible data collection:

Another issue that can be fixed in the MSPEC.

Location of Metering Entities:

MNEs, including MNI and MNR can be located anywhere on the data path.

Access parameters of the Collector Information:

Access parameters of the Collector Information on how to deliver flow records to the Collector is coded in the MSPEC.

Configuration of Metering Entities:

The protocol can configure Metering Entities that are MNEs, or that are controlled by MNEs.

Selection of Metering Entities:

As described in [Section 4.4](#), a MNE should be able to decide to pull out of the metering process. This is realized by the 'Selection of Metering Entities' object.

Metering Configuration State installation and removal:

By means of the CONFIGURE message, the protocol can install and remove Metering Configuration State.

Initiation and maintenance of metering tasks:

Triggers and correlation identifier are transported in the MSPEC.

Reaction to Route Changes:

The protocol detects route changes by a REFRESH or a refreshing CONFIGURE message and installs state along the new route, as described in [Section 4.7](#).

Scoping of configuration:

The M-NSLP needs to provide sufficient means for flexible scoping signaling messages.

Requirements not mentioned in this list are not yet addressed.

10. Security considerations

The process of configuring entities to start and stop metering and to transmit collected resource records to a third party introduces security challenges.

An extensive analysis of security issues related to the Metering NSLP is presented in Section 7 of [[I-D.fessi-nsis-m-nslp-framework](#)]. Based on this section, future versions of the M-NSLP protocol specification will elaborate the required security mechanisms for the M-NSLP.

11. Open Issues

Open issues for the Metering NSLP are discussed here:
<https://www.ri.uni-tuebingen.de/cgi-bin/roundup.cgi/mnslp/>

12. Acknowledgements

The authors would like to thank Robert Hancock, Martin Stiernerling and Andreas Klenk for their valuable input.

Furthermore, the authors would like to thank Angie So. By providing the first running prototype of the Metering NSLP, she gave us a helpful feedback for the protocol specification.

13. References

13.1. Normative References

- [I-D.ietf-nsis-ntlp]
Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", [draft-ietf-nsis-ntlp-12](#) (work in progress), March 2007.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.
- [RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
- [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", [RFC 1982](#),

August 1996.

13.2. Informative References

- [RFC2720] Brownlee, N., "Traffic Flow Measurement: Meter MIB", [RFC 2720](#), October 1999.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", [RFC 2865](#), June 2000.
- [RFC2866] Rigney, C., "RADIUS Accounting", [RFC 2866](#), June 2000.
- [RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", [RFC 3588](#), September 2003.
- [RFC3954] Claise, B., "Cisco Systems NetFlow Services Export Version 9", [RFC 3954](#), October 2004.
- [RFC4080] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", [RFC 4080](#), June 2005.
- [I-D.ietf-ipfix-protocol]
Claise, B., "Specification of the IPFIX Protocol for the Exchange", [draft-ietf-ipfix-protocol-24](#) (work in progress), November 2006.
- [I-D.ietf-ipfix-info]
Quittek, J., "Information Model for IP Flow Information Export", [draft-ietf-ipfix-info-15](#) (work in progress), February 2007.
- [I-D.ietf-psamp-sample-tech]
Zseby, T., "Sampling and Filtering Techniques for IP Packet Selection", [draft-ietf-psamp-sample-tech-07](#) (work in progress), July 2005.
- [I-D.ietf-psamp-mib]
Dietz, T. and B. Claise, "Definitions of Managed Objects for Packet Sampling", [draft-ietf-psamp-mib-06](#) (work in progress), June 2006.
- [I-D.ietf-psamp-info]
Dietz, T., "Information Model for Packet Sampling Exports", [draft-ietf-psamp-info-05](#) (work in progress), October 2006.

- [I-D.dressler-ipfix-aggregation]
Dressler, F., "IPFIX Aggregation",
[draft-dressler-ipfix-aggregation-03](#) (work in progress),
June 2006.
- [RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander,
"Requirements for IP Flow Information Export (IPFIX)",
[RFC 3917](#), October 2004.
- [I-D.ietf-nsis-qos-nslp]
Manner, J., "NSLP for Quality-of-Service Signaling",
[draft-ietf-nsis-qos-nslp-12](#) (work in progress),
October 2006.
- [I-D.ietf-nsis-fw]
Hancock, R., "Next Steps in Signaling: Framework",
[draft-ietf-nsis-fw-07](#) (work in progress), December 2004.
- [I-D.ietf-nsis-nslp-natfw]
Stiemerling, M., "NAT/Firewall NSIS Signaling Layer
Protocol (NSLP)", [draft-ietf-nsis-nslp-natfw-13](#) (work in
progress), October 2006.
- [I-D.ietf-nsis-qspec]
Ash, J., "QoS NSLP QSPEC Template",
[draft-ietf-nsis-qspec-15](#) (work in progress),
February 2007.
- [I-D.fessi-nsis-m-nslp-framework]
Fessi, A., "Framework for Metering NSLP",
[draft-fessi-nsis-m-nslp-framework-03](#) (work in progress),
June 2006.
- [I-D.ietf-aaa-diameter-cc]
Mattila, L., Koskinen, J., Stura, M., Loughney, J., and H.
Hakala, "Diameter Credit-control Application",
[draft-ietf-aaa-diameter-cc-06](#) (work in progress),
August 2004.
- [3GPP.32.240]
3GPP, "Telecommunication management; Charging management;
Charging architecture and principles", 3GPP TS 32.240
6.0.0, September 2004.

Authors' Addresses

Ali Fessi
University of Tuebingen
Wilhelm-Schickard-Institute for Computer Science
Sand 13
Tuebingen 72072
Germany

Phone: +49 7071 29-70576
Email: fessi@informatik.uni-tuebingen.de
URI: <http://net.informatik.uni-tuebingen.de/>

Georg Carle
University of Tuebingen
Wilhelm-Schickard-Institute for Computer Science
Sand 13
Tuebingen 72072
Germany

Phone: +49 7071 29-70505
Email: carle@informatik.uni-tuebingen.de
URI: <http://net.informatik.uni-tuebingen.de/>

Falko Dressler
University of Erlangen
Department of Computer Science 7
Martensstr. 3
Erlangen 91058
Germany

Phone: +49 9131 85-27914
Email: dressler@informatik.uni-erlangen.de
URI: <http://www7.informatik.uni-erlangen.de/>

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/>

Cornelia Kappler
Siemens AG
Siemensdamm 62
Berlin 13627
Germany

Phone: +49 30 386-32894
Email: cornelia.kappler@siemens.com

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

