Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 15, 2009

R. Stewart
The Resource Group
Q. Xie

M. Stillman
Nokia
M. Tuexen
Muenster Univ. of Applied Sciences

July 14, 2008

Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) Parameters
draft-ietf-rserpool-common-param-18.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 January 15, 2009.

Abstract

This document details the parameters of the Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) protocols defined within the Reliable Server Pooling (RSerPool) architecture.

Table of Contents

1. Introduction 4
  1.1. Conventions 4
2. Parameters in General 5
3. ENRP-ASAP Common Parameters 6
  3.1. IPv4 Address Parameter 8
  3.2. IPv6 Address Parameter 8
  3.3. DCCP Transport Parameter 9
  3.4. SCTP Transport Parameter 9
  3.5. TCP Transport Parameter 10
  3.6. UDP Transport Parameter 11
  3.7. UDP-Lite Transport Parameter 12
  3.8. Pool Member Selection Policy Parameter 13
  3.9. Pool Handle Parameter 13
  3.10. Pool Element Parameter 14
  3.11. Server Information Parameter 15
  3.12. Operation Error Parameter 16
    3.12.1. Unspecified Error 17
    3.12.2. Unrecognized Parameter Error 18
    3.12.3. Unrecognized Message Error 18
    3.12.4. Invalid Values Error 18
    3.12.5. Non-unique PE Identifier Error 18
    3.12.6. Inconsistent Pool Policy Error 18
    3.12.7. Lack of Resources Error 18
    3.12.8. Inconsistent Transport Type Error 18
    3.12.9. Inconsistent Data/Control Configuration Error 19
    3.12.10. Rejected due to security considerations 19
    3.12.11. Unknown Pool Handle Error 19
  3.13. Cookie Parameter 19
  3.14. PE Identifier Parameter 19
  3.15. PE Checksum Parameter 20
  3.16. Opaque Transport Parameter 20
4. Common Message Formats 22
5. IANA Considerations 24
  5.1. A New Table for RSerPool Parameter Types 24
  5.2. A New Table for RSerPool Error Causes 26
6. Security Considerations 27
7. Normative References 28
Authors' Addresses 29

Intellectual Property and Copyright Statements 30

1. Introduction

Aggregate Server Access Protocol (ASAP) [I-D.ietf-rserpool-asap] in conjunction with the Endpoint Handlespace Redundancy Protocol (ENRP) [I-D.ietf-rserpool-enrp] provides a high availability data transfer mechanism over IP networks.

Both protocols work together and so share many common parameters used in message formats. This document details the common message parameters shared between the two protocols. This document provides parameter formats only, for procedures and message composition please refer to the respective [I-D.ietf-rserpool-asap] and [I-D.ietf-rserpool-enrp] documents.

1.1. Conventions

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].

2. Parameters in General

All parameters described below MUST be in Network Byte Order (a.k.a. Big Endian, i.e., the most significant byte first) during transmission.

Please note that messages in both ENRP and ASAP are often composed of multiple parameters. These parameters may also be nested. In such a case a nested parameter will include the length of the padding between the nested parameters but not the last padding.

3. ENRP-ASAP Common Parameters

Parameters are defined in the following Type-Length-Value (TLV) format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Parameter Type       |       Parameter Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                       Parameter Value                         :
:                               +-------------------------------:
:                               |        Padding                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Parameter Type: 16 bits (unsigned integer)

The Type field is a 16 bit identifier of the type of parameter. It takes a value of 0 to 65534.
The value of 65535 is reserved for IETF-defined extensions.
Values other than those defined in specific ENRP parameter description are reserved by IETF. (Additional types, when needed, will be defined in the future through appropriate IETF/IANA procedures.)
The Parameter Types are encoded such that the highest-order two bits specify the action that must be taken if the processing endpoint does not recognize the Parameter Type.

00 Stop processing this ENRP or ASAP message and discard it, do not process any further parameters within it.

01 Stop processing this ENRP or ASAP message and discard it, do not process any further parameters within it, and report the unrecognized parameter in an 'Unrecognized Parameter' error (see Section 3.12).

10 Skip this parameter and continue processing.

11 Skip this parameter and continue processing, but report the unrecognized parameter in an 'Unrecognized Parameter' error (see Section 3.12).

The values of parameter types are defined as follows:

Value	Parameter Type
0x0	(reserved by IETF)
0x1	IPv4 Address
0x2	IPv6 Address
0x3	DCCP Transport
0x4	SCTP Transport
0x5	TCP Transport
0x6	UDP Transport
0x7	UDP-Lite
0x8	Pool Member Selection Policy
0x9	Pool Handle
0xa	Pool Element
0xb	Server Information
0xc	Operation Error
0xd	Cookie
0xe	PE Identifier
0xf	PE Checksum
0x10	Opaque Transport
others	(reserved by IETF)
0xffffffff	IETF-defined extensions

Table 1

Parameter Length: 16 bits (unsigned integer)
The Parameter Length field contains the size of the parameter in bytes, including the Parameter Type, Parameter Length, and Parameter Value fields. Thus, a parameter with a zero-length Parameter Value field would have a Length field of 4.
The total length of a parameter (including Type, Parameter Length and Value fields) MUST be a multiple of 4 bytes. If the length of the parameter is not a multiple of 4 bytes, the sender MUST pad the parameter at the end (i.e., after the Parameter Value field) with all zero bytes. The length of this padding is not included in the Parameter Length field. A sender MUST NOT pad with more than 3 bytes. The receiver MUST ignore the padding bytes.

Parameter Value: variable-length.
The Parameter Value field contains the actual information to be transferred in the parameter.

Parameter Padding: variable-length.
The Parameter Padding as described above.

3.1. IPv4 Address Parameter

This parameter defines a TLV that carries an IPv4 address.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x1             |      Length = 0x8             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 Address                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Address: 32 bits (unsigned integer)
Contains an IPv4 address. It is binary encoded.

3.2. IPv6 Address Parameter

This parameter defines a TLV that carries an IPv6 address.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 0x2         |          Length = 0x14        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         IPv6 Address                          |
|                                                               |
|                                                               |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 Address: 128 bit (unsigned integer)
Contains an IPv6 address. It is binary encoded.

3.3. DCCP Transport Parameter

This parameter defines a TLV that describes a user transport using DCCP protocol.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x3             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         DCCP port             |          (reserved)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      DCCP service code                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                     IPv4 or IPv6 Address                      :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, DCCP port, reserved fields, and IP address parameter.

DCCP port: 16 bits (unsigned integer)
The DCCP port number signed to this DCCP user transport.

DCCP service code: 32 bits (unsigned integer)
The DCCP service code signed to this DCCP user transport.

IPv4 or IPv6 Address
Indicates an IPv4 or IPv6 address parameter (as defined above in Section 3.1 and Section 3.2) assigned to this DCCP user transport. Unlike in an SCTP transport parameter, only one IP address parameter can be present in a DCCP transport parameter.

Note: A DCCP port MUST NOT be used for control information. For this reason, no Transport Use field is provided. DCCP MUST always be treated as a "Data Only" type transport use.

3.4. SCTP Transport Parameter

This parameter defines a TLV that describes a user transport using SCTP protocol.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x4             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        SCTP port              |          Transport Use        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                     IPv4 or IPv6 Address #1                   :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                              ...                              :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                     IPv4 or IPv6 Address #n                   :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, SCTP port, reserved fields, and all IP address parameters present.

SCTP port: 16 bits (unsigned integer)
The SCTP port number signed to this SCTP user transport.

Transport use: 16 bits (unsigned integer)
This field represents how the pool element intends this transport address to be used. The field MUST be populated with one of the following values:

Type	Value
DATA ONLY	0x0000
DATA plus CONTROL	0x0001

IPv4 or IPv6 Address #1 - #n
Each indicates an IPv4 or IPv6 address parameter (as defined above in Section 3.1 and Section 3.2) assigned to this SCTP user transport. An SCTP Transport parameter may have a mixed list of IPv4 and IPv6 addresses and at least one IP address parameter MUST be present in an SCTP transport parameter.

3.5. TCP Transport Parameter

This parameter defines a TLV that describes a user transport using TCP protocol.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x5             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         TCP port              |           (reserved)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                     IPv4 or IPv6 Address                      :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, TCP port, reserved fields, and IP address parameter.

TCP port: 16 bits (unsigned integer)
The TCP port number signed to this TCP user transport.

IPv4 or IPv6 Address
Indicates an IPv4 or IPv6 address parameter (as defined above in Section 3.1 and Section 3.2) assigned to this TCP user transport. Unlike in an SCTP transport parameter, only one IP address parameter can be present in a TCP transport parameter.

Note: A TCP port MUST NOT be used for control information. For this reason, no Transport Use field is provided. TCP MUST always be treated as a "Data Only" type transport use.

3.6. UDP Transport Parameter

This parameter defines a TLV that describes a user transport using UDP protocol.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x6             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         UDP port              |          (reserved)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                     IPv4 or IPv6 Address                      :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, UDP port, reserved fields, and IP address parameter.

UDP port: 16 bits (unsigned integer)
The UDP port number signed to this UDP user transport.

IPv4 or IPv6 Address
Indicates an IPv4 or IPv6 address parameter (as defined above in Section 3.1 and Section 3.2) assigned to this UDP user transport. Unlike in an SCTP transport parameter, only one IP address parameter can be present in a UDP transport parameter.

Note: A UDP port MUST NOT be used for control information. For this reason, no Transport Use field is provided. UDP MUST always be treated as a "Data Only" type transport use.

3.7. UDP-Lite Transport Parameter

This parameter defines a TLV that describes a user transport using UDP-Lite protocol.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x7             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      UDP-Lite port            |          (reserved)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      IPv4 or IPv6 Address                     :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, UDP-Lite port, reserved fields, and IP address parameter.

UDP port: 16 bits (unsigned integer)
The UDP-Lite port number signed to this UDP-Lite user transport.

IPv4 or IPv6 Address
Indicates an IPv4 or IPv6 address parameter (as defined above in Section 3.1 and Section 3.2) assigned to this UDP-Lite user transport. Unlike in an SCTP transport parameter, only one IP address parameter can be present in a UDP-Lite transport parameter.

Note: A UDP-Lite port MUST NOT be used for control information. For this reason, no Transport Use field is provided. UDP-Lite MUST always be treated as a "Data Only" type transport use.

3.8. Pool Member Selection Policy Parameter

This parameter defines a pool member selection policy. RSerPool supports multiple pool member selection policies and also allows definition of new selection policies in the future.

The enforcement rules and handling procedures of all the policies are defined in [I-D.ietf-rserpool-asap].

All pool member selection policies, both present and future, MUST use the following general parameter format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = 0x8             |      Length = variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Policy Type                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Policy-specific Data                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, Policy Type, and the Policy-specific Data fields.
Note, the Length field value will NOT include any padding at the end of the parameter.

Policy Type: 32 bits (unsigned integer)
Specifies the type of selection policy. The values are defined in [I-D.ietf-rserpool-policies].

Policy-specific Data:
The structure and fields for each presently defined policy type are described in detail in [I-D.ietf-rserpool-policies].

3.9. Pool Handle Parameter

This parameter holds a pool handle.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type = 0x9            |       Length=variable         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                          Pool Handle                          :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, and Pool Handle string.
Note, the value in Length field will NOT cover any padding at the end of the parameter.

Pool Handle
defined as a sequence of (Length - 4) bytes.

3.10. Pool Element Parameter

This parameter is used in multiple ENRP messages to represent an ASAP endpoint (i.e., a PE in a pool) and the associated information, such as its transport address, selection policy, and other operational or status information of the PE.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type = 0xa            |       Length=variable         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PE Identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Home ENRP Server Identifier                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Registration Life                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       User Transport param                    :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                  Member Selection Policy param                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       ASAP Transport param                    :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of octets, including the Type, Length, PE Identifier, Registration Life, User Transport, and Member Selection Policy parameters.
Note, the value in Length field will NOT cover any padding at the end of this Pool Element parameter.

PE Identifier: 32 bits (unsigned integer)
Uniquely identifies the PE in the pool. The PE picks its identifier when it starts up.

Home ENRP Server Identifier: 32 bits (unsigned integer)
Indicates the current home ENRP server of this PE. Set to all 0's if the PE's home ENRP server is undetermined.

Registration Life: 32 bits (signed integer)
Indicates the life time of the registration in number of seconds. A value of -1 indicates infinite life time.

User Transport
This can be either an DCCP, SCTP, TCP, UDP, UDP-Lite, or Opaque transport parameter (see Section 3.3, Section 3.4, Section 3.5, Section 3.6, Section 3.7, Section 3.16). A PE MUST have one and only one User Transport.

Member Selection Policy
Contains one of the defined member selection policy parameters (see Section 3.8).

ASAP Transport
This indicates the ASAP transport address of the PE and MUST be an SCTP type transport parameter (see Section 3.4).

3.11. Server Information Parameter

This parameter is used in ENRP to pass basic information of an ENRP server.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type = 0xb            |       Length=variable         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Server ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                        Server Transport                       :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of bytes. Note, the value in Length field will NOT cover any padding at the end of the parameter.

Server ID: 32 bit (unsigned integer)
This is the ID of the ENRP server, as defined in [I-D.ietf-rserpool-enrp].

Server Transport:
This is an SCTP Transport Parameter, as defined in Section 3.4 that contains the network access address(es), SCTP port number, etc. of the ENRP server.

3.12. Operation Error Parameter

This parameter is used in both ENRP and ASAP for a message sender to report an error(s) to a message receiver.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = 0xc           |       Length=variable         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                     one or more Error Causes                  :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of bytes. Note, the value in Length field will NOT cover any padding at the end of the parameter.

Error causes are defined as variable-length parameters using the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Cause Code          |       Cause Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                     Cause-specific Information                :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Cause Code: 16 bits (unsigned integer)
Defines the type of error condition being reported.

Cause Code Value	Cause Code
0x0	Unspecified Error
0x1	Unrecognized Parameter
0x2	Unrecognized Message
0x3	Invalid Values
0x4	Non-unique PE Identifier
0x5	Inconsistent Pooling Policy
0x6	Lack of Resources
0x7	Inconsistent Transport Type
0x8	Inconsistent Data/Control Configuration
0x9	Unknown Pool Handle
0xa	Rejected due to security considerations
others	reserved by IETF

Table 2

Cause Length: 16 bits (unsigned integer)
Set to the size of the parameter in bytes, including the Cause Code, Cause Length, and Cause-Specific Information fields, but not including any padding at the end of this error cause TLV.

Cause-specific Information: variable length
This field carries the details of the error condition.

The following subsections (Section 3.12.1 - Section 3.12.9) define specific error causes.

3.12.1. Unspecified Error

This error cause is used to report an unspecified error by the sender. There is no cause specific information.

3.12.2. Unrecognized Parameter Error

This error cause is used to report an unrecognized parameter. The complete unrecognized parameter TLV is included as cause specific information. If a message contains multiple unrecognized parameters, multiple error causes are used.

3.12.3. Unrecognized Message Error

This error cause is used to report an unrecognized message. The unrecognized message TLV is included as cause specific information.

3.12.4. Invalid Values Error

This error cause is used to report one or more invalid values found in a received parameter. The offending TLV that contains the invalid value(s) is included as cause specific information.

3.12.5. Non-unique PE Identifier Error

This error cause is used by an ENRP server to indicate to a registering PE that the PE Identifier it chooses has already been used by another PE in the pool. There is no cause specific information.

3.12.6. Inconsistent Pool Policy Error

This error cause is used by an ENRP server to indicate to a registering PE that the Pool Policy it chooses does not match the overall policy of the pool. A Pool Member Selection Policy TLV (see Section 3.8) that indicates the overall pool policy is included as cause specific information.

3.12.7. Lack of Resources Error

This error cause is used to indicate that the sender does not have certain resources to perform a requested function. There is no cause specific information.

3.12.8. Inconsistent Transport Type Error

This error cause is used by an ENRP server to indicate to a registering PE that the User Transport it chooses does not match the overall user transport of the pool. A Transport TLV that indicates the overall pool user transport type is included as cause specific information.

3.12.9. Inconsistent Data/Control Configuration Error

This error cause is used by an ENRP server to indicate to a registering PE that the Transport Use field in the User Transport it sent in its registration is inconsistent to the pool's overall data/control channel configuration. There is no cause specific information.

3.12.10. Rejected due to security considerations

This error cause is used by any endpoint to indicate a rejection of a request due to a failure in security credentials or authorizations.

3.12.11. Unknown Pool Handle Error

This error cause is used by an ENRP server to indicate to a PE or PU that the requested pool is unknown by the server. There is no cause specific information.

3.13. Cookie Parameter

This parameter defines a TLV that carries a Cookie.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 0xd         |       Length=variable         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                             Cookie                            :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of bytes, including the Type, Length, and Cookie.

Cookie: variable length The Cookie is an arbitrary byte string of (Length - 4) bytes.

3.14. PE Identifier Parameter

This parameter defines a TLV that carries a PE Identifier.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 0xe         |         Length=0x8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         PE Identifier                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PE Identifier: 32 bits (unsigned integer)
Uniquely identifies the PE in the pool. The PE picks its identifier when it starts up. See [I-D.ietf-rserpool-asap] for recommendations on PE identifier generation.

3.15. PE Checksum Parameter

This parameter defines a TLV that carries a PE Checksum.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 0xf         |         Length=0x6            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           PE Checksum         |          Padding              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PE Checksum: 16 bits (unsigned integer)
An overall checksum of all PEs in the current handlespace owned by an ENRP server (which is normally the sender of this TLV). The definition and calculation of this checksum is defined in [I-D.ietf-rserpool-enrp].

3.16. Opaque Transport Parameter

This parameter defines a TLV that carries opaque transport information.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 0x10        |        Length=variable        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                     Opaque Transport Data                     :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Length: 16 bits (unsigned integer)
Indicates the entire length of the parameter in number of bytes, including the Type, Length, and Opaque Transport Data.

Opaque Transport Data: variable length
The Opaque Transport Data is an arbitrary byte string of (Length - 4) bytes.

4. Common Message Formats

The figure below illustrates the common format for all ASAP and ENRP messages. Each message is formatted with a Message Type field, a message-specific Flag field, a Message Length field, and a Value field.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  |   Msg Flags   |        Message Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                        Message Value                          :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type: 8 bits (unsigned integer)
This field identifies the type of information contained in the Message Value field. It takes a value from 0 to 254. The value of 255 is reserved for future use as an extension field.
Message Types are encoded such that the highest-order two bits specify the action that must be taken if the message receiver does not recognize the Message Type.

00 Stop processing this message and discard it.

01 Stop processing this message and discard it, and report the unrecognized message in an 'Unrecognized Message' error (see Section 3.12.3).

10 reserved.

11 reserved.

Message Flags: 8 bits
The usage of these bits depends on the message type as given by the Message Type. Unless otherwise specified, they are set to zero on transmit and are ignored on receipt.

Message Length: 16 bits (unsigned integer)
This value represents the size of the message in bytes including the Message Type, Message Flags, Message Length, and Message Value fields. Therefore, if the Message Value field is zero-length, the Length field will be set to 4.
Note, the value in Message Length field will NOT cover any padding at the end of this message.

Message Value: variable length

The Message Value field contains the actual information to be transferred in the message. The usage and format of this field is dependent on the Message Type.

The total length of a message (including Type, Length and Value fields) MUST be a multiple of 4 bytes. If the length of the message is not a multiple of 4 bytes, the sender MUST pad the message with all zero bytes and this padding is not included in the message length field. The sender should never pad with more than 3 bytes. The receiver MUST ignore the padding bytes.

5. IANA Considerations

[NOTE to RFC-Editor:

"RFCXXXX" is to be replaced by the RFC number you assign this document.

]

This document (RFCXXX) is the reference for all registrations described in this section. All registrations need to be listed on an RSerPool specific page.

5.1. A New Table for RSerPool Parameter Types

RSerPool Parameter Types have to be maintained by IANA. Thirteen initial values should be assigned by IANA as described in Table 1. This requires a new table "RSerPool Parameter Types":

Value	Parameter Type
0x0	(reserved by IETF)
0x1	IPv4 Address
0x2	IPv6 Address
0x3	DCCP Transport
0x4	SCTP Transport
0x5	TCP Transport
0x6	UDP Transport
0x7	UDP-Lite
0x8	Pool Member Selection Policy
0x9	Pool Handle
0xa	Pool Element
0xb	Server Information
0xc	Operation Error
0xd	Cookie
0xe	PE Identifier
0xf	PE Checksum
0x10	Opaque Transport
others	(reserved by IETF)
0xffffffff	IETF-defined extensions

For registering at IANA an RSerPool Parameter Type in this table a request has to be made to assign such a number. This number must be unique. The "Specification Required" policy of [RFC5226] MUST be applied.

5.2. A New Table for RSerPool Error Causes

RSerPool Error Causes have to be maintained by IANA. Eleven initial values should be assigned by IANA as described in Table 2. This requires a new table "RSerPool Error Causes":

Cause Code Value	Cause Code
0x0	Unspecified Error
0x1	Unrecognized Parameter
0x2	Unrecognized Message
0x3	Invalid Values
0x4	Non-unique PE Identifier
0x5	Inconsistent Pooling Policy
0x6	Lack of Resources
0x7	Inconsistent Transport Type
0x8	Inconsistent Data/Control Configuration
0x9	Unknown Pool Handle
0xa	Rejected due to security considerations
others	reserved by IETF

For registering at IANA an RSerPool Error Cause in this table a request has to be made to assign such a number. This number must be unique. The "Specification Required" policy of [RFC5226] MUST be applied.

6. Security Considerations

This document contains common parameter formats only. As such it specifies no new security constraints on either ENRP or ASAP. Details on ENRP and ASAP security constraints are addressed in [I-D.ietf-rserpool-enrp] and [I-D.ietf-rserpool-asap].

7. Normative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[I-D.ietf-rserpool-asap] Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP)", draft-ietf-rserpool-asap-21 (work in progress), July 2008.

[I-D.ietf-rserpool-enrp] Xie, Q., Stewart, R., Stillman, M., Tuexen, M., and A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)", draft-ietf-rserpool-enrp-21 (work in progress), July 2008.

[I-D.ietf-rserpool-policies] Tuexen, M. and T. Dreibholz, "Reliable Server Pooling Policies", draft-ietf-rserpool-policies-10 (work in progress), July 2008.

Authors' Addresses

Randall R. Stewart
The Resource Group
1700 Pennsylvania Ave NW
Suite 56
Washington, DC 20006
USA

Phone:
Email: randall.stewart@trgworld.com

Qiaobing Xie
USA

Phone: +1 224-465-5954
Email: Qiaobing.Xie@gmail.org

Maureen Stillman
Nokia
127 W. State Street
Ithaca, NY 14850
USA

Email: maureen.stillman@nokia.com

Michael Tuexen
Muenster Univ. of Applied Sciences
Stegerwaldstr. 39
48565 Steinfurt
Germany

Email: tuexen@fh-muenster.de

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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.