

Network Working Group
Request for Comments: 3103
Category: Experimental

M. Borella
D. Grabelsky
CommWorks
J. Lo
Candlestick Networks
K. Taniguchi
NEC USA
October 2001

Realm Specific IP: Protocol Specification

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

IESG Note

The IESG notes that the set of documents describing the RSIP technology imply significant host and gateway changes for a complete implementation. In addition, the floating of port numbers can cause problems for some applications, preventing an RSIP-enabled host from interoperating transparently with existing applications in some cases (e.g., IPsec). Finally, there may be significant operational complexities associated with using RSIP. Some of these and other complications are outlined in section 6 of the RFC 3102, as well as in the Appendices of RFC 3104. Accordingly, the costs and benefits of using RSIP should be carefully weighed against other means of relieving address shortage.

Abstract

This document presents a protocol with which to implement Realm Specific IP (RSIP). The protocol defined herein allows negotiation of resources between an RSIP host and gateway, so that the host can lease some of the gateway's addressing parameters in order to establish a global network presence. This protocol is designed to operate on the application layer and to use its own TCP or UDP port. In particular, the protocol allows a gateway to allocate addressing and control parameters to a host such that a flow policy can be enforced at the gateway.

Table of Contents

1. Introduction	3
2. Specification of Requirements	4
3. Terminology	4
4. Architecture	5
5. Transport Protocol	7
6. Host / Gateway Relationships	7
7. Gateway Flow Policy and State	8
7.1. Local Flow Policy	9
7.2. Remote Flow Policy	9
7.3. Gateway State	10
8. Parameter Specification and Formats	11
8.1. Address	11
8.2. Ports	12
8.3. Lease Time	13
8.4. Client ID	13
8.5. Bind ID	13
8.6. Tunnel Type	14
8.7. RSIP Method	14
8.8. 8.8. Error	14
8.9. Flow Policy	15
8.10. Indicator	15
8.11. Message Counter	16
8.12. Vendor Specific Parameter	16
9. Message Types	16
9.1. ERROR_RESPONSE	17
9.2. REGISTER_REQUEST	18
9.3. REGISTER_RESPONSE	19
9.4. DE-REGISTER_REQUEST	19
9.5. DE-REGISTER_RESPONSE	20
9.6. ASSIGN_REQUEST_RSA-IP	21
9.7. ASSIGN_RESPONSE_RSA-IP	22
9.8. ASSIGN_REQUEST_RSAP-IP	23
9.9. ASSIGN_RESPONSE_RSAP-IP	26
9.10. EXTEND_REQUEST	27
9.11. EXTEND_RESPONSE	28
9.12. FREE_REQUEST	28
9.13. FREE_RESPONSE	29
9.14. QUERY_REQUEST	30
9.15. QUERY_RESPONSE	31
9.16. LISTEN_REQUEST	32
9.17. LISTEN_RESPONSE	35
10. Discussion	36
10.1. Use of Message Counters	36
10.2. RSIP Host and Gateway Failure Scenarios	37
10.3. General Gateway Policy	38
10.4. Errors Not From the RSIP Protocol	39

10.5. Address and Port Requests and Allocation	40
10.6. Local Gateways and Flow Policy Interaction	40
11. Security Considerations	40
12. IANA Considerations	41
13. Acknowledgements	41
14. Appendix A: RSIP Error Numbers	42
15. Appendix B: Message Types	44
16. Appendix C: Example RSIP host/gateway transactions	45
17. Appendix D: Example RSIP host state diagram	50
18. References	52
19. Authors' Addresses	53
20. Full Copyright Statement	54

1. Introduction

Network Address Translation (NAT) has gained popularity as a method of separating public and private address spaces, and alleviating network address shortages. A NAT translates the addresses of packets leaving a first routing realm to an address from a second routing realm, and performs the reverse function for packets entering the first routing realm from the second routing realm. This translation is performed transparently to the hosts in either space, and may include modification of TCP/UDP port numbers and IP addresses in packets that traverse the NAT.

While a NAT does not require hosts to be aware of the translation, it will require an application layer gateway (ALG) for any protocol that transmits IP addresses or port numbers in packet payloads (such as FTP). Additionally, a NAT will not work with protocols that require IP addresses and ports to remain unmodified between the source and destination hosts, or protocols that prevent such modifications from occurring (such as some IPsec modes, or application-layer end-to-end encryption).

An alternative to a NAT is an architecture that allows the hosts within the first (e.g., private) routing realm to directly use addresses and other routing parameters from the second (e.g., public) routing realm. Thus, RSIP [RSIP-FRAME] has been defined as a method for address sharing that exhibits more transparency than NAT. In particular, RSIP requires that an RSIP gateway (a router or gateway between the two realms) assign at least one address from the second routing realm, and perhaps some other resources, to each RSIP host. An RSIP host is a host in the first routing realm that needs to establish end-to-end connectivity to a host, entity or device in the second routing realm. Thus, the second routing realm is not directly

accessible from the RSIP host, but this system allows packets to maintain their integrity from the RSIP host to their destination. ALGs are not required in the RSIP gateway.

RSIP requires that hosts be modified so that they place some number of layer three, layer four or other values from those assigned by the RSIP gateway in each packet bound for the second routing realm.

This document discusses a method for assigning parameters to an RSIP host from an RSIP gateway. The requirements, scope, and applicability of RSIP, as well as its interaction with other layer 3 protocols, are discussed in a companion framework document [RSIP-FRAME]. Extensions to this protocol that enable end-to-end IPsec are discussed in [RSIP-IPSEC].

2. Specification of Requirements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "SHALL", "SHALL NOT", "MAY" and "MAY NOT" that appear in this document are to be interpreted as described in [RFC2119].

3. Terminology

Private Realm

A routing realm that uses private IP addresses from the ranges (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) specified in [RFC1918], or addresses that are non-routable from the Internet.

Public Realm

A routing realm with unique network addresses assigned by the Internet Assigned Number Authority (IANA) or an equivalent address registry.

RSIP Host

A host within the private realm that acquires publicly unique parameters from an RSIP gateway through the use of the RSIP client/server protocol.

RSIP Gateway

A router situated on the boundary between a private realm and a public realm and owns one or more public IP addresses. An RSIP gateway is responsible for public parameter management and assignment to RSIP hosts. An RSIP gateway may act as a NAT router for hosts within the private realm that are not RSIP enabled.

RSIP Client

An application program that performs the client portion of the RSIP client/server protocol. An RSIP client application **MUST** exist on all RSIP hosts, and **MAY** exist on RSIP gateways.

RSIP Server

An application program that performs the server portion of the RSIP client/server protocol. An RSIP server application **MUST** exist on all RSIP gateways.

RSA-IP: Realm Specific Address IP

An RSIP method in which each RSIP host is allocated a unique IP address from the public realm. Discussed in detail in [RSIP-FRAME]

RSAP-IP: Realm Specific Address and Port IP

An RSIP method in which each RSIP host is allocated an IP address (possibly shared with other RSIP hosts) and some number of per-address unique ports from the public realm. Discussed in detail in [RSIP-FRAME]

Binding

An association of some combination of a local address, one or more local ports, a remote address, and a remote port with an RSIP host.

Resource

A general way to refer to an item that an RSIP host leases from an RSIP gateway; e.g., an address or port.

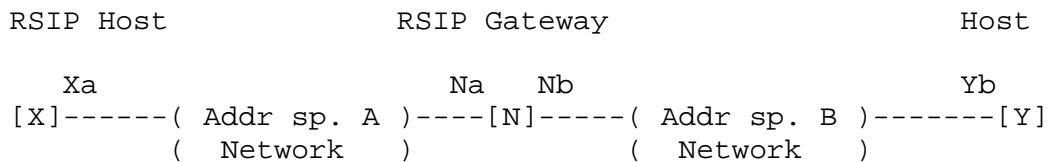
All other terminology found in this document is consistent with that of [RFC2663] and [RSIP-FRAME].

4. Architecture

For simplicity, in the remainder of this document we will assume that the RSIP hosts in the first routing realm (network) use private (e.g., see [RFC1918]) IP addresses, and that the second routing realm (network) uses public IP addresses. (This assumption is made without loss of generality and the ensuing discussion applies to more general

cases.) The RSIP gateway connects the public and private realms and contains interfaces to both. Other NAT terminology found in this document is defined in [RFC2663].

The diagram below describes an exemplary reference architecture for RSIP.



Hosts X and Y belong to different addressing realms A and B, respectively, and N is an RSIP gateway (which may also perform NAT functions). N has two interfaces: Na on address space A, and Nb on address space B. N may have a pool of addresses in address space B which it can assign to or lend to X and other hosts in address space

A. These addresses are not shown above, but they can be denoted as Nb1, Nb2, Nb3 and so on.

Host X, needing to establish an end-to-end connection to a network entity Y situated within address space B, first negotiates and obtains assignment of the resources from the RSIP gateway. Upon assignment of these parameters, the RSIP gateway creates a mapping, of X's addressing information and the assigned resources. This binding enables the RSIP gateway to correctly de-multiplex and forward inbound traffic generated by Y for X. A lease time is associated with each bind.

Using the public parameters assigned by the RSIP gateway, RSIP hosts tunnel data packets across address space A to the RSIP gateway. The RSIP gateway acts as the end point of such tunnels, stripping off the outer headers and routing the inner packets onto the public realm. As mentioned above, an RSIP gateway maintains a mapping of the assigned public parameters as demultiplexing fields for uniquely mapping them to RSIP host private addresses. When a packet from the public realm arrives at the RSIP gateway and it matches a given set of demultiplexing fields, then the RSIP gateway will tunnel it to the appropriate RSIP host. The tunnel headers of outbound packets from X to Y, given that X has been assigned Nb, are as follows:

```

+-----+-----+-----+
| X -> Na | Nb -> Y | payload |
+-----+-----+-----+
```


There are two basic flavors of RSIP: RSA-IP and RSAP-IP. RSIP hosts and gateways MUST support RSAP-IP and MAY support RSA-IP. Details of RSA-IP and RSAP-IP are found in [RSIP-FRAME].

5. Transport Protocol

RSIP is an application layer protocol that requires the use of a transport layer protocol for end-to-end delivery of packets.

RSIP gateways MUST support TCP, and SHOULD support UDP. Due to the fact that RSIP may be deployed across a wide variety of network links, RSIP hosts SHOULD support TCP, because of TCP's robustness across said variety of links. However, RSIP hosts MAY support UDP instead of TCP, or both UDP and TCP.

For RSIP hosts and gateways using UDP, timeout and retransmissions MUST occur. We recommend a binary exponential backoff scheme with an initial duration of 12.5 ms, and a maximum of six retries (seven total attempts before failure). However, these parameters MAY be adjusted or tuned for specific link types or scenarios.

Once a host and gateway have established a registration using either TCP or UDP, they may not switch between the two protocols for the duration of the registration. The decision of whether to use TCP or UDP is made by the client, and is determined by the transport protocol of the first packet sent by a client in a successful registration procedure.

6. Host / Gateway Relationships

An RSIP host can be in exactly one of three fundamental relationships with respect to an RSIP gateway:

Unregistered: The RSIP gateway does not know of the RSIP host's existence, and it will not forward or deliver globally addressed packets on behalf of the host. The only valid RSIP-related action for an RSIP host to perform in this state is to request registration with an RSIP gateway.

Registered: The RSIP gateway knows of the RSIP host and has assigned it a client ID and has specified the flow policies that it requires of the host. However, no resources, such as addresses or ports, have been allocated to the host, and the gateway will not forward or deliver globally addressed packets on behalf of the host. All registrations have an associated lease time. If this lease time expires, the RSIP host automatically reverts to the unregistered state.

Assigned: The RSIP gateway has granted one or more bindings of resources to the host. The gateway will forward and deliver globally addressed packets on behalf of the host. Each binding has an associated lease time. If this lease time expires, the binding is automatically revoked.

Architectures in which an RSIP host is simultaneously registered with more than one RSIP gateway are possible. In such cases, an RSIP host may be in different relationships with different RSIP gateways at the same time.

An RSIP gateway MAY redirect an RSIP host to use a tunnel endpoint for data traffic that is not the RSIP gateway itself, or perhaps is a different interface on the RSIP gateway. This is done by specifying the tunnel endpoint's address as part of an assignment. In such an architecture, it is desirable (though not necessary) for the RSIP gateway to have a method with which to notify the tunnel endpoint of assignments, and the expiration status of these assignments.

Lease times for bindings and registrations are managed as follows. All lease times are given in units of seconds from the current time, indicating a time in the future at which the lease will expire. These expiration times are used in the ensuing discussion.

An initial expiration time (R) is given to a registration. Under this registration, multiple bindings may be established, each with their own expiration times (B_1, B_2, \dots). When each binding is established or extended, the registration expiration time is adjusted so that the registration will last at least as long as the longest lease. In other words, when binding B_i is established or extended, the following calculation is performed: $R = \max(R, B_i)$.

Under this scheme, a registration will never expire while any binding's lease is still valid. However, a registration may expire when the last binding's lease expires, or at some point thereafter.

7. Gateway Flow Policy and State

Since an RSIP gateway is likely to reside on the boundary between two or more different administrative domains, it is desirable to enable an RSIP gateway to be able to enforce flow-based policy. In other words, an RSIP gateway should have the ability to explicitly control which local addresses and ports are used to communicate with remote addresses and ports.

In the following, macro-flow policy refers to controlling flow policy at the granularity level of IP addresses, while micro-flow policy refers to controlling flow policy at the granularity of IP address

and port tuples. Of course there may be no policy at all, which indicates that the RSIP gateway does not care about the flow parameters used by RSIP hosts. We consider two levels of local flow policy and three levels of remote flow policy.

7.1. Local Flow Policy

Local flow policy determines the granularity of control that an RSIP gateway has over the local addressing parameters that an RSIP host uses for particular sessions.

Since an RSIP host must use at least an IP address allocated by the gateway, the loosest level of local flow policy is macro-flow based. Under local macro-flow policy, an RSIP host is allocated an IP address (RSA-IP) or an IP address and one or more ports to use with it (RSAP-IP). However, the host may use the ports as it desires for establishing sessions with public hosts.

Under micro-flow policy, a host is allocated exactly one port at a time. The host may request more ports, also one at a time. This policy gives the gateway very tight control over local port use, although it affords the host less flexibility.

Note that only local macro-flow policy can be used with RSA-IP, while either local macro-flow or local micro-flow policy may be used with RSAP-IP.

Examples of how RSIP flow policy operates are given in Appendix C.

7.2. Remote Flow Policy

Remote flow policy determines the granularity of control that an RSIP gateway has over the remote (public) hosts with which an RSIP host communicates. In particular, remote flow policy dictates what level of detail that a host must specify addressing parameters of a remote host or application before the RSIP gateway allows the host to communicate with that host or application.

The simplest and loosest form of flow policy is no policy at all. In other words, the RSIP gateway allocates addressing parameters to the host, and the host may use these parameters to communicate with any remote host, without explicitly notifying the gateway.

Macro-flow policy requires that the host identify the remote address of the host that it wishes to communicate with as part of its request for local addressing parameters. If the request is granted, the host **MUST** use the specified local parameters only with the remote address specified, and **MUST NOT** communicate with the remote address using any

local parameters but the ones allocated. However, the host may contact any port number at the remote host without explicitly notifying the gateway.

Micro-flow policy requires that the host identify the remote address and port of the host that it wishes to communicate with as part of its request for local addressing parameters. If the request is granted, the host **MUST** use the specified local parameters only with the remote address and port specified, and **MUST NOT** communicate with the remote address and port using any local parameters but the ones allocated.

Remote flow policy is implemented in both the ingress and egress directions, with respect to the location of the RSIP gateway.

7.3. Gateway State

An RSIP gateway must maintain state for all RSIP hosts and their assigned resources. The amount and type of state maintained depends on the local and remote flow policy. The required RSIP gateway state will vary based on the RSIP method, but will always include the chosen method's demultiplexing parameters.

7.3.1. RSA-IP State

An RSIP gateway serving an RSIP host using the RSA-IP method **MUST** maintain the following minimum state to ensure proper mapping of incoming packets to RSIP hosts:

- Host's private address
- Host's assigned public address(es)

7.3.2. RSAP-IP State

An RSIP gateway serving an RSIP host using the RSAP-IP method **MUST** maintain the following minimum state to ensure proper mapping of incoming packets to RSIP hosts:

- Host's private address
- Host's assigned public address(es)
- Host's assigned port(s) per address

7.3.3. Flow State

Regardless of whether the gateway is using RSA-IP or RSAP-IP, additional state is necessary if either micro-flow based or macro-flow based remote policy is used.

If the gateway is using macro-flow based remote policy, the following state must be maintained:

- Remote host's address

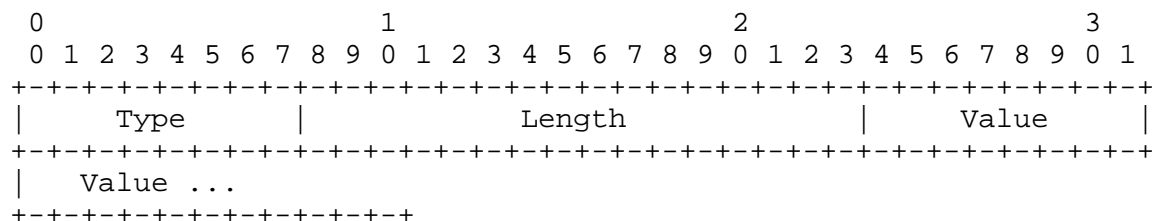
If the gateway is using micro-flow based remote policy, the following state must be maintained:

- Remote host's address
- Remote host's port

More state MAY be used by an RSIP gateway if desired. For example, ToS/DS bytes may be recorded in order to facilitate quality of service support.

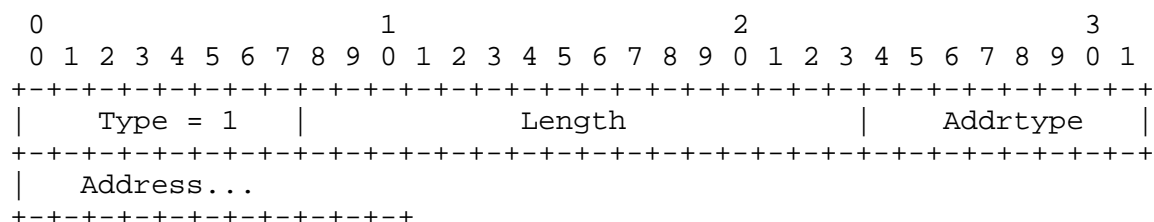
8. Parameter Specification and Formats

In this section we define the formats for RSIP parameters. Each RSIP message contains one or more parameters that encode the information passed between the host and gateway. The general format of all parameters is TLV (type-length-value) consisting of a 1-byte type followed by a 2-byte length followed by a 'length' byte value as shown below.



The value field may be divided into a number of other fields as per the type of the parameter. Note that the length field encodes the number of bytes in the value field, NOT the overall number of bytes in the parameter.

8.1. Address



The address parameter contains addressing information, either an IPv4 address or netmask, an IPv6 address or netmask, or a fully qualified domain name (FQDN). The Addrtype field is 1 byte in length, indicating the type of address.

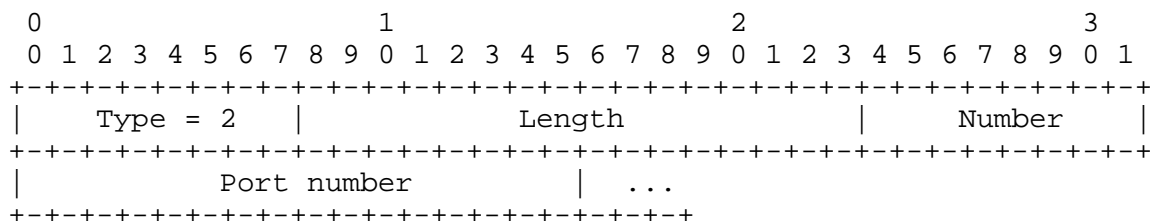
	Addrtype	Length of address field (in bytes)
	----	-----
0	Reserved	0
1	IPv4	4
2	IPv4 netmask	4
3	IPv6	16
4	FQDN	varies

For FQDN (Fully qualified domain name), the length of the address field will be one less than the value of the length field, and the name will be represented as an ASCII string (no terminating character).

In some cases, it is necessary to specify a "don't care" value for an address. This is signified by a setting the length field to 1 and omitting the value field.

It is not valid for a host to request an address with an FQDN type as its local address (See specification of ASSIGN_REQUEST_RSA-IP and ASSIGN_REQUEST_RSAP-IP, below).

8.2. Ports



The ports parameter encodes zero or more TCP or UDP ports. When a single port is specified, the value of the number field is 1 and there is one port field following the number field. When more than one port is specified, the value of the number field will indicate the total number of ports contained, and the parameter may take one of two forms. If there is one port field, the ports specified are considered to be contiguous starting at the port number specified in the port field. Alternatively, there may be a number of port fields equal to the value of the number field. The number of port fields can be extrapolated from the length field.

In some cases, it is necessary to specify a don't care value for one or more ports (e.g., when a client application is using ephemeral source ports). This is accomplished by setting the length field to 1, setting the number field to the number of ports necessary, and omitting all port fields. The value of the number field **MUST** be greater than or equal to one.

If micro-flow based policy applies to a given ports parameter, it **MUST** contain exactly one port field.

8.3. Lease Time

```

      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 = 3      |      Length = 4      |      Lease time  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Lease time    |
+-----+-----+-----+-----+-----+-----+-----+

```

The lease time parameter specifies the length, in seconds, of an RSIP host registration or parameter binding.

8.4. Client ID

```

      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 = 4      |      Length = 4      |      Client ID  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Client ID     |
+-----+-----+-----+-----+-----+-----+-----+

```

The client ID parameter specifies an RSIP client ID. Client ID's by an RSIP gateway to differentiate RSIP hosts.

8.5. Bind ID

```

      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 = 5      |      Length = 4      |      Bind ID      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Bind ID       |
+-----+-----+-----+-----+-----+-----+-----+

```

The bind ID parameter specifies an RSIP bind ID. Bind ID's are used by RSIP hosts and gateways to differentiate an RSIP host's bindings.

8.6. Tunnel Type

```

      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 = 6   |             Length = 1             | Tunnel type |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The tunnel type parameter specifies the type of tunnel used between an RSIP host and an RSIP gateway. Defined tunnel types are:

	Tunnel Type

0	Reserved
1	IP-IP
2	GRE
3	L2TP

8.7. RSIP Method

```

      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 = 7   |             Length = 1             | RSIP method |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The RSIP method parameter specifies an RSIP method. Defined RSIP methods are:

	RSIP method

0	Reserved
1	RSA-IP
2	RSAP-IP

8.8. Error

```

      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 = 8   |             Length = 2             |      Error      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Error   |
+---+---+---+---+---+---+

```

The error parameter specifies an error. The currently defined error values are presented in Appendix A.

8.9. Flow Policy

```

      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 = 9   |               Length = 2               |   Local   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Remote   |
+---+---+---+---+---+---+

```

The flow policy parameter specifies both the local and remote flow policy.

Defined local flow policies are:

```

      Local Flow Policy
      -----
    0   Reserved
    1   Macro flows
    2   Micro flows

```

Defined remote flow policies are:

```

      Remote Flow Policy
      -----
    0   Reserved
    1   Macro flows
    2   Micro flows
    3   No policy

```

8.10. Indicator

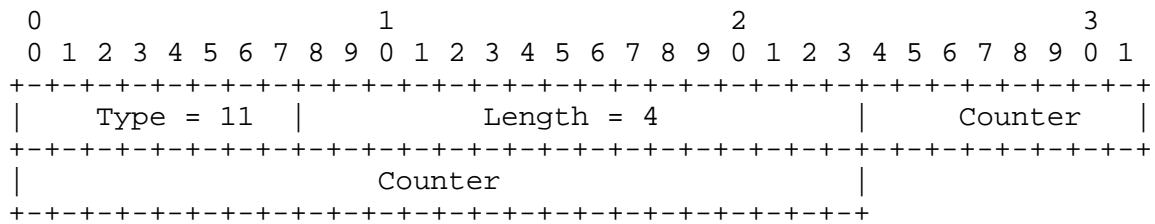
```

      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 = 10  |               Length = 2               |   Value   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Value   |
+---+---+---+---+---+---+

```

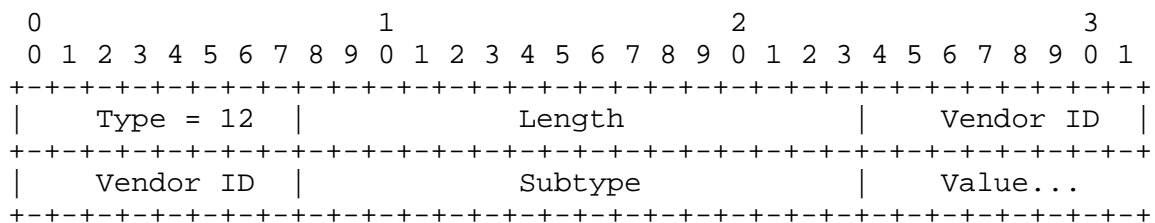
An indicator parameter is a general-purpose parameter, the use of which is defined by the message that it appears in. An RSIP message that uses an indicator parameter **MUST** define the meaning and interpretation of all of the indicator's possible values.

8.11. Message Counter



A message counter parameter is used to mark RSIP messages with sequentially-increasing values. Message counters **MUST** be used with UDP, in order to facilitate reliability.

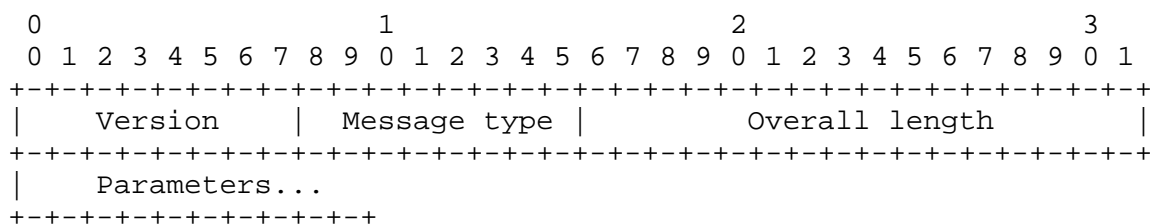
8.12. Vendor Specific Parameter



The vendor specific parameter is used to encode parameters that are defined by a particular vendor. The vendor ID field is the vendor-specific ID assigned by IANA. Subtypes are defined and used by each vendor as necessary. An RSIP host or gateway **SHOULD** silently ignore vendor-specific messages that it does not understand.

9. Message Types

RSIP messages consist of three mandatory fields, version, message type, and overall length, followed by one or more required parameters, followed in turn by zero or more optional parameters. In an RSIP message, all required parameters **MUST** appear in the exact order specified below. Optional parameters **MAY** appear in any order. Message format is shown below:



The version number field is a single byte and specifies the RSIP version number that is being used. The current RSIP version number is 1.

The message type field is a single byte and specifies the message contained in the current packet. There may be only one message per packet. Message types are given numerical assignments in Appendix B.

The overall length field is two bytes and contains the number of bytes in the RSIP message, including the three mandatory fields.

Most parameters are only allowed to appear once in each message. The exceptions are as follows:

- Multiple address parameters MUST appear in ASSIGN_REQUEST_RSA-IP, ASSIGN_RESPONSE_RSA-IP, ASSIGN_REQUEST_RSAP-IP, ASSIGN_RESPONSE_RSAP-IP, LISTEN_REQUEST and LISTEN_RESPONSE.
- Multiple ports parameters MUST appear in ASSIGN_REQUEST_RSAP-IP, ASSIGN_RESPONSE_RSAP-IP, LISTEN_REQUEST and LISTEN_RESPONSE.
- Multiple RSIP method and tunnel type parameters MAY appear in REGISTER_RESPONSE.
- Multiple address parameters and multiple indicator parameters MAY appear in QUERY_REQUEST and QUERY_RESPONSE.

The following message types are defined in BNF. Required parameters are enclosed in <> and MUST appear. Optional parameters are enclosed in [] and MAY appear. Not all message types need to be implemented in order to be RSIP compliant. For example, an RSIP host and/or gateway may not support LISTEN_REQUEST and LISTEN_RESPONSE, or may only support RSAP-IP and not RSA-IP.

9.1. ERROR_RESPONSE

9.1.1. Description

An ERROR_RESPONSE is used to provide error messages from an RSIP gateway to an RSIP host. Usually, errors indicate that the RSIP gateway cannot or will not perform an action or allocate resources on behalf of the host. If the error is related to a particular client ID or bind ID, these associated parameters MUST be included. Multiple errors MAY NOT be reported in the same ERROR_RESPONSE. In situations where more than one error has occurred, the RSIP gateway MUST choose only one error to report.

9.1.2. Format

```
<ERROR_RESPONSE> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Error>
                        [Message Counter]
                        [Client ID]
                        [Bind ID]
```

9.1.3. Behavior

An ERROR_RESPONSE message MUST only be transmitted by an RSIP gateway. An RSIP host that detects an error in a message received from an RSIP gateway MUST silently discard the message. There are no error conditions that can be caused by an ERROR_RESPONSE. An ERROR_RESPONSE is typically transmitted in response to a request from an RSIP host, but also may be transmitted asynchronously by an RSIP gateway.

9.2. REGISTER_REQUEST

9.2.1. Description

The REGISTER_REQUEST message is used by an RSIP host to establish registration with an RSIP gateway. An RSIP host MUST register before it requests resources or services from an RSIP gateway. Once an RSIP host has registered with an RSIP gateway, it may not register again until it has de-registered from that gateway.

9.2.2. Format

```
<REGISTER_REQUEST> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        [Message Counter]
```

9.2.3. Behavior

The following message-specific error conditions exist:

- If the host is already registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the ALREADY_REGISTERED error and the RSIP host's client ID.
- If the gateway's policy will not allow the host to register, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTRATION_DENIED error.

9.3. REGISTER_RESPONSE

9.3.1. Description

The REGISTER_RESPONSE message is used by an RSIP gateway to confirm the registration of an RSIP host, and to provide a client ID, flow policy, and possibly a message counter and one or more RSIP methods and/or tunnel types.

9.3.2. Format

```
<REGISTER_RESPONSE> ::= <Version>
                           <Message Type>
                           <Overall Length>
                           <Client ID>
                           <Lease time>
                           <Flow Policy>
                           [Message Counter]
                           [RSIP Method]...
                           [Tunnel Type]...
```

9.3.3. Behavior

An RSIP gateway MUST assign a different client ID to each host that is simultaneously registered with it. The RSIP gateway MAY respond with one or more RSIP methods and tunnel types that it supports. If an RSIP method is not specified, RSAP-IP MUST be assumed. If a tunnel type is not specified, IP-IP MUST be assumed.

9.4. DE-REGISTER_REQUEST

9.4.1. Description

The DE-REGISTER_REQUEST message is used by an RSIP host to de-register with an RSIP gateway. If a host de-registers from the assigned state, all of the host's bindings are revoked. The host SHOULD NOT de-register from the unregistered state.

9.4.2. Format

```
<DE-REGISTER_REQUEST> ::= <Version>
                           <Message Type>
                           <Overall Length>
                           <Client ID>
                           [Message Counter]
```


9.4.3. Behavior

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.

If there are no errors that result from this message, the gateway MUST respond with an appropriate DE-REGISTER_RESPONSE. Upon de-registering a host, an RSIP gateway must delete all binds associated with that host and return their resources to the pool of free resources. Once a host has de-registered, it may not use any of the RSIP gateway's resources without registering again.

9.5. DE-REGISTER_RESPONSE

9.5.1. Description

The DE-REGISTER_RESPONSE message is used by an RSIP gateway to confirm the de-registration of an RSIP host or to force an RSIP host to relinquish all of its bindings and terminate its relationship with the RSIP gateway. Upon receiving a DE-REGISTER_RESPONSE message, an RSIP host MUST stop all use of the resources that have been allocated to it by the gateway.

9.5.2. Format

```
<DE-REGISTER_RESPONSE> ::= <Version>
                             <Message Type>
                             <Overall Length>
                             <Client ID>
                             [Message Counter]
```

9.5.3. Behavior

An RSIP gateway MUST send a DE-REGISTER_RESPONSE in response to a valid DE-REGISTER_REQUEST. An RSIP gateway MUST send a DE-REGISTER_RESPONSE to an RSIP host when that host's registration lease time times out. An RSIP gateway SHOULD send a DE-REGISTER_RESPONSE if it detects that it will no longer be able to perform RSIP functionality for a given host. An RSIP host MUST be ready to accept a DE-REGISTER_RESPONSE at any moment.

9.6. ASSIGN_REQUEST_RSA-IP

9.6.1. Description

The ASSIGN_REQUEST_RSA-IP message is used by an RSIP host to request resources to use with RSA-IP. Note that RSA-IP cannot be used in combination with micro-flow based local policy.

9.6.2. Format

```
<ASSIGN_REQUEST_RSA-IP> ::= <Version>
                             <Message Type>
                             <Overall Length>
                             <Client ID>
                             <Address (local)>
                             <Address (remote)>
                             <Ports (remote)>
                             [Message Counter]
                             [Lease Time]
                             [Tunnel Type]
```

9.6.3. Behavior

The RSIP host specifies two address parameters. The RSIP host may request a particular local address by placing that address in the first address parameter. To indicate that it has no preference for local address, the RSIP host may place a "don't care" value in the address parameter.

If macro-flow based remote policy is used, the host MUST specify the remote address that it will use this binding (if granted) to contact; however, the remote port number MAY remain unspecified. If micro-flow based remote policy is used, the host MUST specify the remote address and port number that it will use this binding (if granted) to contact. If no flow policy is used, the RSIP host may place a "don't care" value in the value fields of the respective address and ports parameters.

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.

- If the local address parameter is a don't care value and the RSIP gateway cannot allocate ANY addresses, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.
- If the local address parameter is not a don't care value there are three possible error conditions:
 - o If the RSIP gateway cannot allocate ANY addresses, it MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.
 - o If the RSIP gateway cannot allocate the requested address because it is in use, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_INUSE error.
 - o If the RSIP gateway cannot allocate the requested address because it is not allowed by policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNALLOWED error.
- If macro-flow based remote policy is used and the requested remote address is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDR_UNALLOWED error.
- If micro-flow based remote policy is used and the requested remote address / port pair is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDRPORT_UNALLOWED error.
- If an unsupported or unallowed tunnel type is specified, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the BAD_TUNNEL_TYPE error.
- If the host has not specified local or remote address or port information in enough detail, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the FLOW_POLICY_VIOLATION error.

9.7. ASSIGN_RESPONSE_RSA-IP

9.7.1. Description

The ASSIGN_RESPONSE_RSA-IP message is used by an RSIP gateway to deliver parameter assignments to an RSIP host using RSA-IP. A host-wise unique bind ID, lease time, and tunnel type must be provided for every assignment.

9.7.2. Format

```
<ASSIGN_RESPONSE_RSA-IP> ::= <Version>
                                <Message Type>
                                <Overall Length>
                                <Client ID>
                                <Bind ID>
                                <Address (local)>
                                <Address (remote)>
                                <Ports (remote)>
                                <Lease Time>
                                <Tunnel Type>
                                [Address (tunnel endpoint)]
                                [Message Counter]
```

9.7.3. Behavior

If no remote flow policy is used, the RSIP gateway MUST use "don't care" values for the remote address and ports parameters. If macro-flow based remote policy is used, the remote address parameter MUST contain the address specified in the associated request, and the remote ports parameter MUST contain a "don't care" value. If micro-flow based remote policy is used, the remote address and remote ports parameters MUST contain the address and port information specified in the associated request.

If the host detects an error or otherwise does not "understand" the gateway's response, it SHOULD send a FREE_REQUEST with the bind ID from the said ASSIGN_RESPONSE_RSA-IP. This will serve to help synchronize the states of the host and gateway.

The address of a tunnel endpoint that is not the RSIP gateway MAY be specified. If this parameter is not specified, the RSIP gateway MUST be assumed to be the tunnel endpoint.

9.8. ASSIGN_REQUEST_RSAP-IP

9.8.1. Description

The ASSIGN_REQUEST_RSAP-IP message is used by an RSIP host to request resources to use with RSAP-IP. The RSIP host specifies two address and two port parameters, the first of each, respectively, refer to the local address and port(s) that will be used, and the second of each, respectively, refer to the remote address and port(s) that will be contacted.

9.8.2. Format

```
<ASSIGN_REQUEST_RSAP-IP> ::= <Version>
                                <Message Type>
                                <Overall Length>
                                <Client ID>
                                <Address (local)>
                                <Ports (local)>
                                <Address (remote)>
                                <Ports (remote)>
                                [Message Counter]
                                [Lease Time]
                                [Tunnel Type]
```

9.8.3. Behavior

An RSIP host may request a particular local address by placing that address in the value field of the first address parameter. The RSIP host may request particular local ports by placing them in the first port parameter. To indicate that it has no preference for local address or ports, the RSIP host may place a "don't care" value in the respective address or ports parameters.

If macro-flow based remote policy is used, the host MUST specify the remote address that it will use this binding (if granted) to contact; however, the remote port number(s) MAY remain unspecified. If micro-flow based remote policy is used, the host MUST specify the remote address and port number(s) that it will use this binding (if granted) to contact. If no flow policy is used, the RSIP host may place a value of all 0's in the value fields of the respective address or port parameters.

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.
- If the local address parameter is a don't care value and the RSIP gateway cannot allocate ANY addresses, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.

- If the local address parameter is not a don't care value there are five possible error conditions:
 - o If the RSIP gateway cannot allocate ANY addresses, it MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.
 - o If the RSIP gateway cannot allocate the requested address because it is in use, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_INUSE error.
 - o If the RSIP gateway cannot allocate the requested address because it is not allowed by policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNALLOWED error.
 - o If the RSIP gateway cannot allocate a requested address / port tuple because it is in use, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDRPORT_INUSE error.
 - o If the RSIP gateway cannot allocate a requested address / port tuple because it is not allowed by policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDRPORT_UNALLOWED error.
- If the RSIP host requests a number of ports (greater than one), but does not specify particular port numbers (i.e., uses "don't care" values) the RSIP gateway cannot grant the entire request, the RSIP gateway MUST return an ERROR_RESPONSE containing the LOCAL_ADDRPORT_UNAVAILABLE error.
- If macro-flow based remote policy is used and the requested remote address is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDR_UNALLOWED error.
- If micro-flow based remote policy is used and the requested remote address / port pair is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDRPORT_UNALLOWED error.
- If an unsupported or unallowed tunnel type is specified, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the BAD_TUNNEL_TYPE error.

- If the host has not specified local or remote address or port information in enough detail, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the FLOW_POLICY_VIOLATION error.

9.9. ASSIGN_RESPONSE_RSAP-IP

9.9.1. Description

The ASSIGN_RESPONSE_RSAP-IP message is used by an RSIP gateway to deliver parameter assignments to an RSIP host. A host-wise unique bind ID, lease time, and tunnel type must be provided for every assignment.

9.9.2. Format

```
<ASSIGN_RESPONSE_RSAP-IP> ::= <Version>
                                <Message Type>
                                <Overall Length>
                                <Client ID>
                                <Bind ID>
                                <Address (local)>
                                <Ports (local)>
                                <Address (remote)>
                                <Ports (remote)>
                                <Lease Time>
                                <Tunnel Type>
                                [Address (tunnel endpoint)]
                                [Message Counter]
```

9.9.3. Behavior

Regardless of local flow policy, a local address and port(s) MUST be assigned to the host. If macro-flow based local policy is used, the host is assigned an address and one or more ports. If micro-flow based local policy is used, the host is assigned an address and exactly one port.

If no remote flow policy is used, the RSIP gateway MUST use "don't care" values for the remote address and ports parameters. If macro-flow based remote policy is used, the remote address parameter MUST contain the address specified in the associated request, and the remote ports parameter must contain a "don't care" value. If micro-flow based remote policy is used, the remote address and remote ports parameters MUST contain the address and port information specified in the associated request.

If the host detects an error or otherwise does not "understand" the gateway's response, it SHOULD send a FREE_REQUEST with the bind ID from the said ASSIGN_RESPONSE_RSAP-IP. This will serve to help synchronize the states of the host and gateway.

The address of a tunnel endpoint that is not the RSIP gateway MAY be specified. If this parameter is not specified, the RSIP gateway MUST be assumed to be the tunnel endpoint.

9.10. EXTEND_REQUEST

9.10.1. Description

The EXTEND_REQUEST message is used to request a lease extension to a current bind. It may be used with both RSA-IP and RSAP-IP. The host MUST specify its client ID and the bind ID in question, and it MAY suggest a lease time to the gateway.

9.10.2. Format

```
<EXTEND_REQUEST> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Client ID>
                        <Bind ID>
                        [Lease Time]
                        [Message Counter]
```

9.10.3. Behavior

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.
- If the message contains an incorrect bind ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_BIND_ID error.

If the RSIP gateway grants an extension to the host's lease, it MUST RESPOND with an appropriate EXTEND_RESPONSE message. If the lease is not renewed, the RSIP gateway MAY let it implicitly expire by doing nothing or make it explicitly expire by sending an appropriate FREE_RESPONSE message.

9.11. EXTEND_RESPONSE

9.11.1. Description

The EXTEND_RESPONSE message is used by an RSIP gateway to grant a requested lease extension. The gateway MUST specify the client ID of the host, the bind ID in question, and the new assigned lease time.

9.11.2. Format

```
<EXTEND_RESPONSE> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Client ID>
                        <Bind ID>
                        <Lease Time>
                        [Message Counter]
```

9.11.3. Behavior

The RSIP gateway will determine lease time as per its local policy. The returned time is to be interpreted as the number of seconds before the lease expires, counting from the time at which the message is sent/received.

9.12. FREE_REQUEST

9.12.1. Description

The FREE_REQUEST message is used by an RSIP host to free a binding. The given bind ID identifies the bind to be freed. Resources may only be freed using the granularity of a bind ID.

9.12.2. Format

```
<FREE_REQUEST> ::= <Version>
                    <Message Type>
                    <Overall Length>
                    <Client ID>
                    <Bind ID>
                    [Message Counter]
```


9.12.3. Behavior

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.
- If the message contains an incorrect bind ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_BIND_ID error.

If a host receives an error in response to a FREE_REQUEST, this may indicate that the host and gateway's states have become unsynchronized. Therefore, the host SHOULD make an effort to resynchronize, such as freeing resources then re-requesting them, or de-registering then re-registering.

9.13. FREE_RESPONSE

9.13.1. Description

The FREE_RESPONSE message is used by an RSIP gateway to acknowledge a FREE_REQUEST sent by an RSIP host, and to asynchronously deallocate resources granted to an RSIP host.

9.13.2. Format

```
<FREE_RESPONSE> ::= <Version>
                      <Message Type>
                      <Overall Length>
                      <Client ID>
                      <Bind ID>
                      [Message Counter]
```

9.13.3. Behavior

An RSIP host must always be ready to accept a FREE_RESPONSE, even if its lease on the specified bind ID is not yet expired.

9.14. QUERY_REQUEST

9.14.1. Description

A QUERY_REQUEST message is used by an RSIP host to ask an RSIP gateway whether or not a particular address or network is local or remote. The host uses this information to determine whether to contact the host(s) directly (in the local case), or via RSIP (in the remote case).

This message defines an indicator parameter with a 1-byte value field and 2 defined values:

- 1 address
- 2 network

9.14.2. Format

```
<QUERY_REQUEST> ::= <Version>
                    <Message Type>
                    <Overall Length>
                    <Client ID>
                    [Message Counter]
                    [Address Tuple]...
                    [Network Tuple]...
```

where

```
<Address Tuple> ::= <Indicator (address)>
                    <Address>
```

```
<Network Tuple> ::= <Indicator (network)>
                    <Address (network)>
                    <Address (netmask)>
```

9.14.3. Behavior

One or more address or network tuples may be specified. Each tuple encodes a request regarding the locality (local or remote) of the encoded address or network. If no tuple is specified, the RSIP gateway should interpret the message as a request for all tuples that it is willing to provide. Note that the FQDN form of the address parameter cannot be used to specify the address of a network, and only the netmask form of the address parameter can be used to specify the netmask of a network.

If an RSIP gateway cannot determine whether a queried host or network is local or remote, it SHOULD transmit a QUERY_RESPONSE with no response specified for the said host or network.

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.

9.15. QUERY_RESPONSE

9.15.1. Description

A QUERY_RESPONSE message is used by an RSIP gateway to answer a QUERY_REQUEST from an RSIP host.

This message defines an indicator parameter with a 1-byte value field and 4 defined values:

- 1 local address
- 2 local network
- 3 remote address
- 4 remote network

9.15.2. Format

```
<QUERY_RESPONSE> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Client ID>
                        [Message Counter]
                        [Local Address Tuple]...
                        [Local Network Tuple]...
                        [Remote Address Tuple]...
                        [Remote Network Tuple]...
```

where

```
<Local Address Tuple> ::= <Indicator (local address)>
                        <Address>
```

```
<Local Network Tuple> ::= <Indicator (local network)>
                        <Address (network)>
                        <Address (netmask)>
```

```
<Remote Address Tuple> ::= <Indicator (remote address)>
                        <Address>
```



```
<Remote Network Tuple> ::= <Indicator (remote network)>  
                           <Address (network)>  
                           <Address (netmask)>
```

9.15.3. Behavior

An RSIP gateway has some leeway in how it responds to a QUERY_REQUEST. It may just provide the information requested, if it can provide such information. It may provide its complete list of address and networks, in order to minimize the number of requests that the host needs to perform in the future. How an RSIP gateway responds may depend on network traffic considerations as well.

If an RSIP gateway sends a QUERY_RESPONSE that does not contain any tuples, or a QUERY_RESPONSE that does not contain a tuple that applies to an associated tuple in the associated QUERY_REQUEST, this should be interpreted that the RSIP gateway does not know whether the queried host or network is local or remote. Appropriate host behavior upon receipt of such a message is to assume that the queried host or network is remote.

Note that an RSIP gateway is not expected to maintain a complete list of all remote hosts and networks. In fact, a typical RSIP gateway will only maintain a list of the networks and hosts that it knows are local (private with respect to the RSIP host).

9.16. LISTEN_REQUEST

9.16.1. Description

A LISTEN_REQUEST message is sent by an RSIP host that wants to register a service on a particular address and port number. The host must include its client ID, local address parameter and ports parameters, and remote address and ports parameters. The client MAY suggest a lease time and one or more tunnel types.

9.16.2. Format

```
<LISTEN_REQUEST> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Client ID>
                        <Address (local)>
                        <Ports (local)>
                        <Address (remote)>
                        <Ports (remote)>
                        [Message Counter]
                        [Lease Time]
                        [Tunnel Type]...
```

9.16.3. Behavior

If the host wants to listen on a particular address or port, it may specify these in the address and ports parameters. Otherwise it may leave one or both of these parameters with "don't care" values.

If no remote flow policy is being used, the host MUST fill both the remote address and ports parameters with "don't care" values. If macro-flow based remote policy is used, the host MUST specify the remote address, but MAY or MAY NOT specify the remote port(s). If micro-flow based remote policy is used, the host MUST specify the remote address and ports parameter.

Once a LISTEN_REQUEST has been granted, the RSIP gateway MUST forward all packets destined to the address and port in question to the host, even if the remote host address and port tuple has not been previously contacted by the host.

LISTEN_REQUEST is not necessary for RSA-IP.

The following message-specific error conditions exist:

- If the host is not registered with the gateway, the gateway MUST respond with an ERROR_RESPONSE containing the REGISTER_FIRST error.
- If the message contains an incorrect client ID, the gateway MUST respond with an ERROR_RESPONSE containing the BAD_CLIENT_ID error.
- If the local address parameter is a don't care value and the RSIP gateway cannot allocate ANY addresses, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.

- If the local address parameter is not a don't care value there are five possible error conditions:
 - o If the RSIP gateway cannot allocate ANY addresses, it MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNAVAILABLE error.
 - o If the RSIP gateway cannot allocate the requested address because it is in use, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_INUSE error.
 - o If the RSIP gateway cannot allocate the requested address because it is not allowed by policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDR_UNALLOWED error.
 - o If the RSIP gateway cannot allocate the requested address / port tuple because it is in use, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDRPORT_INUSE error.
 - o If the RSIP gateway cannot allocate the requested address / port tuple because it is not allowed by policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the LOCAL_ADDRPORT_UNALLOWED error.
- If macro-flow based remote policy is used and the requested remote address is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDR_UNALLOWED error.
- If micro-flow based remote policy is used and the requested remote address / port pair is not allowed by the RSIP gateway's policy, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the REMOTE_ADDRPORT_UNALLOWED error.
- If an unsupported or unallowed tunnel type is specified, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the BAD_TUNNEL_TYPE error.
- If the host has not specified local or remote address or port information in enough detail, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the FLOW_POLICY_VIOLATION error.

9.17. LISTEN_RESPONSE

9.17.1. Description

A LISTEN_RESPONSE message is used by an RSIP gateway to respond to a LISTEN_REQUEST message from an RSIP host. The RSIP gateway MUST issue a bind ID, and specify the address and port which have been granted to the host. The gateway must also specify a tunnel type and lease time.

If no remote flow policy is being used, the gateway MUST fill both the remote address and ports parameters with "don't care" values. If macro-flow based remote policy is used, the gateway MUST specify the remote address, but MAY or MAY NOT specify the remote port(s). If micro-flow based remote policy is used, the gateway MUST specify the remote address and ports parameter.

9.17.2. Format

```
<LISTEN_RESPONSE> ::= <Version>
                        <Message Type>
                        <Overall Length>
                        <Client ID>
                        <Bind ID>
                        <Address (local)>
                        <Ports (local)>
                        <Address (remote)>
                        <Ports (remote)>
                        <Tunnel Type>
                        <Lease Time>
                        [Address (tunnel endpoint)]
                        [Message Counter]
```

9.17.3. Behavior

If no remote flow policy is being used, the gateway MUST fill both the remote address and ports parameters with "don't care" values. If macro-flow based remote policy is used, the gateway MUST specify the remote address, but MAY or MAY NOT specify the remote port(s). If micro-flow based remote policy is used, the gateway MUST specify the remote address and ports parameter.

The address of a tunnel endpoint that is not the RSIP gateway MAY be specified. If this parameter is not specified, the RSIP gateway MUST be assumed to be the tunnel endpoint.

10. Discussion

10.1. Use of Message Counters, Timeouts, and Retransmissions

Message counters are conceptually similar to sequence numbers. They are necessary to facilitate reliability when UDP is the transport protocol. Each UDP message is marked with a message counter. When such a message is transmitted, the message is stored in a "last message" buffer. For RSIP hosts, a timer is set to expire at the appropriate timeout value.

General rules:

- When an RSIP host transmits a message with a message counter value of *n*, the RSIP gateway's response will contain a message counter value of *n*.
- An RSIP host will not increment its message counter value to *n+1* until it receives a message from the RSIP gateway with a message counter value of *n*.
- An RSIP gateway begins all sessions with a message counter value of 1.
- If the message counter value reaches the maximum possible 32-bit value, it will wrap around to 1, not 0.
- If a message with a message counter value of *n* is transmitted by an RSIP host, but a timer expires before a response to that message is received, the copy of the message (from the "last message" buffer) is retransmitted.
- When an RSIP gateway receives a duplicate copy of a message with a message counter value of *n*, it transmits the contents of its "last message" buffer.
- When the RSIP gateway transmits an asynchronous RSIP message (an RSIP message for which there was no request by the RSIP host), a message counter value of 0 MUST be used. Note that only three RSIP messages can be transmitted asynchronously: `ERROR_RESPONSE`, `DE-REGISTER_RESPONSE`, and `FREE_RESPONSE`. These messages may also be transmitted in response to an RSIP host request, so their message counter values MAY be non-zero.
- If a message counter is not present in a message from an RSIP host, but is required, the RSIP gateway MUST respond with an `ERROR_RESPONSE` containing the `MESSAGE_COUNTER_REQUIRED` error.

10.2. RSIP Host and Gateway Failure Scenarios

When either the RSIP host or gateway suffers from an unrecoverable failure, such as a crash, all RSIP-related state will be lost. In this section, we describe the sequence of events that will occur in both host and gateway failures, and how the host and gateway re-synchronize.

10.2.1. Host Failure

After a host failure, the host will reboot and be unaware of any RSIP state held on its behalf at the gateway.

If the host does not immediately attempt to re-establish a session, it may receive RSIP packets on the RSIP client application port that it was using before it rebooted. If an RSIP client application is not active on this port, these packets will be responded to with ICMP port unreachable messages. If TCP is the transport protocol, it is likely that the connection will be terminated with a TCP RST. If an RSIP client is active on this port, it will not recognize the session that these packets belong to, and it SHOULD silently ignore them.

The RSIP host may also receive packets from a remote host with which it was communicating before it rebooted. These packets will be destined to the RSIP tunnel interface, which should not exist. Thus they SHOULD be silently discarded by the RSIP host's stack, or the RSIP host will transmit appropriate ICMP messages to the tunnel endpoint (e.g., the RSIP gateway). The behavior of the system with respect to sessions that were active before the reboot should be similar to that of a publically addressable non-RSIP host that reboots.

Upon rebooting, an RSIP host may attempt to establish a new RSIP session with the RSIP gateway. Upon receiving the REGISTER_REQUEST message, the RSIP gateway will be able to determine that, as far as it is concerned, the RSIP host is already registered. Thus, it will transmit an ERROR_RESPONSE with the ALREADY_REGISTERED message. Upon receipt of this message, the RSIP host will know the client ID of its old registration, and SHOULD immediately transmit a DE-REGISTER_REQUEST using this client ID. After this is accomplished, the states of the RSIP host and gateway have been synchronized, and a new RSIP session may be established.

If the RSIP host does not de-register itself from the RSIP gateway, it will eventually receive a DE-REGISTER_RESPONSE from the gateway, when the gateway times out the host's session. Since the DE-REGISTER_RESPONSE will refer to a client ID that has no meaning to

the host, the host SHOULD silently ignore such a message. At this point, the states of the RSIP host and gateway have been synchronized, and a new RSIP session may be established.

10.2.2. Gateway Failure

After a gateway failure, the gateway will reboot and be unaware of any RSIP state held by an RSIP host.

Since the gateway will not attempt to contact any of its RSIP hosts, a problem will first be detected when either an RSIP host sends an RSIP message to the gateway, an RSIP host sends tunneled data to the gateway, or data from a remote host intended for an RSIP host arrives.

In the first case, the RSIP gateway SHOULD immediately response to all messages (except for a REGISTER_REQUEST) with an ERROR_RESPONSE with a REGISTER_FIRST error. Upon receipt of such a message, an RSIP host MUST interpret the message as an indication of a loss of synchronization between itself and the RSIP gateway. The RSIP host SHOULD immediately transmit a DE-REGISTRATION_REQUEST with its old client ID (which will generate another error, but this error SHOULD be ignored by the host). At this point, the states of the RSIP host and gateway have been synchronized, and a new RSIP session may be established.

In the second case, all data that an RSIP host sends to the tunneled interface of an RSIP server will either (1) be discarded silently, (2) responded to with an ICMP Destination Unreachable message, such as "Communication Administratively Prohibited", or (3) blindly routed to the intended destination. In all of the above cases, the RSIP gateway will not have an explicit method to notify the RSIP host of the problem. To prevent a long term communications outage, small lease times of several minutes can be set by the RSIP gateway.

In the third case, the RSIP gateway SHOULD discard all incoming packets and/or respond with ICMP Port Unreachable messages.

10.3. General Gateway Policy

There is a significant amount of RSIP gateway policy that may be implemented, but is beyond the scope of this document. We expect that most of this policy will be site-specific or implementation-specific and therefore do not make any recommendations. Examples of general gateway policy include:

- How ports are allocated to RSIP hosts.
- Preferred length of lease times.

- How flow policy is applied to which hosts.
- How an RSIP gateway with multiple public IP addresses that may be leased by RSIP clients determines how to partition and/or lease these addresses.

10.4. Errors Not From the RSIP Protocol

Once an RSIP host and gateway have established a relationship and the host is assigned resources to use, error may occur due to the host's misuse of the resources or its attempting to use unassigned resources. The following error behavior is defined:

- If a host attempts to use a local address which it has not been allocated, the RSIP gateway MUST drop the associated packet(s) and send the host an ERROR_RESPONSE containing the LOCAL_ADDR_UNALLOWED error.
- If a host attempts to use a local address / port tuple which it has not been allocated, the RSIP gateway MUST drop the associated packet(s) and send the host an ERROR_RESPONSE containing the LOCAL_ADDRPORT_UNALLOWED error.
- If a host attempts to contact a remote address which has not been properly specified or otherwise approved (e.g., via an ASSIGN_RESPONSE_RSAP-IP and macro or micro based remote flow policy), the RSIP gateway MUST drop the associated packet(s) and send the host an ERROR_RESPONSE containing the REMOTE_ADDR_UNALLOWED error.
- If a host attempts to contact a remote address / port tuple which has not been properly specified or otherwise approved (e.g., via an ASSIGN_RESPONSE_RSAP-IP and micro based remote flow policy), the RSIP gateway MUST drop the associated packet(s) and send the host an ERROR_RESPONSE containing the REMOTE_ADDRPORT_UNALLOWED error.
- If a host attempts to establish or use an improper tunnel type, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the BAD_TUNNEL_TYPE error.
- If the RSIP gateway's detects a local fault which prevents its RSIP server module from continuing operation, the RSIP gateway MUST respond with an ERROR_RESPONSE containing the INTERNAL_SERVER_ERROR error.

10.5. Address and Port Requests and Allocation

Regardless of local flow policy, an RSIP host may "suggest" that it would like to use a particular local address and/or port number in a particular binding. An RSIP gateway that cannot grant such a request, because the specified resources are already in use, MUST respond with an `ERROR_RESPONSE` containing the `LOCAL_ADDR_INUSE` or `LOCAL_ADDRPORT_INUSE` values.

10.6. Local Gateways and Flow Policy Interaction

An RSIP host may initialize a publically accessible gateway (such as an FTP or HTTP gateway) by transmitting a `LISTEN_REQUEST` message to an RSIP gateway and receiving a `LISTEN_RESPONSE`. However, unless no remote flow policy is used, the gateway will have to specify the address or address and port of a single remote host that will be allowed to contact it. Obviously, such a restriction is not very useful for hosts that require their gateways to be accessible by any remote host.

This indicates that there is a conflict between flow-based policy and support for gateways. The main purpose of enforcing flow-based policy for `LISTEN_REQUEST`s is that it allows an RSIP gateway tight control over how an RSIP host uses ports and the associated accounting. For example, an RSIP host, operating under remote micro-flow based policy and using a protocol such as FTP, will have to specify the address and port that it will receive FTP data on, as well as the address and port that the gateway will transmit data from, in a `LISTEN_REQUEST`.

In general, an RSIP gateway may not allow arbitrary hosts to start public gateways because of the traffic and security concerns. Thus, we recommend that if remote micro-flow based policy is used, that an RSIP gateway only allow public gateways on RSIP hosts via administrative override.

Currently, RSIP hosts can only be identified by their local IP address or MAC address.

11. Security Considerations

RSIP, in and of itself, does not provide security. It may provide the illusion of security or privacy by hiding a private address space, but security can only be ensured by the proper use of security protocols and cryptographic techniques.

An RSIP gateway should take all measures deemed necessary to prevent its hosts from performing intentional or unintentional denial-of-service attacks by request large sets of resources.

Currently, RSIP hosts can only be identified by their local IP address or, in some cases, MAC address. It is desirable to allow RSIP messages sent between a host and gateway to be authenticated. Further discussion of such authentication can be found in [RSIP-FRAME].

Discussion of RSIP support for end-to-end IPsec can be found in [RSIP-IPSEC].

12. IANA Considerations

All of the designations below have been registered by the IANA.

- RSIP port number: 4555
- RSIP error codes (see Appendix A).
- RSIP message type codes (see Appendix B).
- RSIP tunnel types, methods, and flow policies.

RSIP parameter values are designated as follows:

- 0 Reserved
- 1-240 Assigned by IANA
- 241-255 Reserved for private use

New registrations for the above namespaces are recommended to be allocated via the Specification Required method documented in [RFC2434].

13. Acknowledgements

The authors would like to specifically thank Gabriel Montenegro, Pyda Srisuresh, Brian Carpenter, Eliot Lear, Dan Nessel, Gary Jaszewski, Naveen Rajanikantha, Sudhakar Ramakrishna, Jim March, and Rick Cobb for their input. The IETF NAT working group as a whole has been extremely helpful in the ongoing development of RSIP.

14. Appendix A: RSIP Error Numbers

This section provides descriptions for the error values in the RSIP error parameter.

All errors are grouped into the following categories:

100's: General errors.

101: UNKNOWN_ERROR. An error that cannot be identified has occurred. This error should be used when all other error messages are inappropriate.

102: USE_TCP. A host has attempted to use UDP on a server that only supports TCP.

103: FLOW_POLICY_VIOLATION: A host has not specified address or port information in enough detail for its assigned flow policy.

104: INTERNAL_SERVER_ERROR: An RSIP server application has detected an unrecoverable error within itself or the RSIP gateway.

105: MESSAGE_COUNTER_REQUIRED: An RSIP host did not use a message counter parameter in a situation in which it should have.

106: UNSUPPORTED_RSIP_VERSION: An RSIP host sent a message with a version number that is not supported by the RSIP gateway.

200's: Parameter and message errors. The gateway uses these errors when it detects that a parameter or message is malformed, as well as when it does not understand a parameter or message.

201: MISSING_PARAM. The request does not contain a required parameter.

202: DUPLICATE_PARAM. The request contains an illegal duplicate parameter.

203: EXTRA_PARAM. The request contains a parameter that it should not.

204: ILLEGAL_PARAM. The gateway does not understand a parameter type.

205: BAD_PARAM. A parameter is malformed.

- 206: ILLEGAL_MESSAGE. The gateway does not understand the message type. The message type is neither mandatory nor optional.
- 207: BAD_MESSAGE. A message is malformed and gateway parsing failed.
- 208: UNSUPPORTED_MESSAGE: The host has transmitted an optional message that the gateway does not support.
- 300's: Permission, resource, and policy errors. The gateway uses these errors when a host has attempted to do something that it is not permitted to do, or something that violated gateway policy.
- 301: REGISTER_FIRST. The RSIP host has attempted to request or use resources without registering.
- 302: ALREADY_REGISTERED. The host has attempted to register again without first de-registering.
- 303: ALREADY_UNREGISTERED. The host has attempted to de-register but it is already in the unregistered state.
- 304: REGISTRATION_DENIED. The gateway will not allow the host to register.
- 305: BAD_CLIENT_ID. The host has referred to itself with the wrong client ID.
- 306: BAD_BIND_ID. The request refers to a bind ID that is not valid for the host.
- 307: BAD_TUNNEL_TYPE. The request refers to a tunnel type that is not valid for the host.
- 308: LOCAL_ADDR_UNAVAILABLE. The gateway is currently not able to allocate ANY local address, but the host may try again later.
- 309: LOCAL_ADDRPORT_UNAVAILABLE. The gateway is currently not able to allocate ANY local IP address / port tuple of the requested magnitude (i.e., number of ports), but the host may try again later.
- 310: LOCAL_ADDR_INUSE. The gateway was not able to allocate the requested local address because it is currently used by another entity.

311: LOCAL_ADDRPORT_INUSE. The gateway was not able to allocate the requested local address / port tuple because it is currently used by another entity.

312: LOCAL_ADDR_UNALLOWED. The gateway will not let the host use the specified local IP address due to policy.

313: LOCAL_ADDRPORT_UNALLOWED. The gateway will not let the host use the specified local address / port pair due to policy.

314: REMOTE_ADDR_UNALLOWED. The gateway will not allow the host to establish a session to the specified remote address.

315: REMOTE_ADDRPORT_UNALLOWED. The gateway will not allow the host to establish a session to the specified remote address / port tuple.

400's: IPsec errors. All errors specific to RSIP / IPsec operation. See [RSIP-IPSEC].

15. Appendix B: Message Types

This section defines the values assigned to RSIP message types. We also indicate which RSIP entity, host or gateway, produces each messages, and whether it is mandatory or optional. All *_REQUEST messages are only to be implemented on hosts, while all *_RESPONSE messages are only to be implemented on gateways. RSIP implementations (both host and gateway) MUST support all mandatory messages in order to be considered "RSIP compliant".

Value	Message	Implementation	Status
1	ERROR_RESPONSE	gateway	mandatory
2	REGISTER_REQUEST	host	mandatory
3	REGISTER_RESPONSE	gateway	mandatory
4	DE-REGISTER_REQUEST	host	mandatory
5	DE-REGISTER_RESPONSE	gateway	mandatory
6	ASSIGN_REQUEST_RSA-IP	host	optional
7	ASSIGN_RESPONSE_RSA-IP	gateway	optional
8	ASSIGN_REQUEST_RSAP-IP	host	mandatory
9	ASSIGN_RESPONSE_RSAP-IP	gateway	mandatory
10	EXTEND_REQUEST	host	mandatory
11	EXTEND_RESPONSE	gateway	mandatory
12	FREE_REQUEST	host	mandatory
13	FREE_RESPONSE	gateway	mandatory
14	QUERY_REQUEST	host	optional
15	QUERY_RESPONSE	gateway	mandatory
16	LISTEN_REQUEST	host	optional
17	LISTEN_RESPONSE	gateway	optional

16. Appendix C: Example RSIP host/gateway transactions

In this appendix, we present an exemplary series of annotated transactions between an RSIP host and an RSIP gateway. All host to gateway traffic is denote by 'C --> S' and all gateway to host traffic is denoted by 'S --> C'. Parameter values are denoted inside of parentheses. Versions, message types, and overall lengths are not included in order to save space. "Don't care" values are indicated by 0's.

A ports parameter is represented by the number of ports followed by the port numbers, separated by dashes. For example, 2-1012-1013 indicates two ports, namely 1012 and 1013, while 16-10000 indicates 16 ports, namely 10000-10015, and 4-0 indicates four ports, but the sender doesn't care where they are.

IPv4 addresses are assumed.

16.1. RSAP-IP with Local Macro-flow Based Policy and No Remote Flow Policy

This example exhibits the loosest policy framework for RSAP-IP.

C --> S: REGISTER_REQUEST ()

The host attempts to register with the gateway.

S --> C: REGISTER_RESPONSE (Client ID = 1, Local Flow Policy = Macro, Remote Flow policy = None, Lease Time = 600)

The gateway responds, assigning a Client ID of 1, local macro-flow based policy and no remote flow policy. No RSIP method is indicated, so RSAP-IP is assumed. No tunnel type is indicated, so IP-IP is assumed. A lease time of 600 seconds is assigned.

C --> S: ASSIGN_REQUEST_RSAP-IP: (Client ID = 1, Address (local) = 0, Ports (local) = 4-0, Address (remote) = 0, Ports (remote) = 0, Lease Time = 3600)

The host requests an address and four ports to use with it, but doesn't care which address or ports are assigned. The host does not specify the remote address or ports either. The host suggests a lease time of 3600 seconds.

S --> C: ASSIGN_RESPONSE_RSAP-IP: (Client ID = 1, Bind ID = 1, Address (local) = 149.112.240.156, Ports (local) = 4-1234, Address (remote) = 0, Ports (remote) = 0, Lease Time = 1800, Tunnel Type = IP-IP)

The gateway responds by indicating that a bind ID of 1 has been assigned to IP address 149.112.240.156 with ports 1234-1237. Any remote host may be communicated with, using any remote port number. The lease time has been assigned to be 1800 seconds, and the tunnel type is confirmed to be IP-IP.

The host is now able to communicate with any host on the public network using these resources.

C --> S: QUERY_REQUEST: (Client ID = 1, Indicator = network, Address (network) = 10.20.60.0, Address (netmask) = 255.255.255.0)

The host asks the gateway if the network 10.20.60.0/24 is local.

S --> C: QUERY_RESPONSE: (Client ID = 1, Indicator = network, Address (network) = 10.20.60.0, Address (netmask) = 255.255.255.0)

The gateway responds indicating that the network in question is local.

C --> S: ASSIGN_REQUEST_RSAP-IP: (Client ID = 1, Address (local) = 149.112.240.156, Ports (local) = 8-1238, Address (remote) = 0, Ports (remote) = 0, Lease Time = 1800)

The host requests eight more particular ports for use with RSAP-IP with the same address. A lease of 1800 seconds is requested. IP-IP tunneling is implied by default.

S --> C: ASSIGN_RESPONSE_RSAP-IP: (Client ID = 1, Bind ID = 2, Address (local) = 149.112.240.156, Ports (local) = 8-1305, Address (remote) = 0, Ports (remote) = 0, Lease Time = 1800)

The gateway grants the request with the same address, but with a different set of ports. IP-IP tunneling is implied by default.

C --> S: FREE_REQUEST (Client ID = 1, Bind ID = 1)

The host frees bind ID 1; i.e., ports 1234-1237 from IP address 149.112.240.156. Note that the address itself is still assigned to the host because the host is still assigned ports 1305-1314.

S --> C: FREE_RESPONSE (Client ID = 1, Bind ID = 1)

The gateway acknowledges that Bind ID 1 has been freed.

C --> S: EXTEND_REQUEST (Client ID = 1, Bind ID = 2, Lease Time = 1800)

The host request that the lease on bind ID 1 be extended for 1800 seconds.

S --> C: EXTEND_RESPONSE (Client ID = 1, Bind ID = 2, Lease Time = 1800)

The gateway confirms the request.

S --> C: FREE_RESPONSE (Client ID = 1, Bind ID = 2)

The gateway forces the host to free the resources of bind ID 2.

C --> S: DE-REGISTER_REQUEST (Client ID = 1)

The host de-registers with the sever.

S --> C: DE-REGISTER_RESPONSE (Client ID = 1)

The gateway acknowledges that the host has de-registered.

16.2. RSAP-IP with Local Micro-flow Based Policy and Remote Micro-flow Based Policy

This example exhibits the strictest policy framework for RSAP-IP.

C --> S: REGISTER_REQUEST ()

The host attempts to register with the gateway.

S --> C: REGISTER_RESPONSE (Client ID = 5, Local Flow Policy = Micro, Remote Flow policy = Micro, RSIP Method = RSAP-IP, RSIP Method = RSA-IP, Tunnel Type = IP-IP, Tunnel Type = GRE, Lease Time = 600)

The gateway responds, assigning a Client ID of 5, local micro-flow based policy and remote micro-flow based policy. Both RSAP-IP and RSA-IP are supported. Both IP-IP and GRE tunnel types are supported. A lease time of 600 seconds is assigned.

C --> S: ASSIGN_REQUEST_RSAP-IP: (Client ID = 5, Address (local) = 0, Ports (local) = 0, Address (remote) = 38.196.73.6, Ports (remote) = 21, Lease Time = 600, Tunnel Type = IP-IP)

The host requests a local address and a port assignment to use with it. The host indicates that it wants to contact host 38.196.73.6 at port 21 (FTP control). The host requests a lease time of 600 seconds and a tunnel type of IP-IP.

S --> C: ASSIGN_RESPONSE_RSAP-IP: (Client ID = 5, Bind ID = 1, Address (local) = 149.112.240.156, Ports (local) = 2049, Address (remote) = 38.196.73.6, Ports (remote) = 21, Lease Time = 600, Tunnel Type = IP-IP)

The gateway responds by indicating that a bind ID of 1 has been assigned to IP address 149.112.240.156 with port 2049. Only host 38.196.73.6 at port 21 may be contacted. The lease time has been assigned to be 600 seconds, and the tunnel type is confirmed to be IP-IP.

C --> S: LISTEN_REQUEST: (Client ID = 5, Address (local) = 149.112.240.156, Ports (local) = 2050, Address (remote) = 38.196.73.6, Ports (remote) = 20)

The host requests a listen port 2050 at the same address that it has been assigned. Only host 38.196.73.6 from ports 20 (FTP data) will be able to contact it.

S --> C: LISTEN_RESPONSE: (Client ID = 5, Address (local) = 149.112.240.156, Ports (local) = 2050, Address (remote) = 38.196.73.6, Ports (remote) = 20, Lease Time = 600, Tunnel Type = IP-IP)

The gateway confirms the request and assigns a lease time of 600 seconds and a tunnel type of IP-IP.

C --> S: DE-REGISTER_REQUEST (Client ID = 5)

The host de-registers with the sever.

S --> C: DE-REGISTER_RESPONSE (Client ID = 5)

The gateway acknowledges that the host has de-registered. All of the host's bindings have been implicitly revoked.

16.3. RSA-IP with Local Macro-flow Based Policy and Remote Macro-flow based Policy

This example exhibits a medium level of control for RSA-IP.

C --> S: REGISTER_REQUEST ()

The host attempts to register with the gateway.

S --> C: REGISTER_RESPONSE (Client ID = 3, Local Flow Policy = Macro, Remote Flow policy = Macro, RSIP Method = RSAP-IP, RSIP Method = RSA-IP, Tunnel Type = IP-IP, Tunnel Type = L2TP, Lease Time = 600)

The gateway responds, assigning a Client ID of 3, local macro-flow based policy and remote macro-flow based policy. Both RSAP-IP and RSA-IP are supported. Both IP-IP and L2TP tunnel types are supported. A lease time of 600 seconds is assigned.

C --> S: ASSIGN_REQUEST_RSA-IP: (Client ID = 3, Address (local) = 0, Address (remote) = www.foo.com, Ports (remote) = 0, Lease Time = 3600, Tunnel Type = IP-IP)

The host requests a local address and indicates that it wants to contact host www.foo.com.

S --> C: ERROR_RESPONSE: (Error = REMOTE_ADDR_UNALLOWED, Client ID = 3)

The gateway indicates that the host is not permitted to establish communication with www.foo.com.

C --> S: ASSIGN_REQUEST_RSA-IP: (Client ID = 3, Address (local) = 0, Address (remote) = www.bar.com, Ports (remote) = 0, Lease Time = 3600, Tunnel Type = IP-IP)

The host requests a local address and indicates that it wants to contact host www.bar.com.

S --> C: ASSIGN_RESPONSE_RSA-IP: (Client ID = 3, Bind ID = 1, Address (local) = 149.112.240.17, Address (remote) = www.bar.com, Ports (remote) = 0, Lease Time = 3600, Tunnel Type = IP-IP)

The gateway responds by granting local IP address 149.112.240.17 to the host, and permitting it to communicate with www.bar.com, at any port. Requested lease time and tunnel type are also granted.

C --> S: DE-REGISTER_REQUEST (Client ID = 3)

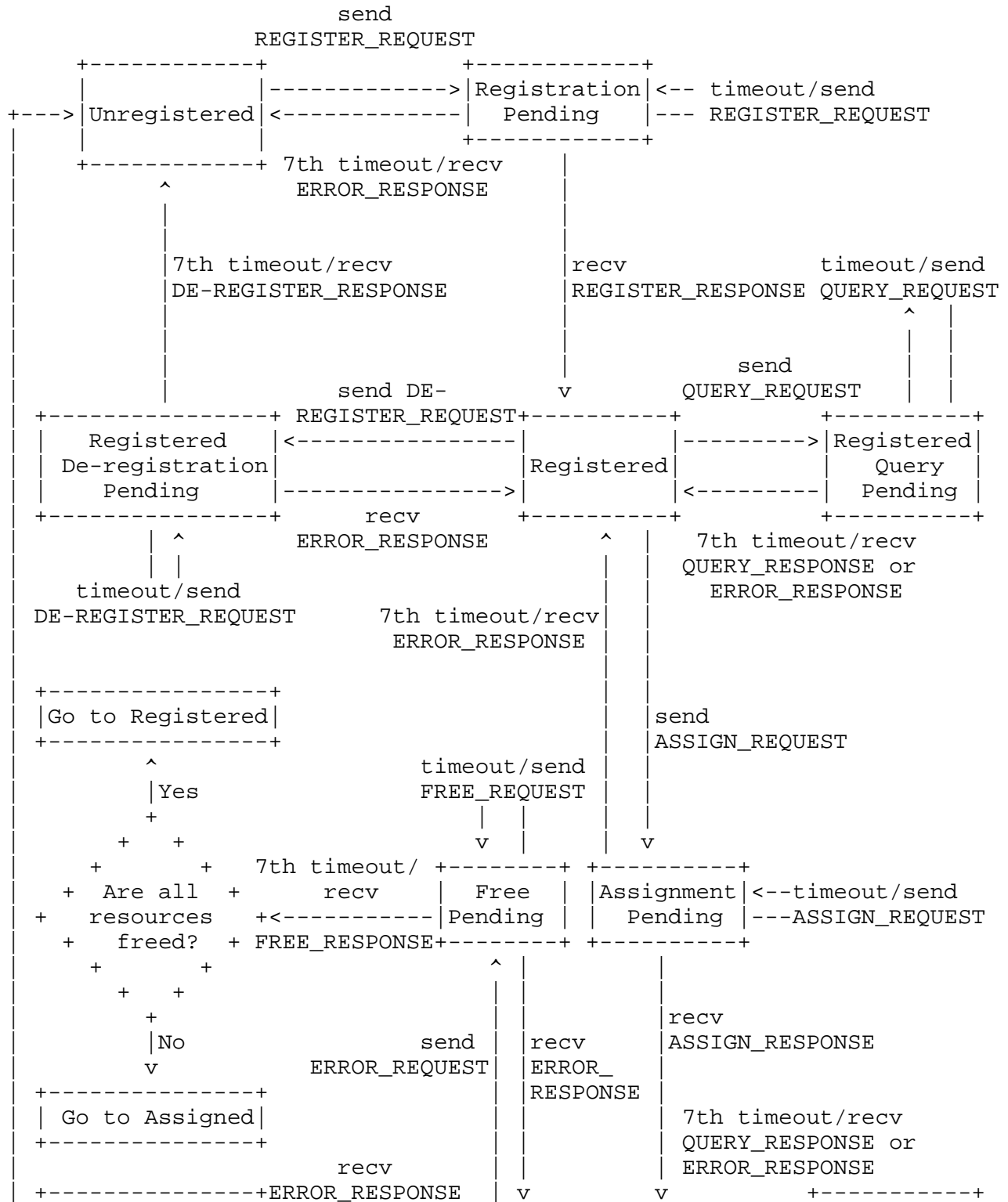
The host de-registers with the sever.

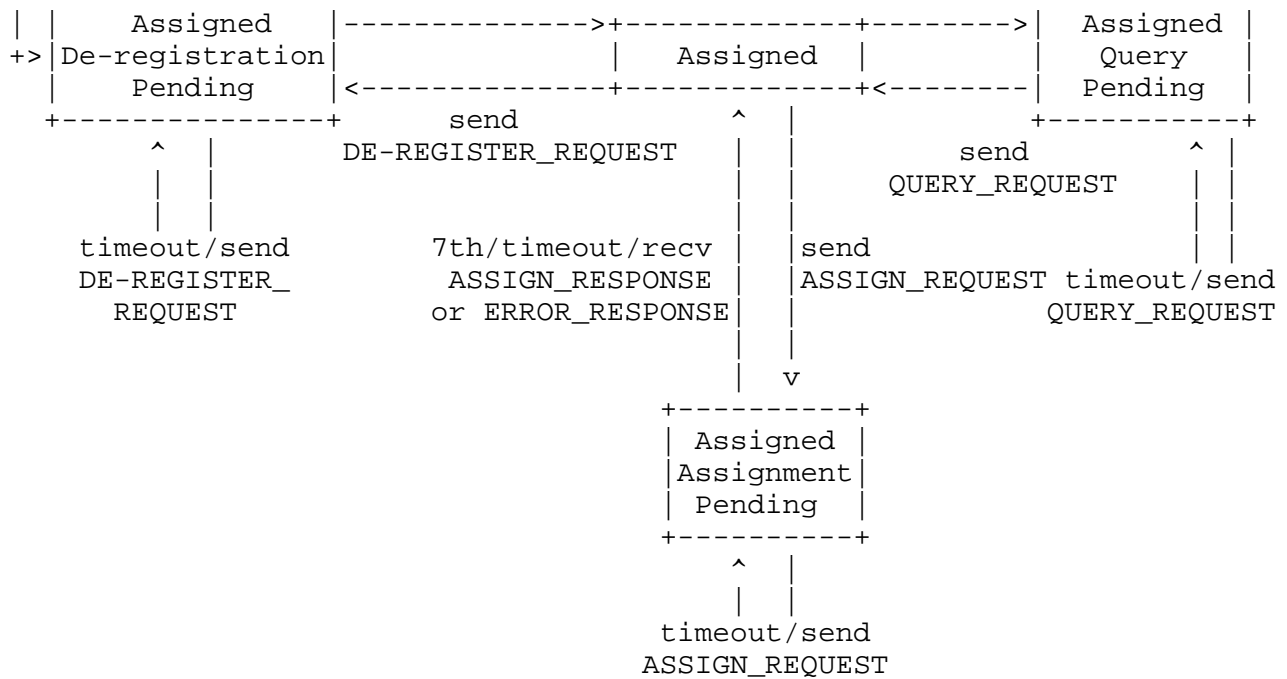
S --> C: DE-REGISTER_RESPONSE (Client ID = 3)

The gateway acknowledges that the host has de-registered. All of the host's bindings have been implicitly revoked.

17. Appendix D: Example RSIP host state diagram

This appendix provides an exemplary diagram of RSIP host state. The host begins in the unregistered state. We assume that for UDP, if a message is lost, the host will timeout and retransmit another copy of it. We recommend a 7-fold binary exponential backoff timer for retransmissions, with the first timeout occurring after 12.5 ms. This diagram does not include transitions for the LISTEN_REQUEST message.





18. References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J. and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to indicate requirement levels", BCP 14, RFC 2119, March 1997.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RSIP-FRAME] Borella, M. Lo, J., Grabelsky, D. and G. Montenegro, "Realm Specific IP: Framework", RFC 3102, October 2001.
- [RSIP-IPSEC] Montenegro, G. and M. Borella, "RSIP Support for End-to-end IPSEC", RFC 3104, October 2001.

19. Authors' Addresses

Michael Borella
CommWorks
3800 Golf Rd.
Rolling Meadows IL 60008

Phone: (847) 262-3083
EMail: mike_borella@commworks.com

David Grabelsky
CommWorks
3800 Golf Rd.
Rolling Meadows IL 60008

Phone: (847) 222-2483
EMail: david_grabelsky@commworks.com

Jeffrey Lo
Candlestick Networks, Inc
70 Las Colinas Lane,
San Jose, CA 95119

Phone: (408) 284 4132
EMail: yidarlo@yahoo.com

Kunihiro Taniguchi
NEC USA
C&C Research Labs.
110 Rio Robles
San Jose, CA 95134

Phone: (408) 943-3031
EMail: taniguti@ccrl.sj.nec.com

20. Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

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

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

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

Acknowledgement

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

