Realm Specific IP: Framework
<draft-ietf-nat-rsip-framework-03.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

Abstract

This document examines the general framework of Realm Specific IP (RSIP). RSIP is intended as an alternative to NAT in which the end-to-end integrity of packets is maintained. We focus on implementation issues, deployment scenarios, and interaction with other layer-three protocols.

1. Introduction
Network Address Translation (NAT) has become a popular mechanism of enabling the separation of addressing spaces. A NAT router must examine and change the network layer, and possibly the transport layer, header of each packet crossing the addressing domains that the NAT router is connecting. This causes the mechanism of NAT to violate the end-to-end nature of the Internet connectivity, and disrupts protocols requiring or enforcing end-to-end integrity of packets.

While NAT does not require a host to be aware of its presence, it requires the presence of a proxy module, the application layer gateway (ALG), within the NAT router for each application that embeds addressing information, IP address or port content, within the packet payload (e.g., FTP). RSIP (Realm Specific IP) provides an alternative to remedy these limitations.

RSIP is based on the concept of granting a host from one addressing realm a presence in another addressing realm by allowing it to use resources (e.g., addresses and other routing parameters) from the second addressing realm. An RSIP server/gateway replaces the NAT router, and RSIP-aware hosts on the private network are referred to as RSIP clients. RSIP requires ability of the RSIP server to grant such resources to RSIP clients. ALGs are not required on the RSIP server for communications between an RSIP client and a host on a different addressing realm.

It is important to note that RSIP is not a replacement for IPv6. We fully advocate the adoption and deployment of IPv6. RSIP has been designed to restore some of the end-to-end transparency that NAT has taken away from the Internet, and it may smooth the IPv6 transition process.

1.1. Document Scope

This document provides a framework for RSIP by focusing on three particular areas:

- Implementation issues that are not specific to the RSIP protocol defined in [RSIP-PROTO].

- Likely initial deployment scenarios.

- Interaction with other layer-three protocols.

The interaction sections will be at an overview level. Detailed modifications that would need to be made to RSIP and/or the interacting protocol are left for separate documents to discuss in detail.
Beyond the scope of this document is discussion of RSIP in large, multiple-gateway networks, or in environments where RSIP state would need to be distributed and maintained across multiple redundant entities.

Discussion of RSIP solutions that do not use some form of tunnel between the RSIP client and RSIP server are also not considered in this document.

1.2. 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 Server

A router situated on the boundary between two addressing realms and owns one or more IP addresses. An RSIP server is responsible for parameter management and assignment from one realm to RSIP clients in the other realm. An RSIP server may act as a normal NAT router for hosts within the a realm that are not RSIP enabled.

RSIP Client

A host within an addressing realm that uses RSIP to acquire addressing parameters from another addressing realm via an RSIP server.

RSA-IP: Realm Specific Address IP

An RSIP method in which each RSIP client is allocated a unique IP address from the public realm.

RSAP-IP: Realm Specific Address and Port IP

An RSIP method in which each RSIP client is allocated an IP address (possibly shared with other RSIP clients) and some
number of per-address unique ports from the public realm.

RSIP-enabled Mobile Node (RMN)

A host that uses RSIP for connectivity to the public network, and also uses Mobile IP for roaming support.

RSIP Home Network (RHN)

A network on which a number of hosts use RSIP to share one or more public IP addresses.

RSIP Home Agent (RHA)

A router, running an RSIP server, that manages Mobile IP connectivity for RSIP-enabled mobile nodes belonging to an RSIP home network.

RSIP Foreign Network (RFN)

A network which can support RSIP-enabled mobile nodes as they roam.

RSIP Foreign Agent (RFA)

A router that manages Mobile IP connectivity for roaming RSIP-enabled mobile nodes. This router may or may not use RSIP on its local network.

Demultiplexing Fields

Any set of packet header or payload fields that an RSIP server uses to route an incoming packet to an RSIP client.

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

1.3. Specification of Requirements

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

2. Architecture

In a typical scenario where RSIP is deployed, there are some number of hosts within one addressing realm connected to another addressing realm by an RSIP server. This model is diagrammatically represented
Hosts X and Y belong to different addressing realms A and B, respectively, and N is an RSIP server (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.

As is often the case, the hosts within address space A are likely to use private addresses while the RSIP server is multi-homed with one or more private addresses from address space A in addition to its public addresses from address space B. Thus, we typically refer to the realm in which the RSIP client resides as "private" and the realm from which the RSIP client borrow addressing parameters as the "public" realm. However, these realms may both be public or private - our notation is for convenience.

Host X, wishing 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 (e.g., addresses and other routing parameters of address space B) from the RSIP server. Upon assignment of these parameters, the RSIP server creates a mapping, referred as a "bind", of X’s addressing information and the assigned resources. This binding enables the RSIP server to correctly demultiplex and forward inbound traffic generated by Y for X. If permitted by the RSIP server, X may create multiple such bindings on the same RSIP server, or across several RSIP servers. A lease time SHOULD be associated with each bind.

Using the public parameters assigned by the RSIP server, RSIP clients tunnel data packets across address space A to the RSIP server. The RSIP server 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 server maintains a mapping of the assigned public parameters as demultiplexing fields for uniquely mapping them to RSIP client private addresses. When a packet from the public realm arrives at the RSIP server and it matches a given set of demultiplexing fields, then the RSIP server will tunnel it to the appropriate RSIP client. The tunnel headers of outbound packets from X to Y, given that X has been assigned Nb, are as follows:
There are two basic flavors of RSIP: RSA-IP and RSAP-IP. RSIP clients and servers MAY support RSA-IP, RSAP-IP, or both.

When using RSA-IP, an RSIP server maintains a pool of IP addresses to be leased by RSIP clients. Upon client request, the RSIP server allocates an IP address to the client. Once an address is allocated to a particular client, only that client may use the address until the address is returned to the pool. Clients MAY NOT use addresses that have not been specifically assigned to them. The clients may use any TCP/UDP port in combination with their assigned address. Clients may also run server applications at any port and these applications will be available to the public network without assistance from the RSIP server. A client MAY lease more than one address from the same or different RSIP servers. The demultiplexing fields of an RSA-IP session MUST include the IP address leased to the client.

When using RSAP-IP, an RSIP server maintains a pool of IP addresses as well as pools of port numbers per address. RSIP clients lease an IP address and one or more ports to use with it. Once an address / port tuple has been allocated to a particular client, only that client may use the tuple until it is returned to the pool(s). Clients MAY NOT use address / port combinations that have not been specifically assigned to them. Clients may run server applications bound to an allocated tuple, but their applications will not be available to the public network unless the RSIP server has agreed to route all traffic destined to the tuple to the client. A client MAY lease more than one tuple from the same or different RSIP servers. The demultiplexing fields of an RSAP-IP session MUST include the tuple(s) leased to the client.

3. Implementation Considerations

RSIP can be accomplished by any one of a wide range of implementation schemes. For example, it can be built into an existing configuration protocol such as DHCP or SOCKS, or it can exist as a separate protocol. This section discusses implementation issues of RSIP in general, regardless of how the RSIP mechanism is implemented.

Note that on a client, RSIP is associated with a TCP/IP stack implementation. Modifications to IP tunneling and routing code, as well as driver interfaces may need to be made to support RSA-IP. Support for RSAP-IP requires modifications to ephemeral port selection code as well. If a host has multiple TCP/IP stacks or
TCP/IP stacks and other communication stacks, RSIP will only operate on the packets / sessions that are associated with the TCP/IP stack(s) that use RSIP. RSIP is not application specific, and if it is implemented in a stack, it will operate transparently to all applications that use the stack.

3.1. Client and Server Requirements

An RSIP client must be able to maintain one or more virtual interfaces for the IP address(es) that it leases from an RSIP server. The client must also support tunneling and be able to serve as an end-point for one or more tunnels to RSIP servers. An RSIP client MUST NOT respond to ARPs for a public realm address that it leases.

An RSIP client supporting RSAP-IP MUST be able to maintain a set of one or more ports assigned by an RSIP server from which choose ephemeral source ports. If the client’s pool does not have any free ports and the client needs to open a new communication session with a public host, it MUST be able to dynamically request one or more additional ports via its RSIP mechanism.

An RSIP server is a multi-homed host that routes packets between two or more realms. Often, an RSIP server is a boundary router between two or more administrative domains. It must also support tunneling and be able to serve as an end-point for tunnels to RSIP clients. The RSIP server MAY be a policy enforcement point, which in turn may require it to perform firewall and packet filtering duties in addition to RSIP. The RSIP server must reassemble all incoming packet fragments from the public network in order to be able to route and tunnel them to the proper client. As is necessary for all fragment reassembly, an RSIP server must timeout fragments that are never fully reassembled.

An RSIP server MAY include NAT functionality so that clients on the private network that are not RSIP-enabled can still communicate with the public network. An RSIP server must manage all resources that are assigned to RSIP clients. This management MAY be done according to local policy.

3.2. Processing of Demultiplexing Fields

Each active RSIP client must have a unique set of demultiplexing fields assigned to it so that an RSIP server can route incoming packets appropriately. Depending on the type of mapping used by the RSIP server, demultiplexing fields have been defined to be one or more of the following:
- destination IP address
- IP protocol
- destination TCP or UDP port
- IPSEC SPI present in ESP or AH header (see [RSIP-IPSEC])
- ISAKMP initiator cookie present in an IKE payload (see [RSIP-IPSEC])
- others

Demultiplexing of incoming traffic can be based on a decision tree. The process begins with the examination of the IP header of the incoming packet, and proceeds to subsequent headers and then the payload.

- In the case where a public IP address is assigned for each client, a unique public IP address is mapped to each RSIP client.

- If the same IP address is used for more than one RSIP client, then subsequent headers must have at least one field that will be assigned a unique value per client so that it is usable as a demultiplexing field. The IP protocol field SHOULD be used to determine what in the subsequent headers these demultiplexing fields ought to be.

- If the subsequent header is TCP or UDP, then destination port number can be used. However, if the TCP/UDP port number is the same for more than one RSIP client, the payload section of the packet must contain a demultiplexing field that is guaranteed to be different for each RSIP client. Typically this requires negotiation of said fields between the RSIP client and server so that the RSIP server can guarantee that the fields are unique per-client.

- If the subsequent header is anything other than TCP or UDP, there must exist other fields within the IP payload usable as demultiplexing fields. In other words, these fields must be able to be set such that they are guaranteed to be unique per-client. Typically this requires negotiation of said fields between the RSIP client and server so that the RSIP server can guarantee that the fields are unique per-client.

It is desirable for all demultiplexing fields to occur in well-known fixed locations so that an RSIP server can mask out and
examine the appropriate fields on incoming packets. Demultiplexing fields that are encrypted MUST NOT be used for routing.

3.3. RSIP Protocol Requirements and Recommendations

RSIP servers and clients must be able to negotiate IP addresses when using RSA-IP, IP address / port tuples when using RSAP-IP, and possibly other demultiplexing fields for use in other modes. In this section we discuss the requirements and implementation issues of an RSIP negotiation protocol.

For each required demultiplexing field, an RSIP protocol MUST, at the very least, allow for:

- RSIP clients to request assignments of demultiplexing fields
- RSIP servers to assign demultiplexing fields with an associated lease time
- RSIP servers to reclaim assigned demultiplexing fields

Additionally, it is desirable, though not mandatory, for an RSIP protocol to negotiate an RSIP method (RSA-IP or RSAP-IP) and the type of tunnel to be used across the private network. The protocol SHOULD be extensible and facilitate vendor-specific extensions.

If an RSIP negotiation protocol is implemented at the application layer, a choice of transport protocol must be made. RSIP clients and servers may communicate via TCP or UDP. TCP support is required in all RSIP servers, while UDP support is optional. In RSIP clients, TCP, UDP, or both may be supported. However, once an RSIP client and server have begun communicating using either TCP or UDP, they MAY NOT switch to the other transport protocol. For RSIP implementations and deployments considered in this document, TCP is the recommended transport protocol, because TCP is known to be robust across a wide range of physical media types and traffic loads.

It is recommended that all communication between an RSIP client and server be authenticated. Authentication, in the form of a message hash appended to the end of each RSIP protocol packet, can serve to authenticate the RSIP client and server to one another, provide message integrity, and (with an anti-replay counter) avoid replay attacks. In order for authentication to be supported, each RSIP client and the RSIP server must either share a secret key (distributed, for example, by Kerberos) or have a private/public
key pair. In the latter case, an entity’s public key can be computed over each message and a hash function applied to the result to form the message hash.

3.4. TCP TIME_WAIT at Public Peers

When a TCP server disconnects a socket, it enters the TCP TIME_WAIT state for a period of time. While it is in this state it will refuse to accept new connections using the same socket (i.e., the same source address/port and destination address/port). Consider the case in which an RSIP client (using RSAP-IP) is leased an address/port tuple and uses this tuple to contact a public address/port tuple. Suppose that the client terminates the session with the public tuple and immediately returns its leased tuple to the RSIP server. If the RSIP server immediately allocates this tuple to another RSIP client (or to the same client), and this second client uses the tuple to contact the same public tuple while the socket is still in the TIME_WAIT phase, then the client’s connection may be rejected by the public host. In order to mitigate this problem, it is recommended that RSIP servers hold recently deallocated tuples for at least two minutes, which is the greatest duration of TIME_WAIT that is commonly implemented [STEV94]. In situations where port space is scarce, the RSIP server MAY choose to allocate ports in a FIFO fashion from the pool of recently deallocated ports.

3.5. ICMP Handling

Like NAT, RSIP servers running RSAP-IP are required to remember recent ICMP packets for which responses cannot be demultiplexed by port number (i.e., echo request packets). ICMP request packets originating on the private network will typically consist of echo request, timestamp request and address mask request. These packets and their responses can be identified by the tuple of source IP address, ICMP identifier, ICMP sequence number, and destination IP address. An RSIP client sending an ICMP request packet tunnels it to the RSIP server, just as it does TCP and UDP packets. The RSIP server must use this tuple to map incoming ICMP responses to the private address of the appropriate RSIP client. Once it has done so, it will tunnel the ICMP response to the client. Note that it is possible for two RSIP clients to use the same values for the tuples listed above, and thus create an ambiguity. However, this occurrence is likely to be quite rare, and is not addressed further in this draft.

Incoming ICMP error response messages can be forwarded to the appropriate RSIP client by examining the IP header and port numbers embedding within the packet. In the case of RSA-IP, only
the source IP address is necessary to determine the RSIP client. In the case of RSAP-IP, the source IP address and source port number is necessary to determine the RSIP client.

Occasionally, an RSIP client will have to send an ICMP response (e.g., port unreachable). These responses are tunneled to the RSIP server, as is done for TCP and UDP packets. All ICMP requests (e.g., echo request) arriving at the RSIP server MUST be processed by the RSIP server and MUST NOT be forwarded to an RSIP client.

3.6. MTU Limitation to Prevent Fragmentation and ID Collision

RSIP clients MUST limit their MTU so that packets transmitted by an RSIP server are not fragmented. If two or more RSIP clients on the same private network transmit outbound packets that get fragmented to the same public server, the public server may experience a reassembly ambiguity if the IP header ID fields of these packets are identical.

For TCP packets, this is not an issue if path MTU discovery works properly. For UDP packets, an artificially small MTU, such as 512 bytes, is required.

3.7. Servers on RSAP-IP Clients

RSAP-IP clients are limited by the same constraints as NAT with respect to hosting servers that use a well-known port. Since destination port numbers are used as routing information to uniquely identify an RSAP-IP client, typically no two RSAP-IP clients sharing the same public IP address can simultaneously operate publically-available servers on the same port. For protocols that operate on well-known ports, this implies that only one public server per RSAP-IP IP address / port tuple is used simultaneously. However, more than one server per RSAP-IP IP address / port tuple may be used simultaneously if and only if there is a demultiplexing field within the payload of all packets that will uniquely determine the identity of the RSAP-IP client, and this field is known by the RSIP server.

In order for an RSAP-IP client to operate a publically-available server, the client must inform the RSIP server that it wishes to receive all traffic destined to that port number, per its IP address. Such a request MUST be denied if the port in question is already in use by another client.

3.8. Determining Locality of Destinations from an RSIP Client
In general, an RSIP client must know, for a particular IP address, whether it should address the packet for local delivery on the private network, or if it has to use an RSIP interface to tunnel to an RSIP server (assuming that it has such an interface available).

If the RSIP clients are all on a single subnet, one hop from an RSIP server, then examination of the local network and subnet mask will provide the appropriate information. However, this is not always the case.

An alternative that will work in general for statically addressed private networks is to store a list of the network and subnet masks of every private subnet at the RSIP server. RSIP clients may query the server with a particular target IP address, or for the entire list.

If the subnets on the local side of the network are changing more rapidly than the lifetime of a typical RSIP session, the RSIP client may have to query the location of every destination that it tries to communicate with.

If an RSIP client transmits a packet addressed to a public host without using RSIP, then the RSIP server will apply NAT to the packet (if it supports NAT) or it may discard the packet and respond with an appropriate ICMP message.

3.9. Implementing RSIP Client Deallocation

An RSIP client MAY free resources that it has determined it no longer requires. For example, on an RSAP-IP subnet with a limited number of public IP addresses, port numbers may become scarce. Thus, if RSIP clients are able to dynamically deallocate ports that they no longer need, more clients can be supported.

However, this functionality may require significant modifications to a vanilla TCP/IP stack in order to implement properly. The RSIP client must be able to determine which TCP or UDP sessions are using RSIP resources. If those resources are unused for a period of time, then the RSIP client may deallocate them. When an open socket’s resources are deallocated, it will cause some associated applications to fail. An analogous case would be TCP and UDP sessions that must terminate when an interface that they are using loses connectivity.

On the other hand, this issue can be considered a resource allocation problem. It is not recommended that a large number (hundreds) of hosts share the same IP address, for performance
purposes. Even if, say, 100 hosts each are allocated 100 ports, the total number of ports in use by RSIP would be still less than one-sixth the total port space for an IP address. If more hosts or more ports are needed, more IP addresses should be used. Thus, it is reasonable, that in many cases, RSIP clients will not have to deallocate ports for the lifetime of their activity.

Since RSIP demultiplexing fields are leased to clients, an appropriately chosen lease time can alleviate some port space scarcity issues.

3.10. Interaction with DNS

An RSIP-enabled network has three uses for DNS: (1) public DNS services to map its static public IP addresses (i.e., the public address of the RSIP server) and for lookups of public hosts, (2) private DNS services for use only on the private network, and (3) dynamic DNS services for RSIP clients.

With respect to (1), public DNS information MUST be propagated onto the private network. With respect to (2), private DNS information MUST NOT be propagated into the public network.

With respect to (3), an RSIP-enabled network MAY allow for RSIP clients with FQDNs to have their A and PTR records updated in the public DNS. These updates are based on address assignment facilitated by RSIP, and should be performed in a fashion similar to DHCP updates to dynamic DNS [DHCP-DNS]. In particular, RSIP clients should be allowed to update their A records but not PTR records, while RSIP servers can update both. In order for the RSIP server to update DNS records on behalf on an RSIP client, the client must provide the server with its FQDN.

Note that when using RSA-IP, the interaction with DNS is completely analogous to that of DHCP because the RSIP client "owns" an IP address for a period of time. In the case of RSAP-IP, the claim that an RSIP client has to an address is only with respect to the port(s) that it has leased along with an address. Thus, two or more RSIP clients’ FQDNs may map to the same IP address. However, a public client may expect that all of the applications running at a particular address are owned by the same logical host, which would not be the case. It is recommended that RSAP-IP and dynamic DNS be integrated with some caution, if at all.

3.11. Locating RSIP Servers

When an RSIP client initializes, it requires (among other things)
two critical pieces of information. One is a local (private) IP address to use as its own, and the other is the private IP address of an RSIP server. This information can be statically configured or dynamically assigned.

In the dynamic case, the client’s private address is typically supplied by DHCP. A DHCP option has been proposed to provide the IP address of an RSIP server in DHCP OFFER messages (the Next Server option) [DHC-NS]. Thus, the client’s startup procedure would be as follows: (1) perform DHCP, (2) if the Next Server option is present in the DHCP OFFER, record the IP address therein as the RSIP server.

4. Deployment

When RSIP is deployed in certain scenarios, the network characteristics of these scenarios will determine the scope of the RSIP solution, and therefore impact the requirements of RSIP. In this section, we examine deployment scenarios, and the impact that RSIP may have on existing networks.

4.1. Possible Deployment Scenarios

In this section we discuss a number of potential RSIP deployment scenarios. The selection below are not comprehensive and other scenarios may emerge.

4.1.1. Small / Medium Enterprise

Up to several hundred hosts will reside behind an RSIP-enabled router. It is likely that there will be only one gateway to the public network and therefore only one RSIP server. This RSIP server may control only one, or perhaps several, public IP addresses. The RSIP server may also perform firewall functions, as well as routing inbound traffic to particular destination ports on to a small number of dedicated servers on the private network.

4.1.2. Residential Networks

This category includes both networking within just one residence, as well as within multiple-dwelling units. At most several hundred hosts will share the server’s resources. In particular, many of these devices may be thin clients or so-called "network appliances" and therefore not require access to the public Internet frequently. The RSIP server is likely to be implemented as part of a residential firewall, and it may be called upon to route traffic to particular destination ports on...
to a small number of dedicated servers on the private network. It is likely that only one gateway to the public network will be present and that the gateway’s RSIP server will control only one IP address. Support for secure end-to-end VPN access to corporate intranets will be important.

4.1.3. Hospitality Networks

A hospitality network is a general type of "hosting" network that a traveler will use for a short period of time (a few minutes or a few hours). Examples scenarios include hotels, conference centers and airports and train stations. At most several hundred hosts will share the server’s resources. The RSIP server may be implemented as part of a firewall, and it will probably not be used to route traffic to particular destination ports on to dedicated servers on the private network. It is likely that only one gateway to the public network will be present and that this gateway’s RSIP server will control only one IP address. Support for secure end-to-end VPN access to corporate intranets will be important.

4.1.4. Dialup Remote Access

RSIP servers may be placed in dialup remote access concentrators in order to multiplex IP addresses across dialup users. At most several hundred hosts will share the server’s resources. The RSIP server may or may not be implemented as part of a firewall, and it will probably not be used to route traffic to particular destination ports on to dedicated servers on the private network. Only one gateway to the public network will be present (the remote access concentrator itself) and that this gateway’s RSIP server will control a small number of IP addresses. Support for secure end-to-end VPN access to corporate intranets will be important.

4.1.5. Wireless Remote Access Networks

Wireless remote access will become very prevalent as more PDA and IP / cellular devices are deployed. In these scenarios, hosts may be changing physical location very rapidly - therefore Mobile IP will play a role. Hosts typically will register with an RSIP server for a short period of time. At most several hundred hosts will share the server’s resources. The RSIP server may be implemented as part of a firewall, and it will probably not be used to route traffic to particular destination ports on to dedicated servers on the private network. It is likely that only one gateway to the public network will be present and that this gateway’s RSIP server
will control a small number of IP addresses. Support for secure end-to-end VPN access to corporate intranets will be important.

4.2. Cascaded RSIP and NAT

It is possible for RSIP to allow for cascading of RSIP servers as well as cascading of RSIP servers with NAT boxes. For example, consider an ISP that uses RSIP for address sharing amongst its customers. It might assign resources (e.g., IP addresses and ports) to a particular customer. This customer may use RSIP to further subdivide the port ranges and address(es) amongst individual end hosts. No matter how many levels of RSIP assignment exists, RSIP MUST only assign public IP addresses.

Note that some of the architectures discussed below may not be useful or desirable. The goal of this section is to explore the interactions between NAT and RSIP as RSIP is incrementally deployed on systems that already support NAT.

4.2.1. RSIP Behind RSIP

A reference architecture is depicted below.

```
+-----------+          +-----------+
|           |          |           |
|   RSIP    |          |   RSIP    |
|  server   +---- 10.0.0.0/8 |  server   +---- 10.0.0.0/8
|     B     |          |     C     |
|           |          |           |
+-----+-----+          +-----+-----+
| 10.0.1.0/24           | 10.0.2.0/24           |
| (149.112.240.0/25)    | (149.112.240.128/25) |
| 149.112.240.0/24      | 149.112.240.0/24      |
| RSIP                  | RSIP                  |
| A ++                  | A ++                  |
+-----------------------+-----------------------+
| 10.0.1.0/24           | 10.0.2.0/24           |
| (149.112.240.0/25)    | (149.112.240.128/25) |
```

RSIP-server A is in charge of the IP addresses of subnet 149.112.240.0/24. It distributes these addresses to RSIP-clients and RSIP-servers. In the given configuration, it distributes addresses 149.112.240.0 - 149.112.240.127 to RSIP-server B, and addresses 149.112.240.128 - 149.112.240.254 to RSIP-server C. Note that the subnet broadcast address, 149.112.240.255, must remain unclaimed, so that broadcast packets can be distributed to arbitrary hosts behind RSIP-server A. Also, the subnets between RSIP-server A and RSIP-servers B and C will use private addresses.

Due to the tree-like fashion in which addresses will be cascaded, we will refer to RSIP-servers A as the ‘parent’ of RSIP-servers B and C, and RSIP-servers B and C as ‘children’ of RSIP-servers A. An arbitrary number of levels of children may exist under a parent RSIP-server.

A parent RSIP-server will not necessarily be aware that the address(es) and port blocks that it distributes to a child RSIP-server will be further distributed. Thus, the RSIP-clients MUST tunnel their outgoing packets to the nearest RSIP-server. This server will then verify that the sending host has used the proper address and port block, and then tunnel the packet on to its parent RSIP-server.

For example, in the context of the diagram above, host 10.0.0.1, behind RSIP-server C will use its assigned external IP address (say, 149.112.240.130) and tunnel its packets over the 10.0.0.0/8 subnet to RSIP-server C. RSIP-server C strips off the outer IP header. After verifying that the source public IP address and source port number is valid, RSIP-server C will tunnel the packets over the 10.0.2.0/8 subnet to RSIP-server A. RSIP-server A strips off the outer IP header. After verifying that the source public IP address and source port number is valid, RSIP-server A transmits the packet on the public network.

While it may be more efficient in terms of computation to have a RSIP-client tunnel directly to the overall parent of an RSIP-server tree, this would introduce significant state and administrative difficulties.

A RSIP-server that is a child MUST take into consideration the parameter assignment constraints that it inherits from its parent when it assigns parameters to its children. For example, if a child RSIP-server is given a lease time of 3600
seconds on an IP address, it MUST compare the current time to
the lease time and the time that the lease was assigned to
calculate the maximum allowable lease time on the address if it
is to assign the address to a RSIP-client or child RSIP-server.

4.2.2. NAT Behind RSIP

+--------+      +--------+
| NAT w/ |      |  RSIP  |
| client |      |        |
|--------+      +--------+

In this architecture, an RSIP server is between a NAT box and
the public network. The NAT is also equipped with an RSIP
client. The NAT dynamically requests resources from the RSIP
server as the clients establish sessions to the public network.
The clients are not aware of the RSIP manipulation. This
configuration does not enable the clients to have end-to-end
transparency and thus the NAT still requires ALGs and the
architecture cannot support IPSEC.

4.2.3. RSIP Behind NAT

+--------+      +--------+
| RSIP   |      |        |
| client |      |        |
|--------+      +--------+

In this architecture, The RSIP clients and server reside behind
a NAT. This configuration does not enable the clients to have
end-to-end transparency and thus the NAT still requires ALGs
and the architecture cannot support IPSEC. The clients may
have transparency if there is another gateway to the public
network besides the NAT box, and this gateway supports cascaded
RSIP behind RSIP.

4.2.4. RSIP Through NAT

+--------+      +--------+
| RSIP   |      |        |
| client |      |        |
|--------+      +--------+

In this architecture, the RSIP clients are separated from the
RSIP server by a NAT. RSIP signaling may be able to pass
through the NAT if an RSIP ALG is installed. The RSIP data flow, however, will have its outer IP address translated by the NAT. The NAT must not translate the port numbers in order for RSIP to work properly. Therefore, only traditional NAT will make sense in this context.

5. Interaction with Other Layer-Three Protocols

Since RSIP affects layer-three objects, it has an impact on other layer three protocols. In this section, we outline the impact of RSIP on these protocols, and in each case, how RSIP, the protocol, or both, can be extended to support interaction.

Each of these sections is an overview and not a complete technical specification. If a full technical specification of how RSIP interacts with a layer-three protocol is necessary, a separate document will contain it.

5.1. IPSEC

RSIP is a mechanism for allowing end-to-end IPSEC with sharing of IP addresses. Full specification of RSIP/IPSEC details are in [RSIP-IPSEC]. This section provides a brief summary.

Since IPSEC may encrypt TCP/UDP port numbers, these objects cannot be used as demultiplexing fields. However, IPSEC inserts an AH or ESP header following the IP header in all IPSEC-protected packets (packets that are transmitted on an IPSEC Security Association (SA)). These headers contain a 32-bit Security Parameter Index (SPI) field, the value of which is determined by the receiving side. The SPI field is always in the clear. Thus, during SA negotiation, an RSIP client can instruct their public peer to use a particular SPI value. This SPI value, along with the assigned IP address, can be used by an RSIP server to uniquely identify and route packets to an RSIP client. In order to guarantee that RSIP clients use SPIs that are unique per address, it is necessary for the RSIP server to allocate unique SPIs to clients along with their address/port tuple.

IPSEC SA negotiation takes place using the Internet Key Exchange (IKE) protocol. IKE is designated to use port 500 on at least the destination side. Some client IKE implementations will use source port 500 as well, but this behavior is not mandatory. If two or more RSIP clients are running IKE at source port 500, they MUST use different initiator cookies (the first eight bytes of the IKE payload) per assigned IP address. The RSIP server will be able to route incoming IKE packets to the proper client based on initiator
cookie value. Initiator cookies can be negotiated, like ports and SPIs. However, since the likelihood of two clients assigned the same IP address attempting to simultaneously use the same initiator cookie is very small, the RSIP server can guarantee cookie uniqueness by dropping IKE packets with a cookie value that is already in use.

5.2. Mobile IP

RSIP can be used as a configuration tool for coarse-grained mobility (nomadity). For example, a laptop or handheld device may use RSIP to temporarily register on a local network. However, once the host de-registers from this network, or otherwise terminates its association with the network, all ongoing communication between the host and its peers must also terminate. It is desirable for RSIP to support fine-grained mobility; i.e., the ability to move between networks, and register and de-register with RSIP servers, without tearing down any communications sessions. Pragmatically speaking, this means that socket parameters, such as the host’s IP address and port number(s) must remain the same as it roams.

Mobile IP [RFC2002] provides the necessary mechanisms for a mobile host to maintain its sessions and socket parameters while it moves between its home network and foreign networks. The goal of this draft is to discuss the architecture, requirements, and feasibility of integrating RSIP and Mobile IP. In doing so we expect that the impact on both protocols will be minimal. In particular, the modifications that we suggest below require minor messaging changes to both RSIP and Mobile IP.

5.2.1. Mobility Architecture

The general architecture that we will consider is illustrated below: This architecture is similar to that discussed in Section 4, but has been annotated specifically for mobility.

\[
\begin{array}{cccccc}
\text{RMNa} & \text{RHAa} & \text{RHAp} & \text{RFAp} & \text{RFAb} & \text{RMNb} \\
\hline
\text{RMN} \rightarrow & \text{RHN} \rightarrow & \text{RFN} \\
\end{array}
\]

In this diagram, an RMN roams between an RHN "a" and an RFN "b". On RHN "a" the RMN uses private address RMNa, and on RFN "b" the RMN uses private address RMNb. The RHA on the RHN uses private address RHAa and public address RHAp. The RFA on the
RFN uses private address RFAb and public address RFAp. Note that the RFN may also act as an RHN for its RMNs, and the RHN may also act as an RFN for roaming RMNs. In order so that the RMN can communicate with peers on the public network, the RHA assigns the RMN address RMNp (not shown). RMNp may be identical to RHAp.

Note that until standard Mobile IP, the address RMNa is not guaranteed to be unique. This address is likely to be from the private space, so when the RMN is on RFN "b", RMNa may collide with other RMNs in the RFN, or with stationary nodes on the network. Thus, it is necessary that the RMN be allocated RMNb when it is on RMN "b". The recommended mechanism to do this with is DHCP.

We assume that the RHA and RFA will always be on the boundary between public and private address spaces. Thus RMNs can always be uniquely identified by the tuple (RMNa, RMNp), although other identification mechanisms may also be used.

5.2.2. Data Flow

This section presents the flow of data between the participants in RSIP / Mobile IP transaction. We ignore both RSIP and Mobile IP control flow and signaling for now - it is discussed in the next section.

5.2.2.1. Non-Roaming RMN

When the RMN is not roaming; i.e., it is on the RHN, it operates just as if it were a stationary node. All data flows according to [RSIP-PROTO].

5.2.2.2. Roaming RMN

When the RMN is roaming, the typical routing scheme of Mobile IP is used. In particular, if the RMN is communicating with a public node (PN) which has address PNp, the RMN will be known to the PN as RMNp. On the RFN, the RMN will have a local IP address of RMNb. In the case of RSAP-IP, the RMN will also be using ports allocated by the RHA. Packets sent from the RN to the RMN will be addressed identically to standard Mobile IP operation:

```
+-------------+---------+
| PNp -> RMNp | payload |
+-------------+---------+
```
Since the RHA will be advertising a route to RMNp, this packet will be received by the RHA. The RHA will determine that the RMN is not on the RHN, and forward the packet to the RMN’s care-of address, RFAp, via an IP-IP tunnel.

```
+-------------------+-------------+---------+
| RHAp -> RFAp      | PNp -> RMNp | payload |
+-------------------+-------------+---------+
```

Once the packet arrives at the RFA, it terminates the IP-IP tunnel from the RHA and initiates a new IP-IP tunnel to the RMN, as shown below.

```
+-------------------+-------------+---------+
| RFAb -> RMNb      | PNp -> RMNp | payload |
+-------------------+-------------+---------+
```

Packets from the RMN are transmitted via the same tunnel back to the RFA, and then on to the PN.

```
+-------------------+-------------+---------+
| RMNb -> RFAb      | RMNp -> PNp | payload |
+-------------------+-------------+---------+
```

5.2.3. Control Flow

In this section, we illustrate the control flow (signaling) requirements for RSIP / Mobile IP.

The RMN is always listening for the ICMP Mobile IP Agent Advertisement messages. Upon receipt of such a message the RMN determines whether or not it is on the RHN. The RMN may also transmit Agent Solicitation messages, as per Mobile IP.

When the RMN determines that it has moved from a RHN to a RFN, or from a RFN to another RFN, it requests a local address (e.g., RMNb from above) then performs the Mobile IP registration process. This process includes informing the RHA of the RMN’s new care-of address. Note that the RMN must identify itself uniquely to the RHA. Since RMNp may be in use by more than one RMN at a time, the RMN must use either RMNa or a combination of RMNp and a port number or range of port numbers (in the case of RSAP-IP) that it has been allocated by the RHA.

All RSIP messages that would normally flow between the RMN and the RHA must be forwarded by the RFA. The RMN may request more resources from or return resources to the RHA. The RMN must
also be prepared to accept DEALLOCATE messages from the RHA which would force it to discontinue use of certain resources. Note that if the RMN de-registers from the RHA, it may lose connectivity entirely.

All RSIP control messages must be tunneled between the RMN and the RFA, as well as between the RFA and the RHA. The form of these tunnels is illustrated below.

From RMN to RFA:

+-------------------------------------------------------+
| RMNb -> RFAb | RMNa -> RHAA | payload     |
+-------------------------------------------------------+

From RFA to RHA:

+-------------------------------------------------------+
| RFAp -> RHAp | RMNa -> RHAA | payload     |
+-------------------------------------------------------+

From RHA to RFA:

+-------------------------------------------------------+
| RHAp -> RFAp | RHAa -> RMNa | payload     |
+-------------------------------------------------------+

From RFA to RMN:

+-------------------------------------------------------+
| RFAb -> RMNb | RHAa -> RMNa | payload     |
+-------------------------------------------------------+

Since the RMN has a presence, in the form of an address (RMNb) on the RFN, it must reply to all ARP messages for RMNb. However, it MUST NOT respond to ARPs for RMNa or RMNp. The RMN may communicate with other nodes on the RFN by using RMNb, and is thus not constrained to use port numbers allocated by the RHA (in the case of RSAP-IP) when communicating locally.

The RHA must perform all RSIP server functions as defined in [RSIP-PROTO]. The RHA must also perform all home agent Mobile IP functions as defined in [RFC2002].

The RFA must perform all foreign agent Mobile IP functions as defined in [RFC2002]. It must also maintain an IP-IP tunnel to all RMNs so that they can pass control and data flow packets.
5.3. Differentiated and Integrated Services

To attain the capability of providing quality of service between two communicating hosts in different realms, it is important to consider the interaction of RSIP with different quality of service provisioning models and mechanisms. In the section, RSIP interaction with the integrated service and differentiated service frameworks is discussed.

5.3.1. Differentiated Services

The differentiated services architecture defined in [RFC2475] allows networks to support multiple levels of best-effort service through the use of "markings" of the IP Type-of-Service (now DS) byte. Each value of the DS byte is termed a differentiated services code point (DSCP) and represents a particular per-hop behavior. This behavior may not be the same in all administrative domains. No explicit signaling is necessary to support differentiated services.

For outbound packets from an edge network, DSCP marking is typically performed and/or enforced on a boundary router. The marked packet is then forwarded onto the public network. In a RSIP-enabled network, a natural place for DSCP marking is the RSIP server. In the case of RSAP-IP, the RSIP server can apply its micro-flow (address/port tuple) knowledge of RSIP assignments in order to provide different service levels to different RSIP client. For RSA-IP, the RSIP server will not necessarily have knowledge of micro-flows, so it must rely on markings made by the RSIP clients (if any) or apply a default policy to the packets.

Given most reasonable RSIP deployment scenarios, it is not likely that supporting differentiated services on the private network will be absolutely necessary (e.g., the RSIP clients and server are one hop apart). However, if differentiated services is to be performed between RSIP clients and servers, it must be done over the tunnel between these entities. Differentiated services over a tunnel is considered in detail in [DS-TUNN], the key points that need to be addressed here are the behaviors of tunnel ingress and egress for both incoming and going packets.

For incoming packets arriving at an RSIP server tunnel ingress, the RSIP server may either copy the DSCP from the inner header to the outer header, leave the inner header DSCP untouched, but place a different DSCP in the outer header, or change the inner header DSCP while applying either the same or a different DSCP.
to the outer header.

For incoming packets arriving at an RSIP client tunnel egress, behavior with respect to the DSCP is not necessarily important if the RSIP client not only terminates the tunnel, but consumes the packet as well. If this is not the case, as per some cascaded RSIP scenarios, the RSIP client must apply local policy to determine whether to leave the inner header DSCP as is, overwrite it with the outer header DSCP, or overwrite it with a different value.

For outgoing packets arriving at an RSIP client tunnel ingress, the client may either copy the DSCP from the inner header to the outer header, leave the inner header DSCP untouched, but place a different DSCP in the outer header, or change the inner header DSCP while applying either the same or a different DSCP to the outer header.

For outgoing packets arriving at an RSIP server tunnel egress, the RSIP server must apply local policy to determine whether to leave the inner header DSCP as is, overwrite it with the outer header DSCP, or overwrite it with a different value.

5.3.2. Integrated Services

The integrated services model as defined by [RFC2205] requires signalling using RSVP to setup a resource reservation in intermediate nodes between the communicating endpoints. In the most common scenario in which RSIP is deployed, receivers located within the private realm initiate communication sessions with senders located within the public realm. In this section, we discuss the interaction of RSIP architecture and RSVP in such a scenario. The less common case of having senders within the private realm and receivers within the public realm is not discussed although concepts mentioned here may be applicable.

With senders in the public realm, RSVP PATH messages flow downstream from sender to receiver, inbound with respect to the RSIP server, while RSVP RESV messages flow in the opposite direction. Since RSIP uses tunneling between the RSIP client and server within the private realm, how the RSVP messages are handled within the RSIP tunnel depends on situations elaborated in [RSVP-Tunnel].

Following the terminology of [RSVP-Tunnel], if Type 1 tunnels exist between the RSIP client and server, all intermediate nodes inclusive of the RSIP server will be treated as a non-
RSVP aware cloud without QoS reserved on these nodes. The tunnel will be viewed as a single (logical) link on the path between the source and destination. End-to-end RSVP messages will be forwarded through the tunnel encapsulated in the same way as normal IP packets. We see this as the most common and applicable deployment scenario.

However, should Type 2 or 3 tunnels be deployed between the tunneling endpoints, end-to-end RSVP session has to be statically mapped (Type 2) or dynamically mapped (Type 3) into the tunnel sessions. While the end-to-end RSVP messages will be forwarded through the tunnel encapsulated in the same way as normal IP packets, a tunnel session is established between the tunnel endpoints to ensure QoS reservation within the tunnel for the end-to-end session. Data traffic needing special QoS assurance will be encapsulated in a UDP/IP header while normal traffic will be encapsulated using the normal IP-IP encapsulation. In the type 2 deployment scenario where all data traffic flowing to the RSIP client receiver are given QoS treatment, UDP/IP encapsulation will be rendered in the RSIP server for all data flows. The tunnel between the RSIP client and server could be seen as a "hard pipe". Traffic exceeding the QoS guarantee of the "hard pipe" would fall back to the best effort IP-IP tunneling.

In the type 2 deployment scenario where data traffic could be selectively channeled into the UDP/IP or normal IP-IP tunnel, or for type 3 deployment where end-to-end sessions could be dynamically mapped into tunnel sessions, integration with the RSIP model could be complicated and tricky. (Note that these are the cases where the tunnel link could be seen as a expandable soft pipe). Two main issues are worth considering.

- For RSIP server implementations that do encapsulation of the incoming stream before passing to the IP layer for forwarding, the RSVP daemon has to be explicitly signaled upon reception of incoming RSVP PATH messages. The RSIP implementation has to recognize RSVP PATH messages and pass them to the RSVP daemon instead of doing the default tunneling. Handling of other RSVP messages would be as described in [RSVP-Tunnel].

- RSIP enables an RSIP client to have a temporary presence at the RSIP server by assuming one of the RSIP server’s global interfaces. As a result, the RSVP PATH messages would be addressed to the RSIP server. Also, the RSVP SESSION object within an incoming RSVP PATH would carry the global destination address, destination port (and protocol) tuples
that were leased by the RSIP server to the RSIP client. Hence the realm unaware RSVP daemon running on the RSIP server has to be presented with a translated version of the RSVP messages. Other approaches are possible, for example making the RSVP daemon realm aware.

A simple mechanism would be to have the RSIP module handle the necessary RSVP message translation. For an incoming RSVP signalling flow, the RSIP module does a packet translation of the IP header and RSVP SESSION object before handling the packet over to RSVP. The global address leased to the client is translated to the true private address of the client. (Note that this mechanism works with both RSA-IP and RSAP-IP). The RSIP module also has to do an opposite translation from private to global parameter (plus tunneling) for end-to-end PATH messages generated by the RSVP daemon towards the RSIP client receiver. A translation on the SESSION object also has to be done for RSVP outbound control messages. Once the RSVP daemon gets the message, it maps them to an appropriate tunnel sessions.

Encapsulation of the inbound data traffic needing QoS treatment would be done using UDP-IP encapsulation designated by the tunnel session. For this reason, the RSIP module has to be aware of the UDP-IP encapsulation to use for a particular end-to-end session. Classification and scheduling of the QoS guaranteed end-to-end flow on the output interface of the RSIP server would be based on the UDP/IP encapsulation. Mapping between the tunnel session and end-to-end session could continue to use the mechanisms proposed in [RSVP-Tunnel]. Although [RSVP-Tunnel] proposes a number of approaches for this purpose, we propose using the SESSION_ASSOC object introduced because of its simplicity.

5.4. IP Multicast

The amount of specific RSIP/multicast support that is required in RSIP clients and servers is dependent on the scope of multicasting in the RSIP-enabled network, and the roles that the RSIP clients will play. In this section, we discuss RSIP and multicast interactions in a number of scenarios.

Note that in all cases, the RSIP server MUST be multicast aware because it is on an administrative boundary between two domains that will not be sharing their all of their routing information. The RSIP server MUST NOT allow private IP addresses to be propagated on the public network as part of any multicast message or as part of a routing table.
5.4.1. Receiving-Only Private Hosts, No Multicast Routing on Private Network

In this scenario, private hosts will not source multicast traffic, but they may join multicast groups as recipients. In the private network, there are no multicast-aware routers, except for the RSIP server.

Private hosts may join and leave multicast groups by sending the appropriate IGMP messages to an RSIP server (there may be IGMP proxy routers between RSIP clients and servers). The RSIP server will coalesce these requests and perform the appropriate actions, whether they be to perform a multicast WAN routing protocol, such as PIM, or to proxy the IGMP messages to a WAN multicast router. In other words, if one or more private hosts request to join a multicast group, the RSIP server MUST join in their stead, using one of its own public IP addresses.

Note that private hosts do not need to acquire demultiplexing fields and use RSIP to receive multicasts. They may receive all multicasts using their private addresses, and by private address is how the RSIP server will keep track of their group membership.

5.4.2. Sending and Receiving Private Hosts, No Multicast Routing on Private Network

This scenario operates identically to the previous scenario, except that when a private host becomes a multicast source, it MUST use RSIP and acquire a public IP address (note that it will still receive on its private address). A private host sending a multicast will use a public source address and tunnel the packets to the RSIP server. The RSIP server will then perform typical RSIP functionality, and route the resulting packets onto the public network, as well as back to the private network, if there are any listeners on the private network.

If there is more than one sender on the private network, then, to the public network it will seem as if all of these senders share the same IP address. If a downstream multicasting protocol identifies sources based on IP address alone and not port numbers, then it is possible that these protocols will not be able to distinguish between the senders.

6. Changelog

02 to 03
- Added section on interaction with Integrated and Differentiated
7. 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.

8. Acknowledgements

The authors would like to thank Pyda Srisuresh, Dan Nessett, Gary Borella et. al.
Jaszewski, Ajay Bakre, Cyndi Jung, and Rick Cobb for their input.
The IETF NAT working group as a whole has been extremely helpful in
the ongoing development of RSIP.

9. References

[DHC-NS] J. Privat and M. Borella, "DHCP Next Server Option," <draft-

[DHCP-DNS] M. Stapp and Y. Rekhter, "Interaction Between DHCP and

[DS-TUNN] D. Black, "Differentiated Services and Tunnels," <draft-

[RSIP-IPSEC] G. Montenegro and M. Borella, "RSIP Support for End-to-
end IPSEC," <draft-ietf-nat-rsip-ipsec-01.txt>, work in progress,

[RSIP-PROTO] M. Borella, D. Grabelsky, J. Lo and K. Taniguchi,
"Realm Specific IP: Protocol Specification," <draft-ietf-nat-rsip-

[RSVP-Tunnel] A. Terzis, J. Krawczyk, J. Wroclawski, L. Zhang, "RSVP
Operation Over IP Tunnels," <draft-ietf-rsvp-tunnel-04.txt>, work-
in-progress, Nov. 1999

[RFC1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot,
and E. Lear, "Address Allocation for Private Internets," RFC 1918,
Feb. 1996.


[RFC2119] S. Bradner, "Key words for use in RFCs to indicate

[RFC2663] P. Srisuresh and Matt Holdrege, "IP Network Address
1999.


Weiss, "An Architecture for Differentiated Services," RFC 2475,
Dec. 1998
10. Authors' Addresses

Michael Borella
3Com Corp.
1800 W. Central Rd.
Mount Prospect IL 60056
(847) 342-6093
mike_borella@3com.com

Jeffrey Lo
NEC USA
C&C Research Labs.
110 Rio Robles
San Jose, CA 95134
(408) 943-3033
jlo@ccrl.sj.nec.com

David Grabelsky
3Com Corp.
1800 W. Central Rd.
Mount Prospect IL 60056
(847) 222-2483
david_grabelsky@3com.com

Gabriel E. Montenegro
Sun Microsystems, Inc.
15 Network Circle
Menlo Park CA 94025
650 786 6288
gab@sun.com

11. Copyright Statement

Copyright (c) The Internet Society (1999). 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.