Speermint Working Group
Internet Draft
Expires: February 2007

R. Penno
Juniper Networks
D. Malas
Level 3
S. Khan
Sprint
A. Uzelac
Global Crossing

September 6, 2006

SPEERMINT Routing Architecture Message Flows
draft-ietf-speermint-flows-00

Status of this Memo

Abstract

This draft provides the message flows associated with the SPEERMINT, SIP Peering and Multimedia Interconnect, routing architecture. This

document provides examples of many different message flows relative to varying peering scenarios.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

Table of Contents

1. Introduction

This document shows the message flows associated with the most relevant SPEERMINT routing architecture peering scenarios. Most of the message diagrams were based on previous work described within existing IETF standards documents.

The document focuses on the messages exchanged for the purpose of Layer 5 peering [7] between two domains. Messages exchanged for the purpose of setting up SIP sessions within a domain are considered out of scope and were already defined in other IETF documents.

The draft separates the Layer 5 peering scenarios in two major peering scenarios.

o On-demand: In this scenario the SIP proxies in domains A and B establish a peering relationship driven by the necessity to deliver a SIP message to another domain. This is sometimes referred as the "email" model.

o Static: In this scenario the peering relationship between proxies A and B is statically provisioned independent of the establishment of any SIP session between users in different domains.

Normally, media for a given SIP session follows a different path, traversing a different device (most commonly a router) when crossing

peering domains. Alternatively, media for a given session can be directed to traverse the same device used for Layer 5 peering, i.e., the same device that handles signaling when crossing domains. This produces two different models:

- Decomposed: In this model SIP proxies perform Layer 5 peering and media is sent directly between the User Agent's (UA's) involved in the session. Signaling and media follow different paths.

- Collapsed: In this model the device that performs Layer 5 peering also processes the associated media when crossing domains. In the light of SPEERMINT these devices may need to process media mainly when peering involves SIP entities in private address spaces. This function is usually referred to as media relay and is usually performed by a B2BUA or SBC (Session Border Controller). See [6] for a complete discussion of SBC functions. The decomposed or basic peering model picture is shown below. It is worth mentioning that Proxy 1 and 2 can be separated by any number of layer 3 hops. We will refer to this picture throughout this document.

```
   ............................          ..............................
   .                          .          .                            .
   .                +-------+ .          . +-------+                  .
   .                |       | .          . |       |                  .
   .                |  DNS  | .          . | DNS   |                  .
   .                |   1   | .          . |  2    |                  .
   .                |       | .          . |       |                  .
   .                +-------+ .          . +-------+                  .
   .                    |     .          .     |                      .
   .                    |     .          .     |                      .
   .                +-------+ .          . +-------+                  .
   .                |       | .          . |       |                  .
   .                | Proxy |--------------| Proxy |                  .
   .                |   1   | .          . |  2    |                  .
   .                |       | .          . |       |                  .
   .              / +-------+ .          . +-------+ \                .
   .             /            .          .            \               .
   .            /             .          .             \              .
   .           /              .          .              \             .
   .          /               .          .               \            .
   .         /                .          .                \           .
   .        /                 .          .                 \          .
   .       /                  .          .                  \         .
   .   +-------+              .          .                +-------+   .
   .   |       |              .          .                |       |   .
   .   |       |              .   Media  .                |       |   .
   .   | UA 1  |<========================================>| UA 2  |   .
   .   |       |              .          .                |       |   .
   .   +-------+              .          .                +-------+   .
   .              Domain A    .          .   Domain B                 .
   ............................          ..............................
```

Figure 1 Basic Peering Picture.

The collapsed model is shown below:

```
   ............................          ..............................
   .                          .          .                            .
   .                +-------+ .          . +-------+                  .
   .                |       | .          . |       |                  .
   .                |  DNS  | .          . | DNS   |                  .
   .                |   1   | .          . |  2    |                  .
   .                |       | .          . |       |                  .
   .                +-------+ .          . +-------+                  .
   .                    |     .          .     |                      .
   .                    |     .          .     |                      .
   .                +-------+ .          . +-------+                  .
   .                | B2BUA | .          . | B2BUA |                  .
   .                |  &    |--------------|  &    |                  .
   .                | other |**************| other |\                 .
   .               /| funct | .          . | funct | \                .
   .              / +-------+ .          . +-------+* \ signaling     .
   .             / *          .          .           * \              .
   .            / *           .          .            * \             .
   .           / *            .          .             * \            .
   .          / * media       .          .              * \           .
   .         / *              .          .               * \          .
   .        / *               .          .                * \         .
   .       / *                .          .                 * \        .
   .   +-------+              .          .                +-------+   .
   .   |       |              .          .                |       |   .
   .   |       |              .          .                |       |   .
   .   | UA 1  |              .          .                | UA 2  |   .
   .   |       |              .          .                |       |   .
   .   +-------+              .          .                +-------+   .
   .              Domain A    .          .   Domain B                 .
   ............................          ..............................
```

Figure 2 Collapsed Peering Picture.

In a decomposed model, the signaling function (SF) and the media function (MF) are implemented in separate entities. A B2BUA is generally on the SIP path in the SF. The vertical control protocol between the SF and MF is out of the scope of this document. The decomposed model is shown below:

```
   ............................          ..............................
   .                          .          .                            .
   .                +-------+ .          . +-------+                  .
   .                |       | .          . |       |                  .
   .                |  DNS  | .          . | DNS   |                  .
   .                |   1   | .          . |  2    |                  .
   .                |       | .          . |       |                  .
   .                +-------+ .          . +-------+                  .
   .                    |     .          .     |                      .
   .                    |     .          .     |                      .
   .                +-------+ .          . +-------+                  .
   .               /| B2BUA |--------------| B2BUA |\                 .
   .              / +-------+              +-------+ \                .
   .             /  +-------+              +-------+  \               .
   .            /   | MF    |**************| MF    |   \              .
   .           /    +-------+ .          . +-------+*   \signaling    .
   .          /    *          .          .           *   \            .
   .         /    *           .          .            *   \           .

   .        /    *            .          .             *   \          .
   .       /    * media       .          .              *   \         .
   .      /    *              .          .               *   \        .
   .     /    *               .          .                *   \       .
   .    /    *                .          .                 *   \      .
   .   +-------+              .          .                +-------+   .
   .   |       |              .          .                |       |   .
   .   |       |              .          .                |       |   .
   .   | UA 1  |              .          .                | UA 2  |   .
   .   |       |              .          .                |       |   .
   .   +-------+              .          .                +-------+   .
   .              Domain A    .          .   Domain B                 .
   ............................          ..............................
```

Figure 3 Collapsed Peering Picture.

2. Peering Message flows

We first depict what we call the basic message flow. The various scenarios differ mostly of how and when peering is implemented. As mentioned earlier peering can be establish following the arrival of a message at a border proxy or statically following an agreement between both domains.

```
        Alice    Proxy 1       DNS      Proxy 2      Bob
           |        |          |         |           |
           |        |          |         |           |
           |        |       Peering      |           |
           |        |        Phase       |           |
           |        |       [Static]     |           |
           |        |<------------------>|           |
           | INVITE |          |         |           |
           |------->|          |         |           |
           |  100   |                    |           |
           |<-------|                    |           |
           |        | NAPTR    |         |           |
           |        | Query    |         |           |
           |        |--------->|         |           |
           |        | NAPTR    |         |           |
           |        | Reply    |         |           |
           |        |"SIPS+D2T"|         |           |
           |        |<---------|         |           |
           |        | SRV      |         |           |
           |        | Query    |         |           |
           |        |--------->|         |           |
           |        | SRV      |         |           |
           |        | Reply    |         |           |
           |        |<---------|         |           |
           |        |                    |           |
           |        |       Peering      |           |
           |        |        Phase       |           |
           |        |     [On-Demand]    |           |
           |        |<------------------>|           |
           |        |  INVITE            |           |
           |        |------------------->|  INVITE   |
           |        |        100         |---------->|
           |        |<-------------------|           |
           |        |                    | 180       |
           |        |        180         |<----------|
           | 180    |<-------------------|           |
           |<-------|                    | 200       |
           |        |        200         |<----------|
           | 200    |<-------------------|           |
           |<-------|                    |           |
           | ACK    |                    |           |
           |------->| ACK                |           |
           |        |------------------->| ACK       |
           |        |                    |---------->|
           |           Both Way RTP Media            |
           |<======================================>|
```

```
           |        |                    | BYE       |
           |        | BYE                |<----------|
           | BYE    |<-------------------|           |
           |<-------|                    |           |
           | 200    |                    |           |
           |------->| 200                |           |
           |        |------------------->| 200       |
           |        |                    |---------->|
           |        |                    |           |
```

In the collapsed model, media would follow the path shown below. All other signaling call flows remain the same, except a B2BUA is used instead of a proxy.

```
        Alice   B2BUA 1       DNS      B2BUA 2      Bob
           |        |          |         |           |
           |        |          |         |           |
           |           Both Way RTP Media            |
           |<======>|<===================>|<==========>|
           |        |                    |           |
```

The following sections show the message flows in several different scenarios broken in two categories, on-demand and static.

3. On-Demand Peering

In the on demand peering scenario, the relationship between proxies in domains A and B is driven by the arrival of a SIP message at proxy A directed to a user in domain B (or vice-versa).

3.1. Transport Layer Security

In the case this is in fact the first call between those two VSPs, than this call will trigger the establishment of the TLS connection. Otherwise we can assume the TLS connection has been established by some other means.

```
        Alice   Proxy 1       DNS      Proxy 2      Bob
           |        |          |         |           |
           |        |          |         |           |
           | INVITE |          |         |           |
           |------->|          |         |           |
           | 100    |          |         |           |
           |<-------|          |         |           |
           |        | NAPTR    |         |           |
           |        | Query    |         |           |
           |        |--------->|         |           |
           |        | NAPTR    |         |           |
           |        | Reply    |         |           |
           |        |"SIPS+D2T"|         |           |
           |        |<---------|         |           |
           |        | SRV      |         |           |
           |        | Query    |         |           |
           |        |--------->|         |           |
           |        | SRV      |         |           |
           |        | Reply    |         |           |
           |        |<---------|         |           |
           |        |          |         |           |
           |        |       Peering      |           |
           |        |  [TLS Connection]  |           |
           |        |<------------------>|           |
           |        |                    |           |
           |        |  INVITE            |           |
           |        |------------------->| INVITE    |
           |        |  100               |---------->|
           |        |<-------------------|           |
           |        |                    | 180       |
           |        | 180                |<----------|
           | 180    |<-------------------|           |
           |<-------|                    | 200       |
           |        | 200                |<----------|
           | 200    |<-------------------|           |
           |<-------|                    |           |
           | ACK    |                    |           |
           |------->| ACK                |           |
           |        |------------------->| ACK       |
           |        |                    |---------->|
           |           Both Way RTP Media            |
           |<=======================================>|
           |        |                    | BYE       |
           |        | BYE                |<----------|
           | BYE    |<-------------------|           |
           |<-------|                    |           |
           | 200    |                    |           |
```

```
           |------->| 200                 |           |
           |        |-------------------->| 200       |
           |        |                     |---------->|
           |        |                     |           |
```

TBD: DNS exchange could present proxy 1 with a set of peering policies that need to be met for the peering with proxy 2 too succeed.

3.2. Proxy Authentication: Subscribe/Notify

In the following example message flow, the authentication credentials exchange method may take place before any INVITE is sent by ALICE. The P2Key is sent by Proxy 2's NOTIFY and is included within subsections of the peering policy event package (PeerPlcyEvtPkg). The P2Key may be stored on Proxy 1 for the duration of the policy subscription. When the subscription expires, the P2Key becomes invalid. At any time before the subscription expires, the P2Key MAY be updated or refreshed as described in [8]. The message flow and authentication exchange may occur in either direction, but for simplicity reasons is only shown unilaterally.

```
   ALICE            Proxy 1(P1)                   Proxy 2(P2)     Bob
      |                |                             |             |
      | INVITE         |                             |             |
      |--------------->|                             |             |
      |     100 Trying |                             |             |
      |<---------------|                             |             |
      |                | Subscribe w/ PeerPlcyEvtPkg |             |
      |                |---------------------------->|             |
      |                |   401 Unauthorized          |             |
      |                |<----------------------------|             |
      |                |Subscribe w/Auth             |             |
      |                |---------------------------->|             |
      |                |    202 Accepted             |             |
      |                |<----------------------------|             |
      |                |    Notify w/P2Key           |             |
      |                |<----------------------------|             |
      |                |          200 OK             |             |
      |                |---------------------------->|             |
      |                |   INVITE                    |             |
      |                |---------------------------->|             |
      |                |  401 Unauthorized           |             |
      |                |<----------------------------|             |
      |                |    INVITE w/P2Key           |             |
      |                |---------------------------->|             |
      |                |      100 Trying             |INVITE       |
      |                |<----------------------------|-----------> |
      |                |                             |180 Ringing  |
      |                |                  180 Ringing |<----------- |
      |     180 Ringing|<----------------------------|             |
      |<---------------|                             |200 OK       |
      |                |       200 OK                |<----------- |
      |     200 OK     |<----------------------------|             |
      |<---------------|                             |             |
      |  ACK           |                             |             |
      |--------------->|ACK                          |             |
      |                |---------------------------->|ACK          |
      |                |                             |-----------> |
```

3.3. Proxy Authentication: Surrogate Registration

In this optional scenario we are assuming a new proxy authentication method exists that allows mutual authentication between two proxies. This authentication can be termed as the "Surrogate Authentication". Generally, a proxy cannot register with another proxy because in between two proxies there is not a child-parent relationship;

however, an originating proxy can register with another proxy on behalf of a UA.

```
      ALICE            Proxy 1(P1)          Proxy 2(P2)     Bob
         |                |                    |             |
         | INVITE         |                    |             |
         |--------------->|                    |             |
         |     100 Trying |                    |             |
         |<---------------|                    |             |
         |                |REGISTER            |             |
         |                |------------------->|             |
         |                |   401 Unauthorized |             |
         |                |<-------------------|             |
         |                |REGISTER w/Auth     |             |
         |                |------------------->|             |
         |                |    100 Trying      |             |
         |                |<-------------------|             |
         |                |    200 OK w/P2Key  |             |
         |                |<-------------------|             |
         |                |          REGISTER  |             |
         |                |<-------------------|             |
         |                |   401 Unauthorized |             |
         |                |------------------->|             |
         |                |     REGISTER w/Auth|             |
         |                |<-------------------|             |
         |                |  100 Trying        |             |
         |                |------------------->|             |
         |                |  200 OK w/P1Key    |             |
         |                |------------------->|             |
         |                |    INVITE w/P2Key  |             |
         |                |------------------->|             |
         |                |      100 Trying    |INVITE       |
         |                |<-------------------|-----------> |
         |                |                    |180 Ringing  |
         |                |       180 Ringing  |<----------- |
         |     180 Ringing|<-------------------|             |
         |<---------------|                    |200 OK       |
         |                |       200 OK       |<----------- |
         |     200 OK     |<-------------------|             |
         |<---------------|                    |             |
         |  ACK           |                    |             |
         |--------------->|ACK                 |             |
         |                |------------------->|ACK          |
         |                |                    |-----------> |
```

4. Static Peering

In the static peering scenario the relationship between proxies A and B is not driven by a SIP session, but before hand through manual provisioning.

4.1. IPSec

In this model an IPSec connection between proxies A and B is provisioned following an agreement between the two domains.

```
        Alice   Proxy 1       DNS      Proxy 2      Bob
           |        |          |          |           |
           |        |                     |           |
           |        |      [Peering]      |           |
           |        |   IPSec Connection  |           |
           |        |<------------------->|           |
           |        |          |          |           |
           \        /          \          /           \
           /        \          /          \           /
           |        |          |          |           |
           | INVITE |          |          |           |
           |------->|          |          |           |
           | 100    |          |          |           |
           |<-------|          |          |           |
           \        /          \          /           \
           /        \          /          \           /
           |        | BYE                 |<----------|
           | BYE    |<--------------------|           |
           |<-------|                     |           |
           | 200    |                     |           |
```

4.2. Co-Location

In this scenario the two proxies are co-located in a physically secure location and/or are members of a segregated network. In this case messages between Proxy 1 and Proxy 2 would be sent as clear text.

```
        Alice   Proxy 1       DNS      Proxy 2      Bob
           |        |          |         |           |
           |        |                    |           |
           |        |      [Peering]     |           |
           |        |     Co-Location    |           |
           |        |<------------------>|           |
           |        |          |         |           |
           \        /          \         /           \
           /        \          /         \           /
           |        |          |         |           |
           | INVITE |          |         |           |
           |------->|          |         |           |
           | 100    |          |         |           |
           |<-------|          |         |           |
           \        /          \         /           \
           /        \          /         \           /
           |        | BYE                |<----------|
           | BYE    |<-------------------|           |
           |<-------|                    |           |
           | 200    |                    |           |
```

5. Federation Based Peering

The Domain Policy DDDS framework [13] can be used to integrate on-demand peering and static peering into one unified setup. The main idea is that the target can use its domain to publish peering-related information in the DNS. Federations as defined in [14] are one way how source and destination network can find a common set of procedures for the peering.

Federation based peering is thus not a substitute to the various authentication, routing, and QoS procedures which are described in this document.

The following examples demonstrate how Alice can use this scheme to dynamically select the correct peering mechanisms when talking to Bob.

The overall message flow is similar to the one from section 3.1. The DP-DDDS queries the DNS for the same NAPTR records as the algorithm from RFC 3263 [3]. While the originating network behavior according to [3] depends solely on the results retrieved from DNS, the DP-DDDS also uses a set of local configuration options to drive the source network behavior. The following examples thus list both the sender configuration and the answers from the DNS.

5.1. Simple Federation Match

The simplest case is when Alice and Bob share membership in one federation ("http://example.com/Wonderland") which stipulates further call-setup according to section 3.1.

Configuration at Alice's DNS list Alice's federations (which includes http://example.com/Wonderland) and rules what do to when a federation is chosen for a call.

```
NAPTR RRset at Bob's domain includes:
 IN NAPTR 10 50 "u" "D2P+SIP:fed" (
    "!^.*$!http://example.com/small-federation!" . )
 IN NAPTR 20 50 "u" "D2P+SIP:fed" (
    "!^.*$!http://example.com/Wonderland!" . )

     Alice   Proxy 1       DNS      Proxy 2      Bob
        |         |          |          |           |
        |         |          |          |           |
        | INVITE  |          |          |           |
        |-------->|          |          |           |
        | 100     |          |          |           |
        |<--------|          |          |           |
        |         | NAPTR    |          |           |
        |         | Query    |          |           |
        |         |--------->|          |           |
        |         | NAPTR    |          |           |
        |         | Reply    |          |           |
        |         |<---------|          |           |
        | Parse D2P+SIP RRs |          |           |
        | Federation match  |          |           |
        |   successful      |          |           |
        |         |          |          |           |
        | Parse NAPTR with  |          |           |
        |    "SIPS+D2T"     |          |           |
        |         |          |          |           |
        |         | SRV      |          |           |
        |         | Query    |          |           |
        |         |--------->|          |           |
          [Rest according to section 3.1]
```

5.2. No federation match

If Bob does not share a federation with Alice, e.g. by just being a member of the "small-federation", then no direct peering is possible between Alice and Bob.

```
   Bob's Domain contains:
    IN NAPTR 10 50 "u" "D2P+SIP:fed" (
       "!^.*$!http://example.com/small-federation!" . )


        Alice   Proxy 1       DNS      Proxy 2      Bob
           |        |          |          |            |
           |        |          |          |            |
           | INVITE |          |          |            |
           |------->|          |          |            |
           | 100    |          |          |            |
           |<-------|          |          |            |
           |        | NAPTR    |          |            |
           |        | Query    |          |            |
           |        |--------->|          |            |
           |        | NAPTR    |          |            |
           |        | Reply    |          |            |
           |        |<---------|          |            |
           | Parse D2P+SIP RRs |          |            |
           | Federation match  |          |            |
           |   failed.         |          |            |
           |        |          |          |            |
           | Bob offers no alternative ways            |
           |  No peering is possible.     |            |
           |        |          |          |            |
```

If no matching federations or referrals are found, Alice can either fall back to PSTN routing or use a transit VSP.

5.3. Federation Referral

If Bob buys transit services from Carol, he can announce this in a "D2P+SIP" NAPTR record. We now have at Bob's domain:

```
    IN NAPTR 10 50 "u" "D2P+SIP:fed" (
       "!^.*$!http://example.com/small-federation!" . )
    IN NAPTR 20 50 "u" "D2P+SIP" "" carol.example.com.
```

If Carol is a member of the Wonderland federation, then we have

```
   $ORIGIN carol.example.com
    IN NAPTR 10 50 "u" "D2P+SIP:fed" (
       "!^.*$!http://example.com/Wonderland!" . )
```

```
        Alice   Proxy 1       DNS      Proxy 2      Bob
          |        |           |          |           |
          |        |           |          |           |
          | INVITE |           |          |           |
          |------->|           |          |           |
          | 100    |           |          |           |
          |<-------|           |          |           |
          |        | NAPTR     |          |           |
          |        | Query     |          |           |
          |        |---------->|          |           |
          |        | NAPTR     |          |           |
          |        | Reply     |          |           |
          |        |<----------|          |           |
          | Parse D2P+SIP RRs |          |           |
          | direct federation |          |           |
          |   match fails     |          |           |
          | Found non-terminal|          |           |
          |        |           |Alice retargets to Carol
          |        | NAPTR     |          |           |
          |        | Query     |          |           |
          |        |---------->|          |           |
          |        | NAPTR     |          |           |
          |        | Reply     |          |           |
          |        |<----------|          |           |
          | Parse D2P+SIP RRs |          |           |
          | Federation match  |          |           |
          |   successful      |          |           |
          |        |           |          |           |
          | Parse NAPTR with  |          |           |
          |    "SIPS+D2T"     |          |           |
          |        |           |          |           |
          |        | SRV       |          |           |
          |        | Query     |          |           |
          |        |---------->|          |           |
            [Rest according to section 3.1]
```

5.4. Federation Specific Call Processing

The output of the federation matching step in the Domain Policy DDDS application is a federation name and a destination domain (which differs from the original destination domain if referrals were followed).

Federations as defined in [14] can specify their own specific rules on how the actual call-setup is to be performed between two federation members. If Alice is a member of more than one federation

then Alice's peering SIP proxy needs to adapt its behavior to the rules of the federation this call is traversing.

The following subsections provide some examples of what a federation could imply for the call processing.

5.4.1. Central Federation Proxy

Federation rules can dictate that calls are to be routed via a federation-maintained central SIP proxy. In that case no further NAPTR/SRV/A lookups are made. Instead, the INVITE will be sent directly via a preconfigured TLS connection to that proxy. This proxy acts as a redirect proxy.

The following message flow provides an example describing this process:

```
        Peer Proxy     Federation Proxy   Peer Proxy          Bob
            |                |                |                |
            |   INVITE       |                |                |
            |--------------->|                |                |
            |     302        |                |                |
            |<---------------|                |                |
            |     ACK        |                |                |
            |--------------->|                |                |
            |     INVITE                      |                |
            |-------------------------------->|    INVITE      |
            |            100                  |--------------->|
            |<--------------------------------|   180          |
            |            180                  |<---------------|
            |<--------------------------------|                |
            |                                 |    200         |
            |            200                  |<---------------|
            |<--------------------------------|                |
            |            ACK                  |                |
            |-------------------------------->|     ACK        |
            |                                 |--------------->|
            |                Both Way RTP Media                |
            |<================================================>|
            |                                 |     BYE        |
            |            BYE                  |<---------------|
            |<--------------------------------|                |
            |            200                  |                |
            |-------------------------------->|     200        |
            |                                 |--------------->|
```

5.4.2. VPN Based Federations

If a federation has established some sort of VPN which connects the SIP elements of all participating VSPs, then matching that federation will cause:

Proxy1 to use e.g. a private DNS within that VPN for further lookups and will direct all further traffic to be routed into that VPN.

IPsec based VPNs are a special case of this.

5.4.3. TLS Based Federation

One of the simplest cases is a TLS based federation.

In that case the federation rules may prescribe the default NAPTR/SRV lookups and only affect the selection of the correct X.509 certificate for the TLS connection.

6. Considerations on Private [13] IP addresses

In Layer 5 peering scenarios, it does not really matter if the peering fabric is public or private. What is relevant is if one of the SIP devices participating in the session is in a public address space and the other in a private.

In this case some observations should be made:

o A SIP device in a private address space can only communicate with a device in a public address space if a NAT binding from private to public address is provided.

o If a SIP device is in a private address space behind a legacy NAT device and implements a NAT traversal method [8], media relay might be needed for the successful establishment of the session. Media relay is most commonly implemented by a B2BUA or SBC. A legacy NAT is one that does not implement a SIP Application Level Gateway (ALG).[4]

7. Considerations on Media Flows

7.1. Decomposition

The scenarios in the previous sections show media flowing between the endpoints involved in the SIP session, but it is important to understand that the domains involved in peering might not carry the media associated with such sessions.

Media associated with the sessions established across the peering interface could be carried by a traditional ISP. The picture below depicts such a scenario.

```
                                 +---------+    _________
                            /--->|   VSP   |<-----\
                           /     |         |       \
                          |      +---------+       |
                Signaling |                        | Signaling
                (peering) |                        | (peering)
                          |                        |
         +------------+   |                        |   +------------+
         |            |<-/      +-----------+_      \->|            |
         |            |         |           |          |            |
         | Domain A   |>------->|    ISP    |>--------->| Domain B   |
         |            |<-------<|___________|<---------<|            |
         |____________|  Media  |           |   Media  |____________|
         +------------+         +-----------+          +------------+
```

7.2. Media Relay

In the event that a calling and/or called entity are part of a private network and the NAT/FW at the CPE is VoIP unaware or the client uses a NAT traversal method, the SIP proxy must find a way to modify the private addresses that remain in the signaling payload (in addition to threading media through the NAT/FW). This modifying process is sometimes referred to as Far-end NAT Traversal (FE-NTRV).

The core of the FE-NTRV process is media relaying. The signaling entity relays media between the two endpoints as a result of the repairing process and to guarantee NAT/FW traversal (symmetric RTP).

It is important to understand that media relay can be use independent of NAT/FW as a way to direct media to a certain device for

processing. In the context of SPEERMINT, media relay could be used to enable the collapsed model and/or perform FE-NTRV.

```
ALICE                    NAT/FW               Media Relay            Bob
10.10.1.2                                Signaling:128.16.5.10    192.32.6.2
                                            Media:168.12.1.8
   |                   |                     |                   |
   |     INVITE        |                     |                   |
   |------------------>|      INVITE         |                   |
   |s:10.10.1.2:9082   |-------------------->|      INVITE       |
   |d:128.16.5.10:5060 |s:140.1.1.1:23040    |------------------>|
   |c= 10.10.1.2       |d:128.16.5.10:5060   |s:128.16.5.10:5060 |
   |m= 11032           |c= 10.10.1.2         |d:192.32.6.2:5060  |
   |                   |m= audio 11032       |c= 168.12.1.8      |
   |                   |                     |m= audio 3600      |
   |                   |                     |                   |
                                             |
                                             v
  +----------------------------------------------------------------------+
  | Media Relay creates  a pair of media relay ports. The first port,    |
  | 3600, is for receiving media from the called party and the 2nd       |
  | port, 7600, is for receiving media from the calling party. As we do |
  | not know what the transport address of the calling party will be     |
  | (post NAPT), any media received from the called party must be        |
  | dropped.                                                             |
  +----------------------------------------------------------------------+

   |                   |                     |      200 OK       |
   |                   |      200 OK         |<------------------|
   |      200 OK       |<--------------------|s:192.32.6.2:5060  |
   |<------------------|s:128.16.5.10:5060   |d:128.16.5.10:5060 |
   |s:128.16.5.10:5060 |d:140.1.1.1:23040    |c= 192.32.6.2      |
   |d:10.10.1.2:9082   |c= 168.12.1.8        |m= audio 9080      |
   |c= 168.12.1.8      |m= audio 7600        |                   |
   |m= audio 7600      |                     |                   |
   |                   |                     |                   |
   |                   |                     |                   |
                                             |
                                             V
                            +-----------------------------+
                            | Media Relay updates remote |
                            | dest. as 192.32.6.2:9080   |
                            +-----------------------------+

   |                   |                     |                   |
   |     ACK (...)     |                     |                   |
   |------------------>|                     |                   |
   |                   |                     |       Media       |
   |                   |                     X<==================|
   |                   |                     |s:168.12.1.8:3600  |
```

```
   |                   |                    |d:192.32.6.2:9080  |
   |                   |                    |                   |
   |                   |                    v                   |
                                         Discarded
   |                 Media                  |                   |
   |=======================================>|==================>|
   |s:10.10.1.2:11032  |s:140.1.1.1:16220   |s:168.12.1.8:3600  |
   |d:168.12.1.8:7600  |d:168.12.1.8:7600   |d:192.32.6.2:9080  |
   |                   |                    |                   |
   |                   |                    v                   |
                               +---------------------------+
                               | Update remote destination |
                               +---------------------------+
   |                   |                    |                   |
   |                 Media                  |                   |
   |<=======================================|<==================|
   |s:168.12.1.8:7600  |s:168.12.1.8:7600   |s:192.32.6.2:9080  |
   |d:10.10.1.2:11032  |d:140.1.1.1:16220   |d:168.12.1.8:3600  |
   |                   |                    |                   |
   |                   |                    |                   |
   |                   |                    |                   |
   |                   |                    |                   |
   |                   |                    |                   |
   |                   |                    |                   |
```

7.3. Media QoS

Media flows for real time communication usually need strict scheduling guarantees in order to not degrade the service. The problem of QoS within an independent administratively managed domain and across independent domains is quite different.

In the case of L5 peering several issues arise around QoS for media flows, especially in the case of on-demand peering. Some of these issues are listed below.

- o How to reconcile general QoS parameters used in domain A across the peering interface with those announced by domain B's peering policy?

- o How domain B can identify media flows crossing the peering interface coming from domain A (and vice-versa) in order to provide the agreed upon QoS treatment? We could potentially be talking about hundreds of calls (and consequently new media flows) per second.

o Moreover, in a decomposed scenario, how the SIP proxy can let the router know the identity of such media flows and the QoS parameters associated with it? This problem was discussed under the TISPAN umbrella related to NGN networks [6].

o Alternatively or in conjunction with dynamic identification there is the issue of trust. Possibly domain B could trust domain A to mark all media packets appropriately. Domain B would honor such markings and give the appropriate treatment announced on its peering policy

8. Considerations on Multilateral Peering

Some of the difficulties discussed in previous sections would be aggravated in the case of multilateral on-demand peering where potentially more than one VSP could carry signaling (and possibly media) to reach a specific endpoint.

How could peer policies be compared to find out the best one for a specific case? In the case of routing protocols a combination of metrics, route filtering, and other techniques provide a solution.

```
                        +---------+
                        |         |
                      / | Domain  |\
                     /  |    C    | \
                    /   |         |  \
                   /    +---------+   \
                  /                    \
   +--------+    /      +---------+     \     +--------+       +-----+
   |        |---/       |         |      \----|        |       |     |
   | Domain |           | Domain  |           | Domain |       | UA  |
   |   A    |-----------|    B    |-----------|    D   |-------|     |
   |        |           |         |           |        |       +-----+
   +--------+           +---------+           +--------+
```

9. SIP Priority and SPEERMINT QoS

There are various QoS aspects that need to be taken into account in the context of SPEERMINT. These contexts include, but are not limited to, Signaling and Media QOS. The following subsections discuss those aspects by first laying out some groundwork and then going through scenarios.

9.1. Problem Statement

When SIP signaling and media packets from UA 1 arrive at peering point destined to UA 2, the Layer 5 peering functions need to make sure those packets receive the proper treatment when crossing the peering fabric into another domain. Proper treatment involves three aspects: packet recognition and marking, accounting, and trust. The scope of resource allocation to ensure predictive per hop behaviors, or QOS, is a matter of local policy within an administrative domain. More often than not, a SIP Session traverses multiple administrative domains. A subset scope of QOS local policies can be shared within a direct or transit peering arrangement.

9.2. Packet Recognition and Marking

If the layer 5 peering devices (referred to as SIP proxies) are going to mark signaling and media packets, they need to first be able to recognize them. Recognizing and marking SIP signaling is not problematic since we assume the Layer 5 peering devices perform SIP proxy functions. The primary source of confusion is in the recognition and marking of the media (RTP, etc) packets.

If the SIP Proxy performs SDP inspection, it will be able to recognize media packets based on the contents of the c and m lines. It is important to notice that there is an implicit assumption of what will be negotiated, and also that this proxy stays in the signaling call flow for the duration of the call - and therefore be aware of mid-call events.

Now we come to the problem of which device will mark the packets. In a decomposed scenario, the SIP proxy needs to let the router know how to identify media packets and which marking to use. One possible solution is the use of a Gate Control Protocol [6].

9.2.1. Peering Classes of Service

In the simplest case the peering fabric will have a set of classes of service that serve as a translation table from one domain to another. So, a domain A only needs to know how to map the classes of service used internally to the ones used in the peering point (and vice-versa).

In the simplest case the peering fabric will have a set of classes of service that serve as a translation table from one domain to another. So, domain A only needs to know how to map the classes of service used internally to the ones used in the peering point (and vice-versa). This could be independent or above and beyond any QoS policy exchanges. We should read "a packet received with an EF DSCP should be marked with AF41".

```
   -------------------------
   | Ingress |   Egress    |
   |  DSCP   |   DSCP      |
   |  name   |   name      |
   +=========+=============+
   |   EF    |   AF41      |
   +---------+-------------+
   |  CS5    |    CS5      |
   +---------+-------------+
   |AF41,AF42|   AF41      |
   |  AF43   |             |
   +---------+-------------+
   |  CS4    |    CS4      |
   +---------+-------------+
```

In the transit VoIP peering model, in order to maintain some consistency with classification of packets, there needs to be a common denominator for originating and terminating domains to understand. This only pertains to a transit peering model as a direct peering strategy does not have an abstracting 3rd party to the ultimate terminating domain.

```
  _Origin. Domain__,___Transit Domain_,_Termin. Domain___
 |                 |                  |                 |
 |     EF          |    "Highest"     |      EF         |
 |                 |                  |                 |
 |     AF1         |    "High"        |      AF         |
 |                 |                  |                 |
 |     AF2         |    "Medium"      |      AF2        |
 |                 |                  |                 |
 |     BE          |    "Low"         |      BE         |
 |                 |                  |                 |
 |     BE          |    "Who Cares?"  |      BE         |
 |_________________|__________________|_________________|
```

9.2.2. Network Address Translation (NAT)

The use of NAT media makes packet recognition problem more severe. As discussed in section 6, in certain scenarios the identification of the media flows require special processing.

9.3. Accounting

Accounting refers to the tracking consumption of network resources by sessions. In order to accomplish inter-domain accounting, it is required to know the exchanged policies, resources available and reachability. Within the information gathered, it is important to know the identity of the session, the nature of the service delivered, when the service began, and when it ended.

9.4. Trust

If Proxy 1 trusts that its users will mark packets correctly, the issue of packet recognition and marking can be mitigated. Of course that does not imply that Proxy 2 trusts Domain 1 to mark packets correctly. That is where a QoS policy exchange comes into play.

10. SIP Policy Enforcement and Definition

Within the following description, there is an assumption that the SIP proxy will know via policy exchange, variables that will weigh potentially in routing decisions from Proxy A to Proxy B, (e.g. defined relationship, trust established, etc).

In the inter-domain exchange of SIP signaled real-time sessions, the SIP proxy will be the policy decision point that enforces exchanged session policies. In this signaling plane enforcement model, all bearer traffic will receive the same level of QoS (e.g. EF). Real-time traffic (voice, video, etc) share the same sensitivity to latency, jitter, and packet loss. Therefore any direct inter-domain QoS mapping of service levels is not needed. Should one type of traffic (Video) have more significance than another (voice) then the SIP proxy will enforce that policy, possible preempting existing sessions if required.

In both the collapsed and decomposed inter-domain call models, the SIP proxies of both the originating and terminating domains have the authority to permit, deny, preempt and throttle sessions. Inspecting and classifying at the SIP layer brings an added differentiation superseding Layer 3 policies.

10.1. Local SIP Policy

Local SIP Policy is defined as that which has local significance, or not appropriate to exchange beyond administrative domains. Examples of local policies would be preferential treatment of sessions based on hierarchical subscriber groupings ("gold level" subscribers), path selection based on time of day, or presence.

10.2. Remote SIP Policy

Remote SIP policy is defined as policies that are learned via exchange mechanism with a peer in a remote administrative domain. Examples of a remote policy would be the preferred codec, or number of sessions permitted.

10.3. SIP Proceed Policy

The SIP Proceed policy is used to determine if a session attempt should be permitted to continue or not. The SIP Proceed policy is constructed from a merging of local and remote policies learned via an exchange mechanism. Permitting of a session to proceed or not can be done by any SIP proxy involved in inter-domain signaling of the session.

The need for a scalable/fast implementation that will track current state information in real-time can be achieved by RADIUS. A real-time and historical session activity database will have a full history of all active sessions. When a session attempt is made from an UA, it will be accounted for on a session accounting element of the SIP proxy. The accounting element(s) can maintain data on whatever criteria pertinent to track (codec, domain, timestamps etc...) When a new session attempt is made, an accounting look-up is done and a search on whatever criteria of interest is done to determine if session signaling can proceed. See the Policy Decision Point (PDP) in the following call flow:

```
            Alice   Proxy 1     Radius     Proxy 2      Bob
            |        |           |          |          |
            |INVITE  |           |          |          |
            |------->|           |          |          |
            | 100    |           |          |          |
            |<-------| Session   |          |          |
            |        | Attempt   |          |          |
            |        |---------->|          |          |
            |        | Current   |          |          |
            |        | Peer Stats|          |          |
            |        |<----------|          |          |
            |      (PDP)         |          |          |
            |        | INVITE    |          |          |
            |        |--------------------->| INVITE   |
            |        | 100       |          |--------->|
            |        |<---------------------|180       |
            |        | 180       |          |<---------|
            | 180    |<---------------------|          |
            |<-------|           |          |200       |
            |        | 200       |          |<---------|
            | 200    |<---------------------|          |
            |<-------|           |          |          |
            | ACK    |           |          |          |
            |------->| ACK       |          |          |
            |        |--------------------->|ACK       |
            |        |           |          |--------->|
            |        |           |          |          |
```

SIP proxies talk to a list of radius servers for accounting purposes. The radius servers should be on a local network to the proxy.

Prior to Proxy 1 sending INVITE to Proxy 2, a determination will be made based on the exchanged policies if the attempt at session establishment should be permitted.

11. Peering Domain Information Exchange

11.1. Domain Routes

In some cases, it may be required to exchange specific domain route information between peers. The following describes a method for a relationship between proxies in domains A and B to exchange domain routes using a SIP peering policy event package. This event package may contain specific sections, which will provide routing information

for the peering proxy server to update its routing table with new peering routes. This method utilizes a SUBSCRIBE method, and routes may be updated through expiry timers and subscription refreshes as defined in [8].

```
             Proxy 1                      Proxy 2
               |                             |
               |Subscribe w/PeerPlcyEvtPkg   |
               |---------------------------->|
               |   401 Unauthorized          |
               |<----------------------------|
               |Subscribe w/Auth             |
               |---------------------------->|
               |    202 Accepted             |
               |<----------------------------|
               |    Notify                   |
               |<----------------------------|
               |          200 OK             |
               |---------------------------->|
```

11.2. Authentication Credentials

In some cases, authorization credentials for authentication methods such as HTTP digest may want to be exchanged and utilized by domain proxies for authenticating new message requests from subscribers intended for a UA in another domain. The following describes a method for a relationship between proxies in domains A and B to exchange authentication information using a SIP peering policy event package. This event package may contain specific sections, which will provide authentication methods to be used for authenticating to the peer's proxy. This method utilizes a SUBSCRIBE method similar to the method described in section 3.2.

```
                   Proxy 1                     Proxy 2
                     |                             |
                     |Subscribe w/ PeerPlcyEvtPkg  |
                     |---------------------------->|
                     |   401 Unauthorized          |
                     |<----------------------------|
                     |Subscribe w/Auth             |
                     |---------------------------->|
                     |    202 Accepted             |
                     |<----------------------------|
                     |    Notify                   |
                     |<----------------------------|
                     |          200 OK             |
                     |---------------------------->|
```

12. Peering Message Flow Phases

The message flow phases are Discovery, Policy Exchange, Security Establishment, Signaling Exchange, and Media Exchange. The following flow provides an overview of the phases. Each of the phases is described individually in the following subsections. In the following flow, the policy and peering proxy have been combined; however, these two functions may be separated. Also, the signaling and media exchange phase descriptions have been omitted for clarity purposes, because their functionality has not changed for the purposes of peering. However, they have been explained further in the following subsections.

```
          Alice      Peer Proxy         DNS   Peer Policy/Proxy     Bob
            |            |                |            |              |
            |INVITE      |                |            |              |
            |----------->|                |            |              |
            |         100|                |            |              |
            |<-----------|                |            |              |
                         |NAPTR Query     |            |              |
                   +---->|--------------->|            |              |
                   |     |     NAPTR Reply|            |              |
   Discovery Phase |     |<---------------|            |              |
   ----------------|     |SRV Query       |            |              |
                   |     |--------------->|            |              |
                   |     |       SRV Reply|            |              |
                   +---->|<---------------|            |              |
                         |                |            |              |
                         | INVITE                      |              |
                         |---------------------------->|              |
```

```
                         |            401 Unauthorized |              |
                         |<----------------------------|              |
                         |                             |              |
                         | SUBSCRIBE                   |              |
                   +---->|---------------------------->|              |
                   |     |                202 Accepted |              |
   Policy Exchange |     |<----------------------------|              |
   ----------------|     |                    Notify   |              |
        Phase      |     |<----------------------------|              |
                   |     | 200 OK                      |              |
                   +---->|---------------------------->|              |
                         | INVITE                      |              |
                   +---->|---------------------------->|              |
                   |     |        [TLS Connection]     |              |
    Security Exch. |     |<--------------------------->|              |
   ----------------|     |            401 Unauthorized |              |
         Phase     |     |<----------------------------|              |
                   |     | INVITE                      |              |
                   +---->|---------------------------->|INVITE        |
           |             |                 100 Trying  |------------->|
           |             |<----------------------------|   180 Ringing|
           |             |                180 Ringing  |<-------------|
           |  180 Ringing|<----------------------------|       200OK  |
           |<------------|                     200OK   |<-------------|
           |       200OK |<----------------------------|              |
           |<------------|                             |              |
           | ACK         |                             |              |
           |------------>| ACK                         |              |
           |             |---------------------------->|ACK           |
           |             |                             |------------->|
           |             |      Both Way RTP Media     |              |
           |<==========================================================>|
           |             |                             |          BYE |
           |             |                      BYE    |<-------------|
           |          BYE|<----------------------------|              |
           |<------------|                             |              |
           | 200OK       |                             |              |
           |------------>| 200OK                       |              |
           |             |---------------------------->|200OK         |
           |             |                             |------------->|
```

12.1. Discovery Phase

The first phase of static or dynamic peering requests is discovery. The discovery process can be summarized by querying the Location Function to determine the next phase in the message flow. The discovery phase can take place via a local or external federation location function. Examples of the function may be comprised of an ENUM/DNS or redirect server. After the discovery phase has completed, the peering process will progress to a subsequent phase, usually the policy or authentication phase. The following message flows provide examples of the discovery phase.

Discovery phase utilizing an ENUM/DNS server as a location function:

```
               Alice      Peer Proxy         DNS          Peer Proxy
                 |            |                |               |
                 |INVITE      |                |               |
                 |----------->|                |               |
                 |         100|                |               |
                 |<-----------|                |               |
                              |NAPTR Query     |               |
                        +---->|--------------->|               |
                        |     |     NAPTR Reply|               |
        Discovery Phase |     |<---------------|               |
       -----------------|     |SRV Query       |               |
                        |     |--------------->|               |
                        |     |       SRV Reply|               |
                        +---->|<---------------|               |
                              |INVITE                          |
                              |------------------------------->|
```

Discovery phase utilizing a REDIRECT server as a location function:

```
                   Peer Proxy    Federation Proxy   Peer Proxy
                       |                |               |
                       |   INVITE       |               |
                 +---->|--------------->|               |
 Discovery Phase |     |     302        |               |
-----------------|     |<---------------|               |
                 |     |     ACK        |               |
                 +---->|--------------->|               |
                       |     INVITE                     |
                       |------------------------------->|
```

12.2. Policy Exchange Phase

Since the originating peer proxy does not know if the destination AOR is a PF or a SF, it must progress with a normal dialog request with the assumption it is a SF. In the event a request fails due to an authentication failure (401 Unauthorized), and no known authentication credentials exist or no longer appear to be working, the requesting proxy may issue a SUBSCRIBE [8] request to the attempted peer's AOR received through the discovery phase. The SUBSCRIBE request should be a request to attain a, currently, undefined peering policy event package. In some cases, the requesting proxy already knows it must attain the peering policy event package, and may forego the initial INVITE attempt and issue a SUBSCRIBE request instead. Once this phase is completed, after extracting and following any specific received policies, the authentication phase is attempted as the policy permits or requires. The following message flow provides an example of the policy exchange phase. The following message flow assumes the discovery phase has already completed using one of the methods described in section 12.1.

```
                   Peer Proxy                      Policy Server
                       |                                 |
                       | INVITE                          |
                       |-------------------------------->|
                       |               401 Unauthorized  |
                       |<--------------------------------|
                       |                                 |
                       | SUBSCRIBE                       |
                 +---->|-------------------------------->|
                 |     |                   202 Accepted  |
 Policy Exchange |     |<--------------------------------|
-----------------|     |                       Notify    |
      Phase      |     |<--------------------------------|
                 |     | 200 OK                          |
                 +---->|-------------------------------->|
                       | INVITE                          |
                       |-------------------------------->|
```

12.3. Security Establishment Phase

The security establishment phase follows the described methods in previous sections of this document. After the originating proxy receives the policy event package, it extracts the necessary security policy information. The security policy may contain many different combinations of security requirements. For example, it may contain a simple digest authentication method or may require TLS with digest

authentication. This is determined by the destination peer, and must be followed to successfully complete this phase. This phase follows standard methods described in [2], so the following flow provides an example of this phase, but does not incorporate all possibilities. This phase assumes the previous phases were successfully completed or purposefully omitted per peering implementation.

```
                   Peer Proxy                        Peer Proxy
                       |                                 |
                       | INVITE                          |
                 +---->|-------------------------------->|
                 |     |        [TLS Connection]         |
   Security Exch. |     |<------------------------------->|
  ---------------|     |               401 Unauthorized |
       Phase     |     |<--------------------------------|
                 |     | INVITE                          |
                 +---->|-------------------------------->|
                       |                     100 Trying |
                       |<--------------------------------|
                       |                    180 Ringing |
                       |<--------------------------------|
```

12.4. Signaling Exchange Phase

The signaling exchange phase is a necessary step to progress towards establishing peering. This phase may incorporate the security exchange phase, but it is not required. This phase follows standard methods described in [2], so the following flow provides an example of this phase, but does not incorporate all possibilities.

```
                   Peer Proxy                        Peer Proxy
                       |                                 |
                       | INVITE                          |
                 +---->|-------------------------------->|
                 |     |        [TLS Connection]         |
                 |     |<------------------------------->|
                 |     |               401 Unauthorized |
                 |     |<--------------------------------|
                 |     | INVITE                          |
                 |     |-------------------------------->|
  Signaling Exch. |     |                     100 Trying |
 ----------------|     |<--------------------------------|
       Phase     |     |                    180 Ringing |
                 |     |<--------------------------------|
                 |     |                         200 OK |
                 |     |<--------------------------------|
                 |     | ACK                             |
```

```
                    |     |------------------------------->|
                    |     |                            BYE |
                    |     |<-------------------------------|
                    |     | 200 OK                         |
                    +---->|------------------------------->|
                          |                                |
```

12.5. Media Exchange Phase

The media exchange phase is negotiated and established during the signaling exchange phase. This phase follows standard methods described in [2], so the following flow provides an example of this phase, but does not incorporate all possibilities.

```
                      Alice      Peer Proxy          Peer Proxy      Bob
                  +---->|INVITE      |                   |            |
                  |     |----------->| INVITE            |            |
                  |     |        100 |------------------>|            |
                  |     |<-----------|  [TLS Connection] |            |
                  |     |            |<----------------->|            |
                  |     |            |  401 Unauthorized |            |
                  |     |            |<------------------|            |
     Media Exch.  |     |            | INVITE            |            |
   ---------------|     |            |------------------>|INVITE      |
        Phase     |     |            |        100 Trying |----------->|
                  |     |            |<------------------|        180 |
                  |     |            |       180 Ringing |<-----------|
                  |     |        180 |<------------------|        200 |
                  |     |<-----------|            200 OK |<-----------|
                  |     |        200 |<------------------|            |
                  |     |<-----------|                   |            |
                  |     |ACK         |                   |            |
                  +---->|----------->|ACK                |            |
                        |            |------------------>|ACK         |
                        |            |                   |----------->|
                        |             Both Way RTP Media |            |
                        |<=========================================>|
                        |            |                   |            |
                        |            |                   |        BYE |
                        |            |               BYE |<-----------|
                        |         BYE|<------------------|            |
                        |<-----------|                   |            |
                        |200         |                   |            |
                        |----------->|200                |            |
                        |            |------------------>|200         |
                        |            |                   |----------->|
```

13. Security Considerations

The level of security required during the establishment and maintenance of a SIP peering relationship between two proxies can vary greatly. In general all security considerations related to the SIP protocol are also applicable in a peering relationship.

If the two proxies communicate over an insecure network, and consequently are subject to attacks, the use of TLS or IPSec would be advisable.

If there is physical security and the proxies are co-located, or the proxies are situated in a segregated network (such as a VPN), one could argue that basic filtering based on IP address is enough.

14. IANA Considerations

N/A

15. Conclusions

The purpose of this draft is to show SPEERMINT message flows but also to raise awareness through questions and detailed considerations of several issues the industry might have to deal with in different peering scenarios.

16. Acknowledgments

Thanks to Otmar Lendl for the Federation Call flows

17. References

17.1. Normative References

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

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP:Session Initiation Protocol", RFC 3261, June 2002.

[3] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[4] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[5]   Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", BCP 75, RFC 3665, December 2003.

[6]   ETSI TS 102 333: " Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Gate control protocol".

[7]   Babiarz, J., "Configuration Guidelines for DiffServ Service Classes", draft-ietf-tsvwg-diffserv-service-classes-02 (work in progress), February,2006.

[8]   Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[9]   Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

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

17.2. Informative References

[11] Meyer, D., "SPEERMINT Requirements and Terminology", Internet Draft,   draft-ietf-speermint-reqs-and-terminology-01

[12] Boulton, C., Rosenberg, J., Camarillo, G., "Best Current Practices for NAT Traversal for SIP", Internet Draft, draft-ietf-sipping-nat-scenarios-04

[13] Lendl, O., "The Domain Policy DDDS Application", draft-lendl-domain-policy-ddds-00 (work in progress), February 2006.

[14] Lendl, O., "A Federation based VoIP Peering Architecture", draft-lendl-speermint-federations-02 (work in progress), August 2006.

[15] Camarillo, G., Penfield, R., Hawrylyshen, A., "Requirements from SIP (Session Initiation Protocol) Session Border Controller Deployments", Internet Draft, draft-camarillo-sipping-sbc-funcs-03

Author's Addresses

Daryl Malas
Level 3 Communications LLC
1025 Eldorado Blvd.
Broomfield, CO 80021
USA
EMail: daryl.malas@level3.com

Sohel Khan, Ph.D.
Technology Strategist
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
U.S.A
Email: Sohel.Q.Khan@sprint.com

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA
USA
Email: rpenno@juniper.net

Adam Uzelac
Global Crossing
1120 Pittsford Victor Road
PITTSFORD, NY 14534
USA
Email: adam.uzelac@globalcrossing.com

Intellectual Property Statement

Acknowledgment

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