MMUSIC   B. Stucker
Internet-Draft   Unaffiliated
Intended status: Informational   H. Tschofenig
Expires: August 2, 2012   Nokia Siemens Networks
                                                        G. Salgueiro
                                                        Cisco Systems
                                                     January 30, 2012

Analysis of Middlebox Interactions for Signaling Protocol Communication along the Media Path
draft-ietf-mmusic-media-path-middleboxes-04.txt

Abstract

   Middleboxes are defined as any intermediary box performing functions
   apart from normal, standard functions of an IP router on the data
   path between a source host and destination host. Two such functions
   are network address translation and firewalling.

   When Application Layer Gateways, such as SIP entities, interact with
   NATs and firewalls, as described in the MIDCOM architecture, then
   problems may occur in the transport of media traffic when signaling
   protocol interaction takes place along the media path, as it is the
   case for recent key exchange proposals (such as DTLS-SRTP). This
   document highlights problems that may arise. Unfortunately, it is
   difficult for the end points to detect or predict problematic
   behavior and to determine whether the media path is reliably
   available for packet exchange.

   This document aims to summarize the various sources and effects of
   NAT and firewall control, the reasons that they exist, and possible
   means of improving their behavior to allow protocols that rely upon
   signaling along the media path to operate effectively.

Status of this Memo

material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2, 2012.

Copyright Notice

Table of Contents

1. Introduction

According to by RFC 3234 [RFC3234] middleboxes are defined as any
intermediary box performing functions apart from normal, standard
functions of an IP router on the data path between a source host and
destination host.

In the context of SIP a SIP ALG may interact with a node along the
media path to control network address translation, firewalling, and
other functions.

   With firewall control packet filters are installed based on the
   SIP signaling interaction to implement a behavior of 'deny by
   default' in order to reduce the risk of unwanted traffic. This
   function is often referred to as 'gating'. Depending on the
   timing of the packet filter installation and the content of the
   packet filter signaling traffic along the media, such as DTLS-SRTP
   or ICE, may be treated in an unexpected way.

   In cases where the middlebox is involved in overcoming unmanaged
   NAT traversal the case is similar. The key feature of this type
   of NAT traversal is a desire to overcome the possible lack of
   information about any [RFC4787] address and/or port mapping by a
   possibly unknown NAT device (server reflexive address and
   filtering properties). In particular, a NAT binding for an
   endpoint may not exist yet for the address and port identified in
   the endpoint's SDP. As such, a pilot packet sent by that endpoint
   behind the NAT is required to create the necessary mappings in the
   NAT for the media relay to deliver media destined for that
   endpoint. Until that pilot packet is received no media packets
   may be reliably forwarded to the endpoint by the relay.

This document presents a summary of these two techniques, discusses
their impact upon other protocols such as ICE and DTLS-SRTP, and
proposes a set of recommendations to mitigate the effects of gating
and latching on in-band negotiation mechanisms.

2. Terminology

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

We use the terms filter, policy action (or action), policy rule(s),
MIDCOM agent, and MIDCOM Policy Decision Point (PDP) as defined in
[RFC3303]. The MIDCOM agent is co-located with a SIP ALG that
communicates with the firewall or the media relay.

3. Architecture

   Figure 1 shows the architecture that is being considered in this
   document with respect to firewall and NAT traversal using media
   relaying. The timing and directionality with which media packets are
   allowed to traverse a particular edge device is the subject of this
   investigation. The MIDCOM agent thereby pushes policy rules to the
   middlebox that allow or deny certain flows to bypass. Additionally,
   in case of media relaying it is important for the MIDCOM agent to
   adjust the signaling messages.

```
                 SIP      +-----------------+     SIP
      +-----+  Signaling  |    SIP ALG      | Signaling  +-----+
      | UAC |<----------->+-----------------+<---------->| UAS |
      +-----+             |   MIDCOM Agent  |            +-----+
         ^                +-----------------+               ^
         |                        ^                         |
         |         Policy rule(s) | and NAT bindings        |
         |                        v                         |
         |      Media        +-------------+      Media     |
         +----------------->|  Middlebox  |<----------------+
                            +-------------+
```
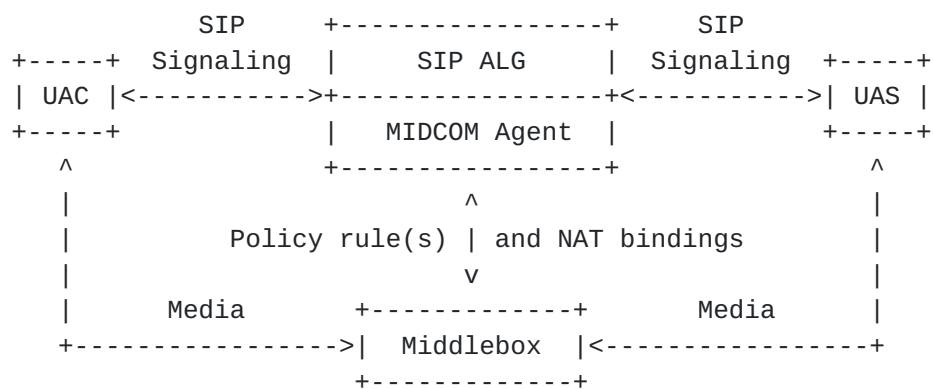
                Figure 1: Analysed Firewalling Architecture

   The aspects of packet filtering are described in Section 4 whereas
   NAT traversal is illustrated in Section 5.

4. Packet Filtering

   Figure 1 highlights the interaction between the MIDCOM agent and the
   middlebox. These two elements inspect call control signaling and
   media path packets and determine when packets from a given source to
   a given destination are allowed to flow between endpoints. It is
   common for the gate controller to be the local outbound proxy for a
   given SIP UA being gated.

   The primary responsibility of the MIDCOM agent, which is co-located
   with a SIP entity, is to examine the call control signaling to
   determine the media addresses and ports used to define the media path
   between the gated device and the endpoint(s) with which it is
   corresponding. For SIP, this would correspond to the media addresses
   described within SDP after at least one full offer/answer exchange.

   This information is used to create one or more packet filters that
   describe the expected media path(s) for the call. These packet

filters are combined with an algorithmic determination, typically
based on the state of the call, as to which direction(s) media
packets are allowed to flow between the endpoints, if at all. The
filter and the action that is being installed by the MIDCOM agent at
the middlebox may change during the lifetime of a SIP signaling
session, depending on the state of the call or on changes of the
address and port information of one (or even both) of the end points.

It is possible that the gate controller may not be able to establish
an exact address or port for one endpoint involved in the call in
which case it may wildcard the address and/or port for the source
and/or destination endpoint in the packet flow filter. In such a
case, the packet flow filter is considered to have matched against a
given media packet for the wildcarded field.

Note that it is possible to specify the filter using wildcards, for
example, if some end point address information is not known at a
given point in time. Additionally, the default firewalling policy is
subject to local configuration ('deny per default' vs. 'permit per
default'). For a given SIP signaling sessions the policy at the
MIDCOM agent might be very strict with respect to the packets that
are allowed to flow in a particular direction. For example, packets
may be allowed to flow in both directions, only in one direction for
a specific media stream. No particular behavior can be assumed.

When a media session is destroyed (end of call, deleted from the
session description, etc.), the MIDCOM agent removes policy rules
created for that media session at the middlebox.

4.1. Protocol Interaction

MIDCOM agents may employ a variety of models to determine when to
change the status of a particular policy rule. This is especially
true when a call is being established. For SIP, this would be when
an early dialog is established between endpoints. Although there is
the potential for a great deal of variability due to an intentional
lack of specification, typically, one of two models is used by the
MIDCOM agent to determine the state of a policy rule during call
setup: single-stage and two-stage commit. The term 'commit' here
refers to the point at which a policy rule is setup that allows media
traffic to flow. For example, this would be the point at which
packets for a media stream marked a=sendrecv in SDP was allowed to
flow bi-directionally by the middlebox.

4.1.1. Single-Stage Commit

Single stage commit is commonly used when the MIDCOM agent is most
involved only in firewalling. For SIP, MIDCOM agents use a single-

stage commit model typically install policy rules for the call when
the 200 OK to the INVITE is received in the case that the INVITE
contained an SDP offer, or when the ACK is received if the initial
offer was sent in the 200 OK itself.

This model is often used to prevent media from being sent end-to-end
prior to the call being established.

```
                UAC Side         MIDCOM          UAS Side
     UAC        Middlebox        Agent           Middlebox        UAS
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        |            |              |                |            |
        | (1)  INVITE + SDP Offer   |                |            |
        |-------------------------->| (2)  INVITE + SDP Offer     |
        |    c=IN IP4 47.0.0.1       |---------------------------->|
        |    m=audio 49170 RTP/AVP 0 |   c=IN IP4 47.0.0.1         |
        |    a=sendrecv              |   m=audio 49170 RTP/AVP 0   |
        |            |              |   a=sendrecv                |
        |            |              |                |            |
        |            |              |      (3) 200 OK + SDP Answer |
        |            |              |<---------------------------|
        |            |              |   c=IN IP4 47.0.0.2         |
        |            |              |   m=audio 36220 RTP/AVP 0   |
        |            |              |   a=sendrecv   |            |
        |            |              |                |            |
        |            | (5) Policy   | (4) Policy     |            |
        |            |<---------------|-------------->|            |
        |            |   Src: 47.0.0.2 | Src: 47.0.0.1 |          |
        |            |     port 36220 |   port 49170  |            |
        |            |   Dst: 47.0.0.1 | Dst: 47.0.0.2 |          |
        |            |     port 49170 |   port 36220  |            |
        |            |   sendrecv     | sendrecv      |            |
        |            |   action=permit | action=permit |          |
        |            |              |                |            |
        |            |              |                |   RTP      |
        |<=========================================================>|
        |            |              |                |            |
        |    (6) 200 OK + SDP Answer |                |            |
        |<--------------------------|                |            |
        |  c=IN IP4 47.0.0.2         |                |            |
        |  m=audio 36220 RTP/AVP 0   |                |            |
        |  a=sendrecv                |                |            |
        |            |              |                |            |
        |    (7)    ACK             |   (8)      ACK  |            |
        |-------------------------->|---------------------------->|
        |            |              |                |            |
```
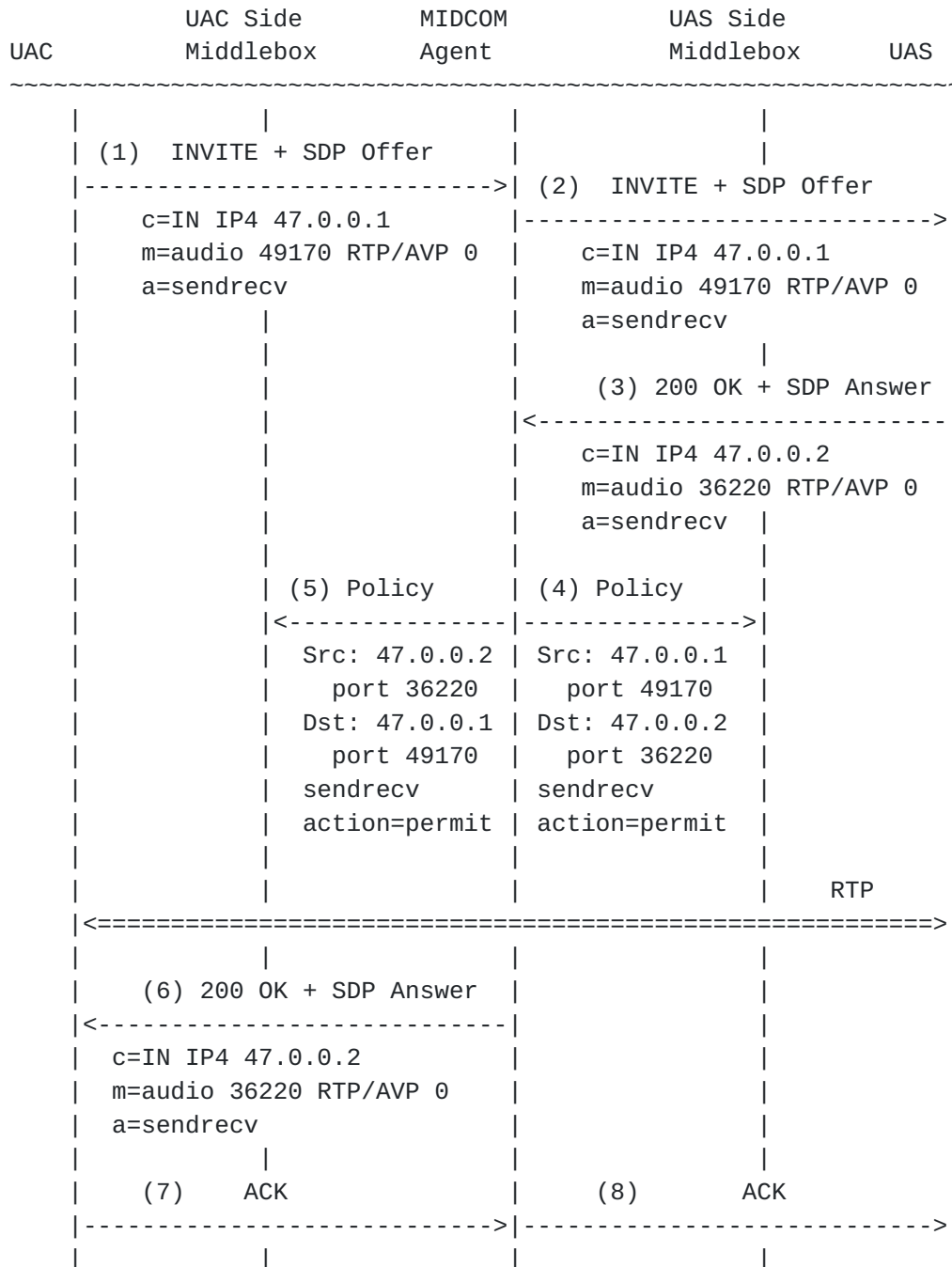
               Figure 2: Example Single-stage Commit with SIP and SDP

   In the example above, policy is created in steps 4 and 5 to allow bi-
   directional media flow based on the SDP exchanged in steps 1 and 3.
   In particular, the rules at the UAC side middlebox would indicate
   that traffic exchanged between IP address 47.0.0.1 and port number
   49170 and IP address 47.0.0.2 and port number 36220 is allowed in
   both directions.

   In this example, the MIDCOM agent installs the policies after the 200
   OK to the INVITE arrives in step 3. With a firewalling policy of
   'deny by default' media sent prior to steps 5 and 4 by the UAC or UAS
   is discarded by the middleboxes.

   Noted that early media that arrives before the 200 OK would require
   special treatment since otherwise it would be dropped as well.

4.1.2. **Two-Stage Commit**

   Two-stage commit is used when the MIDCOM agent also providers
   functionality, such as Quality of Service signaling that may require
   resources to be reserved early on in the call establishment process
   before it is known if the call will be answered. An example of this
   would be where the MIDCOM agent is responsible for guaranteeing a
   minimum level of bandwidth along the media path. In this case an
   initial set of policies may be sent by the MIDCOM agent to the
   middlebox even though they are put into a pending state but trigger a
   resource reservation. Later, when the call is accepted, the gate
   controller may update the state of the policies to active them.

```
              UAC Side        MIDCOM          UAS Side
      UAC     Middlebox       Agent           Middlebox       UAS
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        |         |              |              |             |
        | (1)  INVITE + SDP Offer |             |             |
        |------------------------->| (2)  INVITE + SDP Offer   |
        |    c=IN IP4 47.0.0.1     |-------------------------->|
        |    m=audio 49170 RTP/AVP 0 |   c=IN IP4 47.0.0.1     |
        |    a=sendrecv           |    m=audio 49170 RTP/AVP 0 |
        |         |              |    a=sendrecv               |
        |         |              |              |             |
        |         |              |    (3) 180 + SDP Answer     |
        |    (4) 180 + SDP Answer |<---------------------------|
        |<------------------------|    c=IN IP4 47.0.0.2       |
        |   c=IN IP4 47.0.0.2     |    m=audio 36220 RTP/AVP 0 |
        |   m=audio 36220 RTP/AVP 0 |   a=sendrecv             |
        |   a=sendrecv            |              |             |
        |         |              |              |             |
        |         | (5) Policy   | (6) Policy   |             |
        |         |<-------------|------------->|             |
        |         |  Src: 47.0.0.2 | Src: 47.0.0.1 |          |
        |         |    port 36220 |    port 49170 |           |
        |         |  Dst: 47.0.0.1 | Dst: 47.0.0.2 |          |
        |         |    port 49170 |    port 36220 |           |
        |         |  rule inactive | rule inactive |          |
        |         |  action=permit | action=permit |         |
        |         |              |              |             |
        |         |              |    (7)    200 OK           |
        |         |              |<---------------------------|
        |         |              |              |             |
        |         | (9) UpdateGate | (8) UpdateGate |         |
        |         |<-------------|------------->|             |
        |         |  G: sendrecv | G: sendrecv  |             |
        |         |              |              |     RTP     |
        |<=========================================================>|
        |         |              |              |             |
        |    (10)    200 OK      |              |             |
        |<------------------------|              |             |
        |         |              |              |             |
        |    (11)    ACK         |    (12)      ACK           |
        |------------------------>|-------------------------->|
        |         |              |              |             |
```
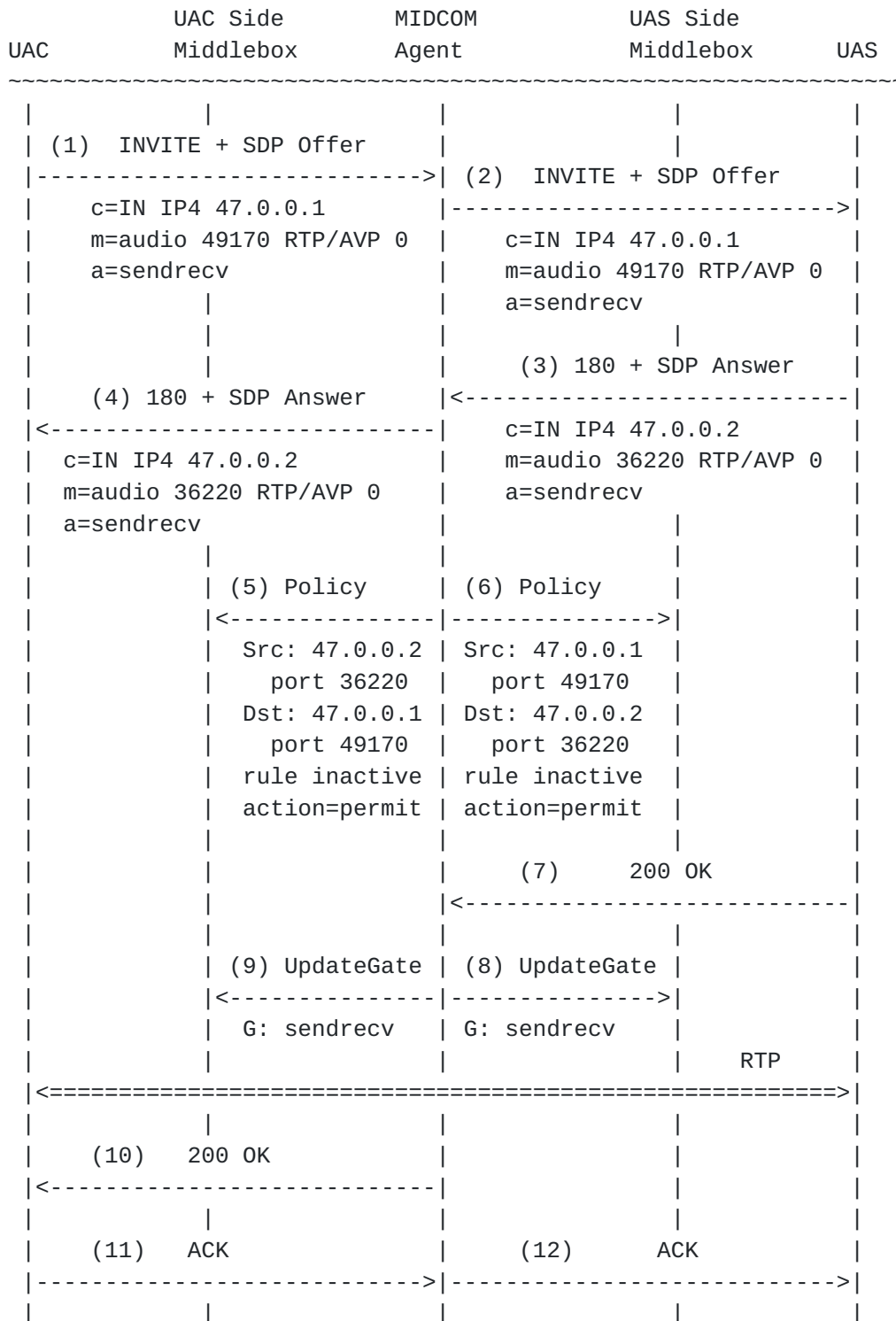
Figure 3: Example Two-stage Commit with SIP and SDP

In the example above, policies are created in steps 5 and 6 based off
of the SDP sent in steps 1 and 3 in an initial inactive state (no
packets are allowed to flow) despite the SDP indicating the media

should be bi-directional. This interaction with the middlebox,
however, triggers a QoS reservation to take place. Later, when the
200 OK to the INVITE comes in step 7, the policies are updated in
steps 8 and 9 to indicate that packets should be allowed to flow bi-
directionally. Although functionally equivalent to the single-stage
commit example given earlier in Figure 2, other operations at the
gate agent may have been performed simultaneously in steps 5 and 6
that justifies the early explicit definition of the gates in an
inactive state. The full usage of PRACK here is not shown for
purposes of brevity.

4.2. Further Reading

Packet filtering based on the approach described in this document has
been described in a number of documents. Although the usage of this
architecture can also be found on the Internet their behavior is
largely specified only in documents that relate to IMS
standardization. The behavior of the devices deployed on the
Internet is therefore largely undocumented. Nevertheless, the
following documents give the reader a better idea of the
functionality and the signaling interaction. These documents may
also specify an additional behavior in relation to how packet
filtering is used when the MIDCOM agent is responsible for processing
SIP/SDP call control signaling and the middlebox is responsible for a
variety of activities beyond pure filtering. For example, it is
common for middleboxes to exempt RTCP flows from being blocked even
though the associated RTP flows are not allowed to flow in order to
support RTCP signaling while a call is on hold. These references are
given here for the reader to gather a better understanding of how
this is mechanism is used in various forums and is non-exhaustive:

1. 3GPP, "TS 23.203: Policy and charging control architecture"
    [TS-23.203]

2. 3GPP, "TS 29.212: Policy and Charging Control over Gx reference
    point" [TS-29.212]

3. 3GPP, "TS 29.213: Policy and Charging Control signaling flows and
    QoS parameter mapping" [TS-29.213]

4. 3GPP, "TS 29.214: Policy and charging control over Rx reference
    point" [TS-29.214]

5. ETSI TISPAN, "ES 282-003: Telecommunications and Internet
    converged Services and Protocols for Advanced Networking
    (TISPAN); Resource and Admission Control Sub-system (RACS);
    Functional Architecture" [TISPAN-ES-282-003]

6. Cablelabs, "PacketCable 2.0: Quality of Service Specification
    (PKT-SP-QOS-I01-070925)" [PKT-SP-QOS-I01-070925]

Note that different terms are used for the MIDCOM agent and the
middlebox. For example, in an IMS context the MIDCOM agent would be
part of the P-CSCF and PCRF elements or in TISPAN it would be part of
the P-CSCF, A-RACF and SPDF that are involved in controlling gating
operations. Many different elements perform the role of a middlebox:
GSM GGSN, CDMA PDSN, SAE serving gateway, TISPAN PCEF and A-BGF/
C-BGF/I-BGF, PacketCable CMTS, etc. These functions may be present
in the network in a unified or decomposed architecture.

5. NAT Traversal

Two distinct types of NAT traversal can be supported by a MIDCOM
agent and the connected middlebox:

1. The MIDCOM agent and the attached middlebox act as a B2BUA at the
    border of an operator's network to protect this network and to
    perform the IP address and port conversion, which may be required
    because private address spaces are used within the network, or
    because IPv4 and IPv6 address realms are interfacing. For this
    use case, the middlebox itself performs functions similar to a
    NAT and is deployed instead of a NAT at a network border.

2. The MIDCOM agent and attached middlebox support the traversal of
    a residential NAT (also termed customer premise equipment), which
    is typically located at the user's side of an access network, for
    instance within a DSL router. The middlebox thereby acts as kind
    of media relay.

Both functions can be combined by the same MIDCOM agent and connected
middlebox, for instance by a TISPAN C-BGF.

As shown in Figure 1 the MIDCOM agent that is being co-located with
the SIP ALG functionality interacts with the middlebox that is also a
NAT in order to request and allocate NAT bindings and then modifies
the SDP offer and answer within SIP to insert the IP addresses and
port allocated by the NAT as destination for the media in both
directions. A consequence of the interaction with a (double) NAT is
that the media traffic is forced to traverse a certain NAT in both
directions (also called media anchoring). The opening of pinholes
through the middlebox is only done on request of the MIDCOM agent,
and not triggered by the detection of outbound media flows. Such
middleboxes are for instance the TISPAN A-BGF/C-BGF/I-BGF and the
3GPP IMS Access Gateway.

The functionality and control of the middlebox becomes comparable to
a media gateway and TISPAN standardized the usage of the H.248 /
MEGACO protocol for the control of the middlebox by the midcom MIDCOM
agent.

This architecture could be compared with a STUN relay [RFC5766] that
is being controlled by the MIDCOM agent rather than the end point
itself. The motivation why this technique is being used in favor to
other NAT traversal techniques is that clients do not have to support
anything beyond RFC 3261 [RFC3261] and network administrators can
control and apply local policy to the relay binding process in a
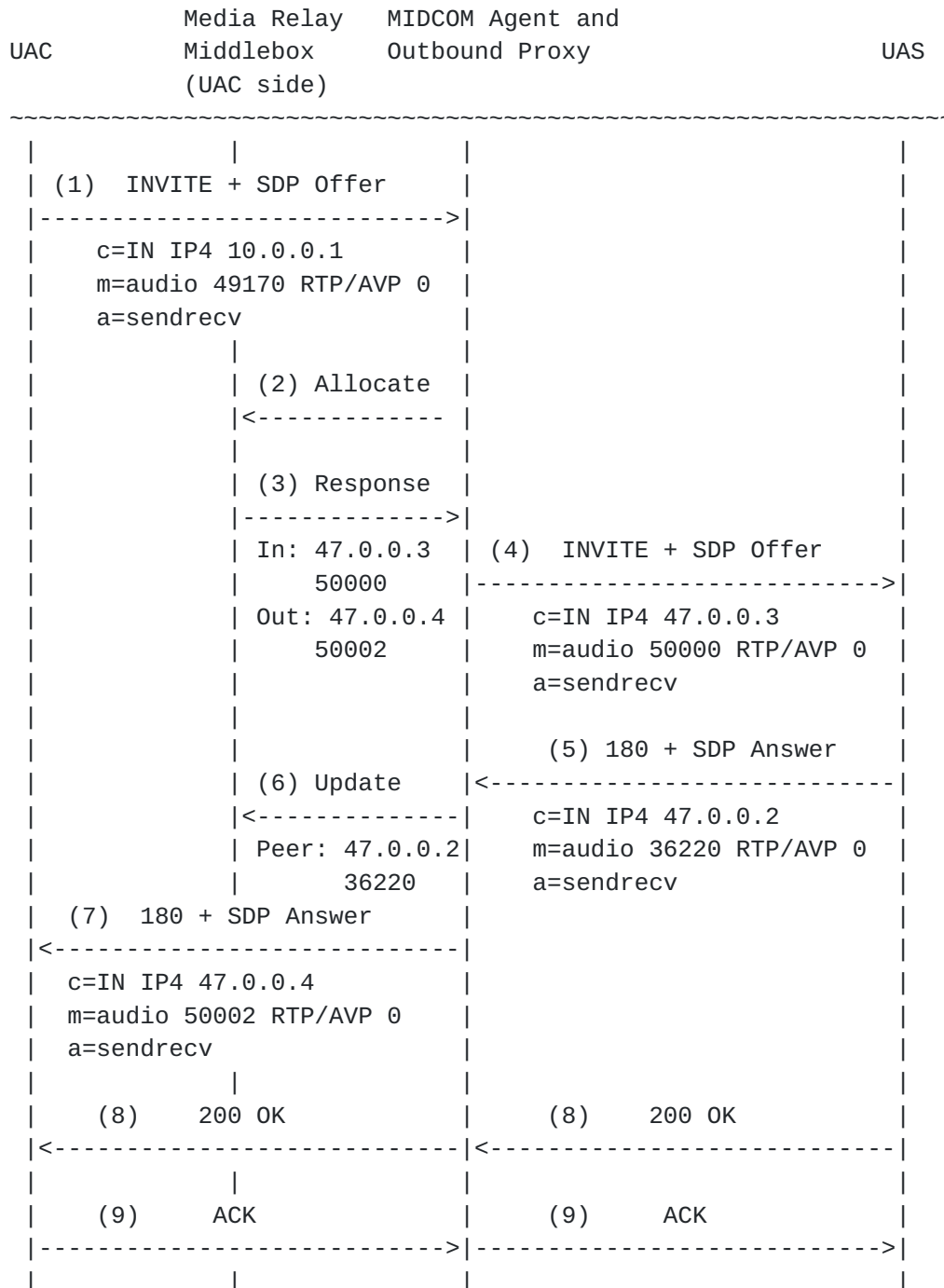centralized manner.

5.1. Protocol Interaction

The MIDCOM agent's role is to inspect call control signaling and
update media address and port values based upon media relay binding
information allocated with the middlebox/media relay. For SIP, this
minimally involves updating the c= and m= lines in the SDP, although
some implementations may also update other elements of the SDP for
various reasons.

Because the endpoints may not be able to gather a server reflexive
address for their media streams, the MIDCOM agent employs the
following algorithm to ensure that media can flow to the given
endpoint:

1. When receiving an initial SDP offer, the MIDCOM agent requests
    authorization for the request arriving at the middlebox,
    configures the middlebox to forward media between the offerer and
    the destination address / port as received in the incoming SDP
    offer, reserves a local IP address and port, and replaces the
    destination address and port from the incoming offer with the IP
    address / port used by the middlebox in the forwarded offer.

2. When receiving an initial SDP answer, the MIDCOM agent configures
    the middlebox for the corresponding session to send media towards
    the answerer towards the destination address and port as received
    in the incoming SDP answer, request the middlebox to reserve a
    local IP address / port, and exchange the destination address and
    port from the incoming answer with that middlebox IP address and
    port in the forwarded answer.

3. If the middlebox supports the traversal of residential NATs, it
    applies a technique called "media latching": The destination IP
    address of packets forwarded by the middlebox in the outbound
    direction is derived from the source IP address of packets
    received in the inbound direction. This overrides a destination

address possibly configured by the MIDCOM agent.

An example of this algorithm is shown in Figure 4 when using SIP and
SDP. In this example the UAC is the endpoint served by the MIDCOM
agent, which is also acting as a local outbound proxy, and the UAS is
the corresponding endpoint. We assume that the UAC is located behind
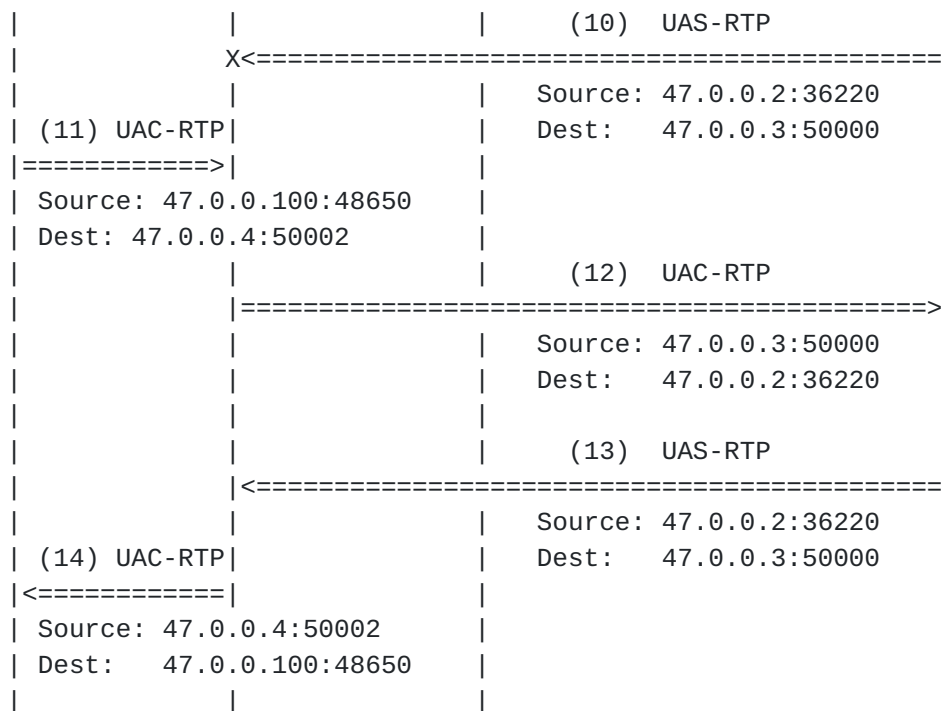a residential NAT; this NAT is, however, not shown in Figure 4.

```
                 Media Relay   MIDCOM Agent and
    UAC          Middlebox     Outbound Proxy                     UAS
                 (UAC side)
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     |              |            |                                 |
     | (1)  INVITE + SDP Offer   |                                 |
     |-------------------------->|                                 |
     |     c=IN IP4 10.0.0.1      |                                 |
     |     m=audio 49170 RTP/AVP 0 |                               |
     |     a=sendrecv            |                                 |
     |              |            |                                 |
     |              | (2) Allocate |                               |
     |              |<------------- |                              |
     |              |            |                                 |
     |              | (3) Response |                               |
     |              |------------->|                               |
     |              | In: 47.0.0.3 | (4)   INVITE + SDP Offer      |
     |              |      50000   |---------------------------->|
     |              | Out: 47.0.0.4 |   c=IN IP4 47.0.0.3          |
     |              |      50002   |   m=audio 50000 RTP/AVP 0    |
     |              |            |   a=sendrecv                    |
     |              |            |                                 |
     |              |            |    (5) 180 + SDP Answer         |
     |              | (6) Update |<----------------------------|
     |              |<-------------|   c=IN IP4 47.0.0.2          |
     |              | Peer: 47.0.0.2|   m=audio 36220 RTP/AVP 0   |
     |              |      36220   |   a=sendrecv                  |
     |   (7)  180 + SDP Answer    |                                 |
     |<-------------------------- |                                 |
     |   c=IN IP4 47.0.0.4        |                                 |
     |   m=audio 50002 RTP/AVP 0  |                                 |
     |   a=sendrecv              |                                 |
     |              |            |                                 |
     |    (8)    200 OK          |    (8)    200 OK                 |
     |<--------------------------|<----------------------------|
     |              |            |                                 |
     |    (9)    ACK             |    (9)    ACK                    |
     |-------------------------->|---------------------------->|
     |              |            |                                 |
```

```
   |               |             |      (10)  UAS-RTP           |
   |             X<=============================================|
   |               |             |  Source: 47.0.0.2:36220     |
   | (11) UAC-RTP|             |  Dest:   47.0.0.3:50000     |
   |============>|             |                             |
   | Source: 47.0.0.100:48650    |                             |
   | Dest: 47.0.0.4:50002    |                             |
   |               |             |      (12)  UAC-RTP           |
   |             |=============================================>|
   |               |             |  Source: 47.0.0.3:50000     |
   |               |             |  Dest:   47.0.0.2:36220     |
   |               |             |                             |
   |               |             |      (13)  UAS-RTP           |
   |             |<=============================================|
   |               |             |  Source: 47.0.0.2:36220     |
   | (14) UAC-RTP|             |  Dest:   47.0.0.3:50000     |
   |<===========|             |                             |
   | Source: 47.0.0.4:50002    |                             |
   | Dest:   47.0.0.100:48650    |                             |
   |               |             |                             |
```

                   Figure 4: Call Flow with SIP + SDP

   Step (1): UAC sends INVITE to local outbound proxy, which is also a
      MIDCOM agent, with an SDP offer.

   Step (2): The MIDCOM agent looks at the signaling and asks the
      middlebox to allocate a media relay binding. At this point in
      time the MIDCOM agent can only provide the IP address it finds
      inside the offer, i.e., the IP address and port where the UAC is
      expecting to receive traffic sent by the UAS. In this example the
      IP address equals 10.0.0.1 and the port number is 49170.

   Step (3): The middlebox responds with a media relay binding that
      consists of an inbound address/port for media sent by the UAS, and
      an outbound address/port for media sent by the UAC. The IP
      address and port of the middlebox allocated for the inbound side
      47.0.0.3:50000 and the address and port on the outbound side is
      47.0.0.4:50002.

   Step (4): The MIDCOM agent updates the addresses in the SDP offer
      with the inbound address/port information from the middlebox/media
      relay binding response, namely with 47.0.0.3:50000.

   Step (5): The UAS responds with a 180 containing an SDP answer.
      This answer indicates that traffic will be sent from the IP
      address and port 47.0.0.2:36220.

Step (6): The MIDCOM agent interacts with the middlebox to update
   the destination address/port information from the SDP answer for
   media to be sent to the UAS, and changes the addresses/ports in
   the SDP answer to the UAC with the outbound address/port
   information from the middlebox binding from step 3. Media can now
   flow to the UAS from the UAC at the middlebox/media relay, i.e.,
   in the outbound direction.

Step (7): The UAC receives the SDP answer containing the media relay
   outbound address/port information, namely 47.0.0.4:50002.

Step (8): The UAS answers the INVITE with a 200 OK.

Step (9): The UAC acknowledges with an ACK.

Step (10): RTP for the UAS, which may have begun flowing prior to
   answer, goes to the middlebox, but the middlebox has no reliable
   address to relay the media to for the UAC yet. Media will
   typically be dropped.

Step (11): RTP arrives at the media relay on the inbound address/
   port from the UAC. The middlebox observes the source address and
   port of the arriving packet and completes the binding process.
   The source address and port of the media from the UAC is now the
   destination address/port for media arriving on the outbound port
   of the middlebox/media relay from the UAS.

Step (12): Media originating from the UAC is relayed by the
   middlebox to the UAS.

Step (13): Media from the UAS is sent towards the middlebox.

Step (14): The middlebox forwards the media traffic to the UAC.

5.2. Further Reading

In TS 23.228 the 3GPP standardized the usage of a SIP-ALG residing in
the P-CSCF to control an IMS Access Gateway, acting as middlebox at
the interface between the IMS and the access network (see Annex G),
and the usage of a SIP-ALG residing in the IBCF to control an TrGW as
a middlebox at the interface between the IMS and external networks or
other IMS networks (see Annex I).

Although the described residential NAT traversal approach is used by
a number of implementations to overcome incorrect address/port
information in call control signaling from an endpoint behind a NAT,
only one reference is known that describes the functionality in a
standardized manner.

   1. ETSI TISPAN, "ES 282-003: Telecommunications and Internet
       converged Services and Protocols for Advanced Networking
       (TISPAN); Resource and Admission Control Sub-system (RACS);
       Functional Architecture" [TISPAN-ES-282-003]. The TISPAN Ia
       interface between the TISPAN BGF and SPDF is the relevant
       specification.

6. Interactions between Media Path Signaling and Middlebox Behavior

   This section points to the problems that occur when signaling
   exchanges are performed along the media path when middleboxes are
   present that behave in the way described in this document.

6.1. Packet Filtering

   The description in Section 4 highlighted that the timing of the
   policy rule installation by the MIDCOM agent towards the middlebox
   has an impact on when and what media traffic is allowed to traverse.

   The installation of policy rules is a prerequisite for related media
   to flow. As those policy rules are derived from information from
   both SDP offer and answer, they are typically installed at the
   completion of the first offer-answer exchange.

   Furthermore, the middlebox may prevent the exchange of packets in the
   media path after this point by closing "gates" until the session
   establishment signaling has reached a pre-configured milestone where
   the MIDCOM agent signals to the middlebox that packets are allowed to
   traverse in both directions. Prior to this, packets may be allowed
   to flow uni-directionally to satisfy certain service requirements or
   may be entirely blocked by the middlebox. For SIP [RFC3261] the
   typical milestone that must be reached is offer/answer exchange
   [RFC3264] accompanied by an acknowledgement that the dialog has been
   accepted by the UAS (i.e., 200 OK to the INVITE). It depends on the
   policy of an operator when to open gates. The policy may take into
   account the requirements of special media types to have early
   bidirectional media exchanges, e.g. if the usage of DTLS is indicated
   in SDP.

   A concrete example of the impact can be found with the case of key
   exchange along the media path, as it is provided by DTLS-SRTP. The
   ladder diagram in Section 7.1 of [RFC5763] shows that the arrival of
   the SIP INVITE at the UAS triggers the DTLS handshake. This message
   would be blocked by the middlebox, as described in Section 4 since
   the MIDCOM agent has not yet installed policy rules. The consequence
   is that the communication fails unless the UAS repeats attempts for
   an DTLS handshake until connectivity is established in both

directions by the installation of policy rules and the presence of
opened gates. Due to extra time required for the DTLS exchange the
user may experience clipping.

According to 3GPP standards, gates for RTCP are always opened when
policy rules for related media are installed, even if related media
traffic is still blocked. Therefore, signaling embedded in RTCP is
likely to pass after the completion of the first offer-answer
exchange. Standardized policy rules only inspect source and
destination information of IP packets and the transport protocol
(e.g., UDP and TCP). Obviously, this is not a property that can be
guaranteed to be true in the future.

6.2. NAT Traversal

The described NAT traversal interaction prevents asynchronous
exchange of packets in the media path until a pilot packet has been
received by the middlebox from the endpoint being served. It can be
employed for both the [RFC3264] offerer and/or answerer. Therefore,
in the worst case, both endpoints must generate a pilot packet
towards each other to ensure a bi-directional media path exists. Any
signaling on the media path that relies upon a uni-directional
handshake in the reverse direction may not complete until media in
the forward direction by the other endpoint. If signaling on the
media path is required to complete prior to media generation the
handshake may stall indefinitely.

Middleboxes as described in Section 5 will not allow any media to
pass through without being configured to do so by the MIDCOM agent
when the first offer-answer exchange is completed. Without latching,
it may be technically feasible to pass media packets from answerer
towards the offerer after the offer has passed the MIDCOM agent, but
existing implementations hardly show that behavior. Furthermore,
such middleboxes may apply gating policies similar to the policies
discussed in Section 6.1 in addition.

The described latching technique for residential NAT traversal
interaction requires that a pilot packet has been received by the
middlebox from the endpoint being served before the middlebox is able
to send packets towards the endpoint. This latching technique can be
employed for both the RFC 3264 offerer and answerer. Therefore, in
the worst case, both endpoints must generate a pilot packet towards
each other to ensure that a bi-directional media path exists. If the
first packets to be exchanged in the media path are signaling packets
and a particular directionality of those packets is required,
communication may fail. To overcome these problems, empty packets
could be sent by the endpoint that has to receive rather than to send
the first signaling message. The offer is capable of sending the

pilot packet only when receiving the destination information within
the answer. Thus, before that point in time the offerer will also
not be able to receive any media packets or related signaling.

In a similar manner as outlined in Section 6.1, any in-path signaling
messages that are sent before the offer-answer exchange is completed
will be dropped.

7. **Preliminary Recommendations**

The following preliminary recommendations are suggested:

REC #1:   It is recommended that any protocol handshake on the media
   path ensure that a mechanism exists that causes both endpoints to
   send at least one packet in the forward direction as part of, or
   prior to, the handshake process. Retransmission of STUN
   connectivity checks (see [RFC5389]) as part of ICE [RFC5245] is an
   example of such a mechanism that satisfies this recommendation.
   Sending of no-op RTP packets (see [I-D.ietf-avt-rtp-no-op]) is
   another example.

REC #2:   It is recommended that middleboxes present on the media
   path allow at least a nominal amount of traffic to be exchanged
   between endpoints after the completion of the first offer-answer
   exchange to enable the completion of media path signaling prior to
   the session being established. Such policies may be restricted to
   media types that use in-path signaling. The amount of traffic
   necessary to complete the signaling between endpoints is expected
   to be orders of magnitude smaller than that of any sufficiently
   interesting fraudulent traffic.

REC #3:   It is recommended that failure to complete signaling on the
   media path not automatically cause the session establishment to
   fail unless explicitly specified by one or more endpoints. A
   fallback scenario where endpoints retry signaling on the media
   path is recommended. Recommended points in time to retry
   signaling on the media path are after the completion of the first
   offer-answer exchange and again after the session has been
   established. Additional retries with adequate pacing may be used
   in addition.

REC #4: If signaling on the media path is required before media can
   flow, the answer should send the SDP answer as soon as possible,
   for example within a provisional SIP response, to allow the media
   path signaling to bypass middleboxes and therefore to avoid
   clipping.

REC #5: It is recommended that middleboxes present on the media path
   allow at least a nominal amount of traffic to be exchanged between
   endpoints for at least one RTT after the middlebox receives a
   message from the MIDCOM agent indicating the media session being
   terminated. This will ensure that any transit signaling packets
   on the media path exchanged during the session termination pass
   through the middlebox.

8. Security Considerations

This document talks about security related functionality and the
impact of one security mechanism, namely firewalling, to another one,
namely key management for media security.

9. IANA Considerations

This document does not require actions by IANA.

10. Acknowledgements

We would like to thank Steffen Fries, Dan Wing, Eric Rescorla, and
Francois Audet for their input to this document. Furthermore, we
would like to thank Jason Fischl, Guenther Horn, Thomas Belling,
Peter Schneider, Jari Arkko, Cullen Jennings for the discussion input
to this problem space.

We would also like to thank the participants of the IETF#70 MMUSIC
working group meeting for their feedback.

Thomas Belling provided text proposals in April 2008. We are
thankful for his detailed suggestions.

This document has benefited from the discussion and review of the
MMUSIC working group, especially the detailed review and thoughtful
comments of Peter Musgrave and Muthu Arul Mozhi Perumal.

11. References

11.1. Normative References

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

   [RFC3261] 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.

   [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
              with Session Description Protocol (SDP)", RFC 3264,
              June 2002.

   [RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and
              A. Rayhan, "Middlebox communication architecture and
              framework", RFC 3303, August 2002.

11.2. Informative References

   [I-D.ietf-avt-rtp-no-op]
              Andreasen, F., "A No-Op Payload Format for RTP",
              draft-ietf-avt-rtp-no-op-04 (work in progress), May 2007.

   [PKT-SP-QOS-I01-070925]
              CableLabs, "PacketCable 2.0: Quality of Service
              Specification", September 2007, <http://www.cablelabs.com/
              specifications/PKT-SP-QOS-I01-070925.pdf>.

   [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and
              Issues", RFC 3234, February 2002.

   [RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer
              Security", RFC 4347, April 2006.

   [RFC4787] Audet, F. and C. Jennings, "Network Address Translation
              (NAT) Behavioral Requirements for Unicast UDP", BCP 127,
              RFC 4787, January 2007.

   [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment
              (ICE): A Protocol for Network Address Translator (NAT)
              Traversal for Offer/Answer Protocols", RFC 5245,
              April 2010.

   [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,
              "Session Traversal Utilities for NAT (STUN)", RFC 5389,
              October 2008.

   [RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework
              for Establishing a Secure Real-time Transport Protocol
              (SRTP) Security Context Using Datagram Transport Layer
              Security (DTLS)", RFC 5763, May 2010.

   [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using

Relays around NAT (TURN): Relay Extensions to Session
Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[TISPAN-ES-282-003]
ETSI, "Telecommunications and Internet converged Services
and Protocols for Advanced Networking (TISPAN); Resource
and Admission Control Sub-system (RACS); Functional
Architecture", June 2006, <http://webapp.etsi.org/>.

[TS-23.203]
3GPP, "Policy and charging control architecture",
September 2007,
<http://www.3gpp.org/ftp/Specs/html-info/23203.htm>.

[TS-29.212]
3GPP, "Policy and Charging Control over Gx reference
point", June 2008,
<http://www.3gpp.org/ftp/Specs/html-info/29212.htm>.

[TS-29.213]
3GPP, "Policy and Charging Control signaling flows and QoS
parameter mapping", June 2008,
<http://www.3gpp.org/ftp/Specs/html-info/29213.htm>.

[TS-29.214]
3GPP, "Policy and charging control over Rx reference
point", June 2008,
<http://www.3gpp.org/ftp/Specs/html-info/29214.htm>.

Authors' Addresses

Brian Stucker
Unaffiliated

Email: obsidian97@gmail.com
URI: http://www.linkedin.com/in/bstucker

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: gsalguei@cisco.com