AVT
Internet-Draft
Intended status: Standards Track
Expires: October 14, 2010

A. Begen
B. VerSteeg
Cisco

April 12, 2010

Port Mapping Between Unicast and Multicast RTP Sessions
draft-ietf-avt-ports-for-ucast-mcast-rtp-01

Abstract

This document presents a port mapping solution that allows RTP receivers to choose their own receive ports for an auxiliary unicast session in RTP applications using both unicast and multicast services. The solution requires multiplexing RTP and RTCP on a single port on both endpoints in the unicast session.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 14, 2010.

Table of Contents

1. Introduction

In (any-source or source-specific) multicast RTP applications, destination ports, i.e., the ports on which the multicast receivers receive the RTP and RTCP packets, are defined declaratively. In other words, the receivers cannot choose their receive ports and the sender(s) use the pre-defined ports.

In unicast RTP applications, the receiving end needs to choose its receive ports for RTP and RTCP since these ports are local resources and only the receiving end can determine which ports are available to use. The receiving may convey its request to the sending end through different ways, one of which is the Offer/Answer Model [RFC3264] for the Session Description Protocol (SDP) [RFC4566]. However, the Offer/Answer Model requires offer/answer exchange(s) between the endpoints, and the resulting delay may not be desirable in delay-sensitive real-time applications. Furthermore, the Offer/Answer Model may be burdensome for the endpoints that are concurrently running a large number of unicast sessions with other endpoints.

In this specification, we consider an RTP application that uses one or more unicast and multicast RTP sessions together. While the declaration and selection of the ports are well defined and work well for multicast and unicast RTP applications, respectively, the usage of the ports introduces complications when a receiving end mixes unicast and multicast RTP sessions within the same RTP application.

An example scenario is where the RTP packets are distributed through source-specific multicast (SSM) and a receiver sends unicast RTCP feedback to a local repair server (also functioning as a feedback target) [RFC5760] asking for a retransmission of the packets it is missing, and the local repair server sends the retransmission packets over a unicast RTP session [RFC4588].

Another scenario is where a receiver wants to rapidly acquire a new primary multicast RTP session and receives one or more RTP burst packets over a unicast session before joining the SSM session [I-D.ietf-avt-rapid-acquisition-for-rtp]. Similar scenarios exist in applications where some part of the content is distributed through multicast while the receivers get additional and/or auxiliary content through one or more unicast connections, as sketched in Figure 1.

In this document, we discuss this problem and introduce a solution that we refer to as Port Mapping. This solution allows receivers to choose their desired RTP and RTCP receive ports for every unicast session when they are running RTP applications using both unicast and multicast services.

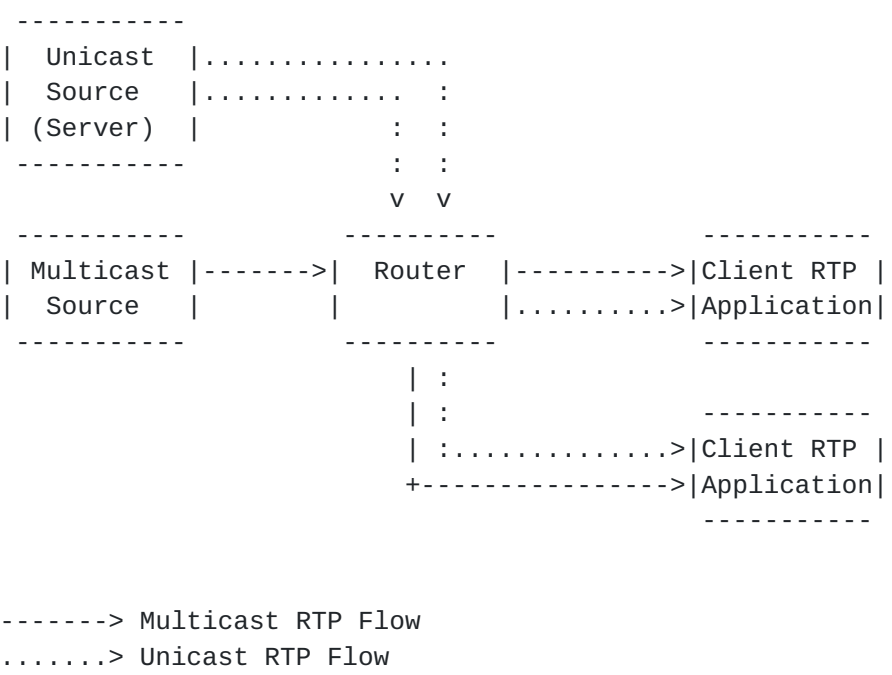


Figure 1: RTP applications simultaneously using both unicast and multicast services

In the remainder of this document, we refer to the RTP endpoints that serve other RTP endpoints over a unicast session as the Servers. The receiving RTP endpoints are referred to as Clients.

2. Requirements Notation

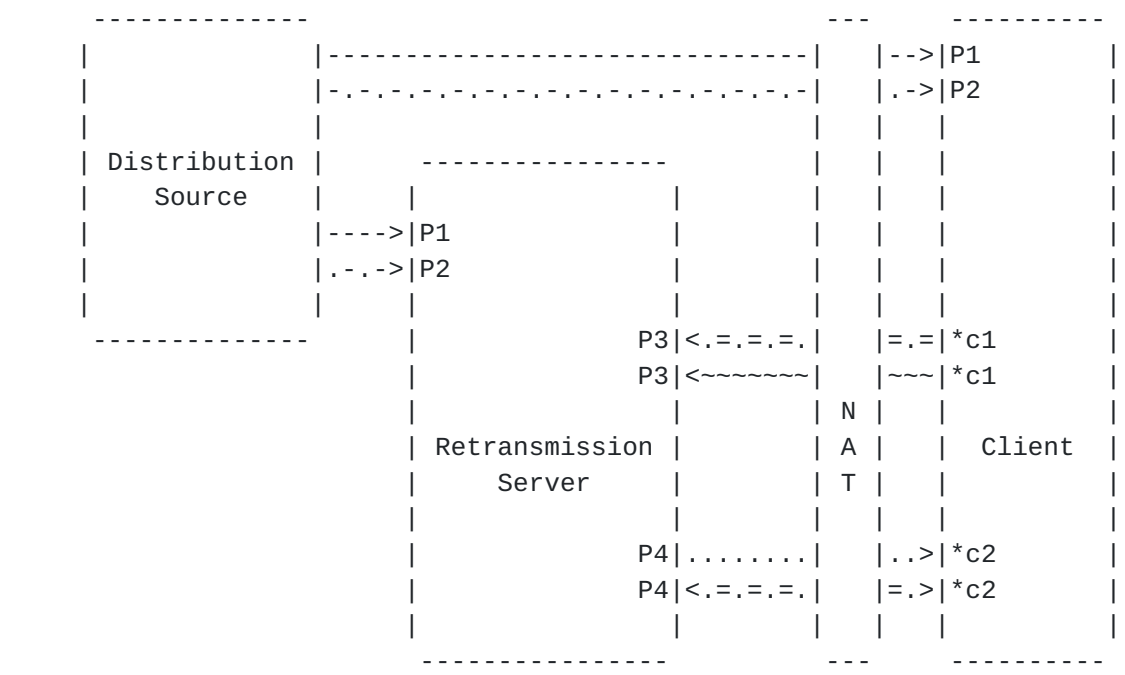
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Port Mapping

We present the details of the port mapping solution in the context of an illustrative example.

Consider an SSM distribution network where a distribution source multicasts RTP packets to a large number of clients, and one or more retransmission servers function as feedback targets to collect unicast RTCP feedback from these clients [RFC5760]. The retransmission servers also join the primary multicast session to receive the multicast packets and cache them for a certain time period. When a client detects missing packets in the primary multicast session, it requests a retransmission from one of the retransmission servers by using an RTCP NACK message [RFC4585]. The retransmission server pulls the requested packet(s) out of the cache and retransmits them to the requesting client.

The pertaining RTP and RTCP flows are sketched in Figure 2. Between the client and server, there may be one or more NAT devices [RFC4787].



-------> Multicast RTP Flow
.-.-.-.> Multicast RTCP Flow
.=.=.=.> Unicast RTCP Reports
~~~~~~~> Unicast RTCP Feedback Messages
.......> Unicast RTP Flow

Figure 2: Example scenario showing an SSM distribution with support for retransmissions from a server

In this figure, we have the following multicast and unicast ports:

- Ports P1 and P2 denote the destination RTP and RTCP ports in the primary multicast session, respectively. The clients listen to these ports to receive the multicast RTP and RTCP packets. Ports P1 and P2 are defined declaratively.

- Port P3 denotes the RTCP port on the feedback target running on the retransmission server to collect the RTCP feedback messages, and RTP receiver and extended reports from the clients in the primary multicast session. Port P3 is defined declaratively.

- Port P4 denotes the port on the retransmission server used for the unicast session. The server multiplexes RTP and RTCP traffic on this single port [I-D.ietf-avt-rtp-and-rtcp-mux] in the unicast session. Port P4 is defined declaratively.
~~~~~~~

- Ports *c1 and *c2 are chosen by the client. *c1 denotes the port on the client used to send the unicast RTCP feedback in the primary multicast session. *c2 denotes the port on the client used in the unicast session. The client muxes RTP and RTCP traffic on this single port [I-D.ietf-avt-rtp-and-rtcp-mux] in the unicast session. Ports c1 and c2 do not have to be different ports.

Once the unicast session is established, the server needs to remember the public IP address and public port of the client as part of the session state information. The public ports of the client are denoted by c1' and c2'.

In addition to the ports, we use the following notation:

- DS: IP address of the distribution source
- G: Destination multicast address
- S: IP address of the retransmission server
- C: IP address of the client
- C': Public IP address of the client (as seen by the server)

We assume that the information declaratively defined is available as part of the session description information and is provided to the clients. The Session Description Protocol (SDP) [RFC4566] and other session description methods can be used for this purpose.

3.1. Steps for Establishing a Unicast Session

The steps to establish a unicast session are provided below:

1. The client ascertains server address (S) and port numbers (P3 and P4) from the session description.

2. The client determines its receive port numbers (*c1 and *c2).

3. The client sends a message to the server via a new RTCP message, called PortMappingRequest. This message MUST be sent from port c2 to port P4. The server learns client's public IP address (C') and its public RTP/RTCP port (c2') from the received message.

4. The server generates an opaque encapsulation (called Cookie) that conveys client's addressing information using a reversible transform only known to the server.

5. The server sends the Cookie back to the client using a new RTCP message, called PortMappingResponse. This message MUST be sent from port P4 to port c2'.

6. The client includes the Cookie when necessary in the subsequent messages sent to the server.

7. Normal flows ensue, with the server using the addressing encapsulated in the opaque Cookie. The client is responsible for keeping the NAT binding alive for the duration of the unicast session.

3.2. Implications of NATs

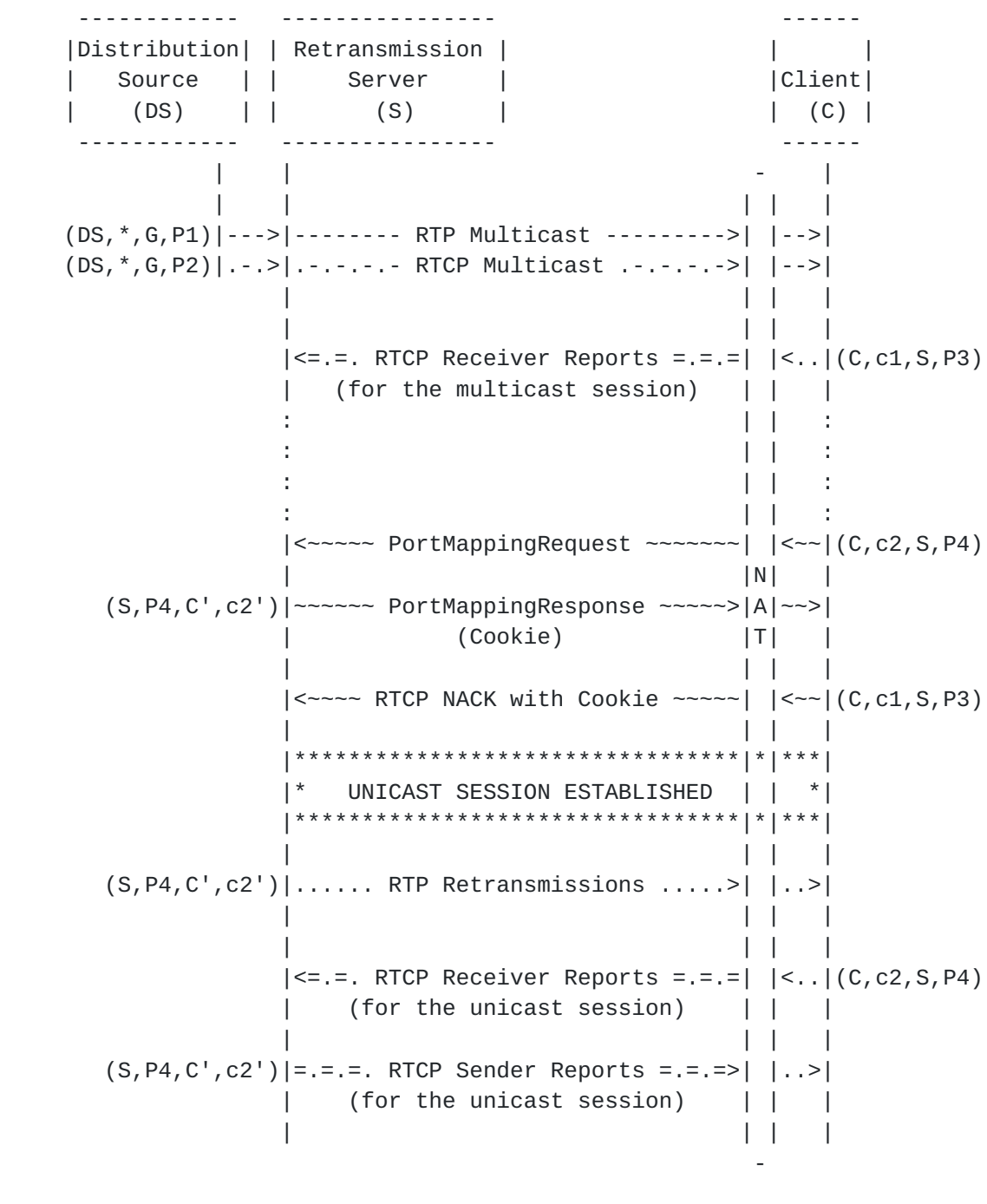
There may be one or more NAT devices between the client and server. Without an external mechanism such as STUN [RFC5389], the client cannot determine whether there are any NATs between itself and the server. Such NAT devices would block all incoming traffic unless the client sent traffic of the same transport protocol to the server first. Thus, the client has always to assume that there is at least one NAT device and send periodic packets to keep the NAT binding alive [I-D.ietf-avt-app-rtp-keepalive]. Since the client multiplexes RTP and RTCP on a single port, it has to keep a single NAT binding alive for each unicast session. See Section 3.4 for further details.

If the NAT device fails for some reason and then restarts, the public IP address and/or port assigned to a particular client may change. This will invalidate the previously acquired cookies and may result in a failure in the unicast session. Upon detecting the failure, the client must acquire new cookies. Applications using this method must be aware of the potential temporary interruptions.

The NAT device may have endpoint-independent mappings [RFC4787], meaning that it assigns the same public IP address and port for the packets sent from the same internal IP address and port, even when the client is talking to different destinations. Oppositely, the NAT device may have endpoint-dependent mappings in which case the public IP address and port of the outgoing packets may differ when they are sent to different destinations. In practice, however, it is a difficult task to determine the type of a NAT device [I-D.ietf-behave-nat-behavior-discovery].

3.3. Message Flows

Figure 3 shows the message flows, where each message is appended with the (Source Address, Source Port, Destination Address, Destination Port) information.



-------> Multicast RTP Flow
.-.-.-.> Multicast RTCP Flow
.=.=.=.> Unicast RTCP Reports
~~~~~~~> Unicast RTCP Feedback Messages
.......> Unicast RTP Flow

Figure 3: Message flows for establishing a unicast session
~~~~~~~

In the example above, the compound RTCP packet carrying the NACK message also carries the Cookie since the server must know which port the client is expecting to receive the RTP retransmission packet(s) and RTCP sender reports on. If an RTCP message from the client will not trigger any transmission from the server (e.g., RTCP receiver and extended reports), it does not have to include the Cookie.

3.4. Keeping the NAT Binding(s) Alive

Editor's note: We need to determine the best option to keep the NAT bindings alive [I-D.ietf-avt-app-rtp-keepalive].

Editor's note: Are RTCP receiver/extended reports enough to keep the binding alive?

TBD.

3.5. SDP Description

The SDP describing the scenario given in Figure 2 can be written as:

```
v=0
o=ali 1122334455 1122334466 IN IP4 nack.example.com
s=Local Retransmissions
t=0 0
a=group:FID 1 2
a=rtcp-unicast:rsi
m=video 41000 RTP/AVPF 98
i=Primary Multicast Stream
c=IN IP4 233.252.0.2/255
a=source-filter:incl IN IP4 233.252.0.2 198.51.100.1
a=rtpmap:98 MP2T/90000
a=multicast-rtcp:42000
a=rtcp:43000 IN IP4 192.0.2.1
a=rtcp-fb:98 nack
a=mid:1
m=video 51000 RTP/AVPF 99
i=Unicast Retransmission Stream
c=IN IP4 192.0.2.1
a=sendonly
a=rtpmap:99 rtx/90000
a=rtcp-mux
a=fmtp:99 apt=98; rtx-time=5000
a=mid:2
```

Figure 4: SDP describing an SSM distribution with support for retransmissions from a local server

In this SDP, the source stream is multicast from a distribution source (with a source IP address of 198.51.100.1) to the multicast destination address of 233.252.0.2 (G) and port 41000 (P1). The associated RTCP packets are multicast in the same group to port 42000 (P2). A retransmission server including feedback target functionality with an IP address of 192.0.2.1 (S) and port of 43000 (P3) is specified with the 'rtcp' attribute. The server uses port 51000 (P4) for the unicast sessions.

4. Message Formats

The common packet format for the RTCP feedback messages is defined in Section 6.1 of [RFC4585]. A feedback message has a fixed-length field for version, padding, feedback message type (FMT), payload type (PT), length, SSRC of packet sender, SSRC of media sender as well as a variable-length field for feedback control information (FCI).

In the PortMappingRequest and PortMappingResponse messages, the PT field is set to RTPFB (205), and the respective FMT fields are set to PMReq (7) and PMRes (8). Depending on the specific scenario, it may be desirable to send these messages in a reduced-size RTCP packet [RFC5506]. However, unless support for [RFC5506] has been signaled, compound RTCP packets MUST be used by following [RFC3550] rules.

Editor's note: Should the server always respond to the PMReq message as soon as possible?

Following the rules specified in [RFC3550], all integer fields in the messages defined below are carried in network-byte order, that is, most significant byte (octet) first, also known as big-endian. Unless otherwise noted, numeric constants are in decimal (base 10). Any Reserved field SHALL be set to zero and ignored.

4.1. PortMappingRequest (PMReq)

Editor's note: How do we set the media source SSRC field in the following message? Is it application specific (e.g., retransmissions, RAMS, etc.)?

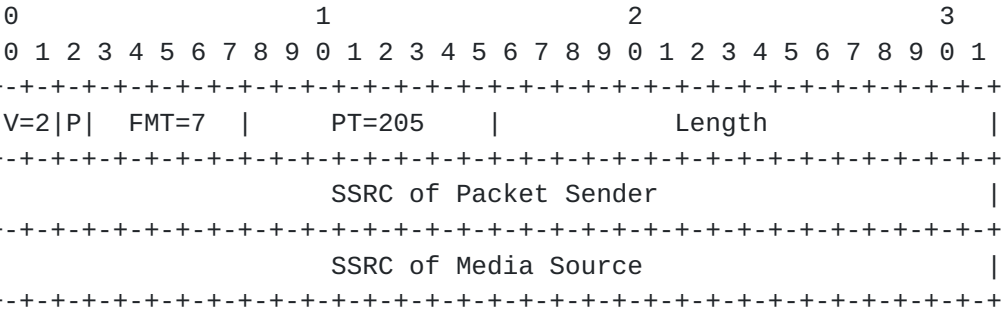


Figure 5: Syntax for the PortMappingRequest (PMReq) message

Editor's note: What else do we need to transmit in the PMReq message?

4.2. PortMappingResponse (PMRes)

Editor's note: How do we set the packet sender SSRC and media source SSRC fields in the following message? Are they application specific (e.g., retransmissions, RAMS, etc.)?

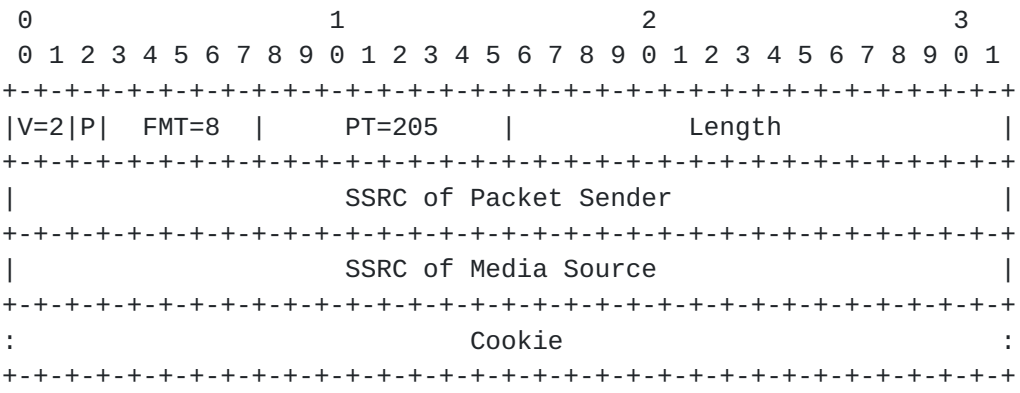


Figure 6: Syntax for the PortMappingResponse (PMRes) message

Editor's note: What else do we need to transmit in the PMRes message?

5. Procedures for Cookie Construction

Editor's notes:

The Cookie may contain

- A 32-bit value randomly generated by the client [RFC4086]
- Client's IP address and port (Note that the PMReq and NACK messages are sent from different client ports (and maybe from different public IP addresses as well), thus the server cannot use this information to check whether a cookie is used by the true owner of that cookie)
- Client's CNAME
- A timestamp to protect against replay attacks (Should the server tell the client about the expiration date so that the client may request a new cookie before the current one expires?)
- HMAC [RFC2104] of the above information (where only the server knows the HMAC secret)

Details are TBC.

6. Security Considerations

Editor's notes:

- Cookie expiration via timestamping. This could be important for clients behind the same NAT (The clients may still generate the same random number)
- Stealing cookies. Can CNAME be used to avoid this for the clients behind the same NAT?
- Modifying cookies. Can somebody manipulate the cookies to redirect the traffic?

7. IANA Considerations

The following contact information shall be used for all registrations in this document:

Ali Begen
abegen@cisco.com

170 West Tasman Drive
San Jose, CA 95134 USA

Note to the RFC Editor: In the following, please replace "XXXX" with the number of this document prior to publication as an RFC.

7.1. Registration of FMT Values

Within the RTPFB range, the following format (FMT) values are registered:

Name:	PMReq
Long name:	Port Mapping Request
Value:	7
Reference:	[RFCXXXX]

Name:	PMRes
Long name:	Port Mapping Response
Value:	8
Reference:	[RFCXXXX]

8. Contributors and Acknowledgments

Many individuals in the AVT and MMUSIC WGs have contributed to this work, reviewed earlier versions of this specification and provided feedback. The authors thank each of them.

9. References

9.1. Normative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

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

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

[I-D.ietf-avt-rtp-and-rtcp-mux] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", draft-ietf-avt-rtp-and-rtcp-mux-07 (work in progress), August 2007.

[I-D.ietf-avt-app-rtp-keepalive] Marjou, X. and A. Sollaud, "Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows.", draft-ietf-avt-app-rtp-keepalive-07 (work in progress), December 2009.

9.2. Informative References

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

[I-D.ietf-avt-rapid-acquisition-for-rtp] Steeg, B., Begen, A., Caenegem, T., and Z. Vax, "Unicast-Based Rapid Acquisition of Multicast RTP Sessions", draft-ietf-avt-rapid-acquisition-for-rtp-08 (work in progress), March 2010.

[I-D.ietf-behave-nat-behavior-discovery]
MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using STUN", draft-ietf-behave-nat-behavior-discovery-08 (work in progress), September 2009.

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

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

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

[RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

Authors' Addresses

Ali Begen
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: abegen@cisco.com

Bill VerSteeg
Cisco
5030 Sugarloaf Parkway
Lawrenceville, GA 30044
USA

Email: billvs@cisco.com