AVT
Internet-Draft
Intended status: Standards Track
Expires: May 12, 2011

A. Begen
D. Wing
Cisco
T. VanCaenegem
Alcatel-Lucent

November 8, 2010

Token-Based Port Mapping Between Unicast and Multicast RTP Sessions
draft-begen-avt-token-for-portmapping-03

Abstract

This document presents a port mapping solution that allows RTP receivers to choose their own ports for an auxiliary unicast session in RTP applications using both unicast and multicast services (almost) without the need for retrieving pre-authorization.

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 May 12, 2011.

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 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 UDP ports for RTP and RTCP in every unicast session when they are running RTP applications using both unicast and multicast services, and offer/answer exchange is not available. This solution is not applicable in cases where TCP is used as the

transport protocol in the unicast sessions. For such scenarios, refer to [RFC4145].

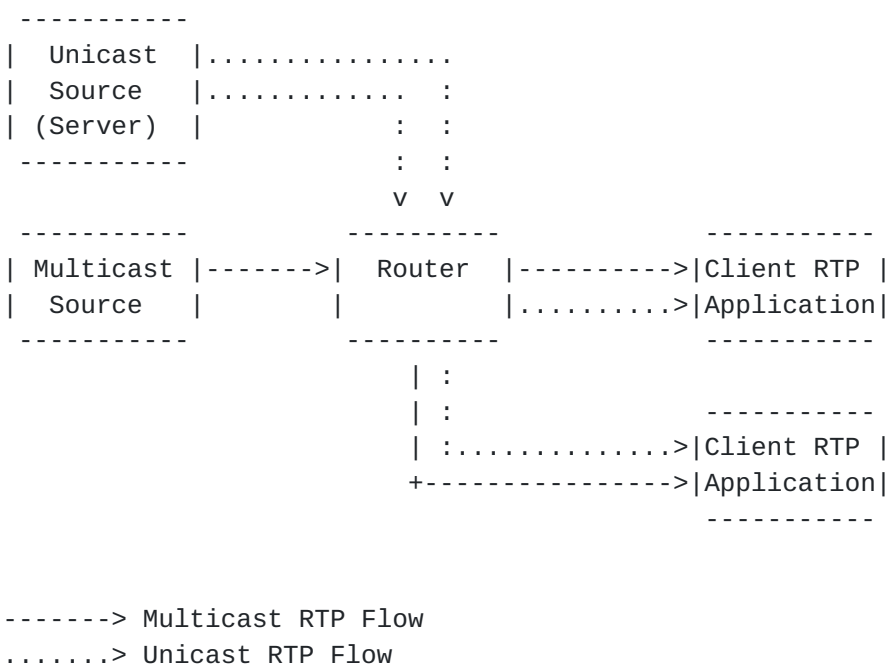

Figure 1: RTP applications simultaneously using both unicast and multicast services

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. Token-Based Port Mapping

Token-based Port Mapping consists of two steps: (i) Token request and retrieval, and (ii) unicast session establishment. These are described below.

3.1. Token Request and Retrieval

This first step is required to be completed only once. Once a Token is retrieved from a particular server, it can be used for all the unicast sessions the client will be running with this particular server. By default, Tokens are server specific. However, the client can use the same Token to communicate with different servers if these servers are provided with the same key used to generate the Token. The Token becomes invalid if client's public IP address changes or when the server expires the Token. In these cases, the client has to request a new Token.

The Token is essentially an opaque encapsulation that conveys client's IP address information (as seen by the server) using a reversible transform only known to the server. When a request is received, the server creates a Token for this particular client, and sends it back to the client. Later, when the client wants to establish a unicast session, the Token will be validated by the server, making sure that the IP address information matches. This is effective against DoS attacks, e.g., an attacker cannot simply spoof another client's IP address and start a unicast transmission towards random clients.

3.2. Unicast Session Establishment

We illustrate the second step with an 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 multicast session to receive the multicast packets and cache them for a certain time period. When a client detects missing packets in the 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 [RFC4588].

The pertaining RTP and RTCP flows are sketched in Figure 2. Between the client and server, there can be one or more Network Address Port Translators (NAPT - hereafter simply called NAT) devices [RFC4787].

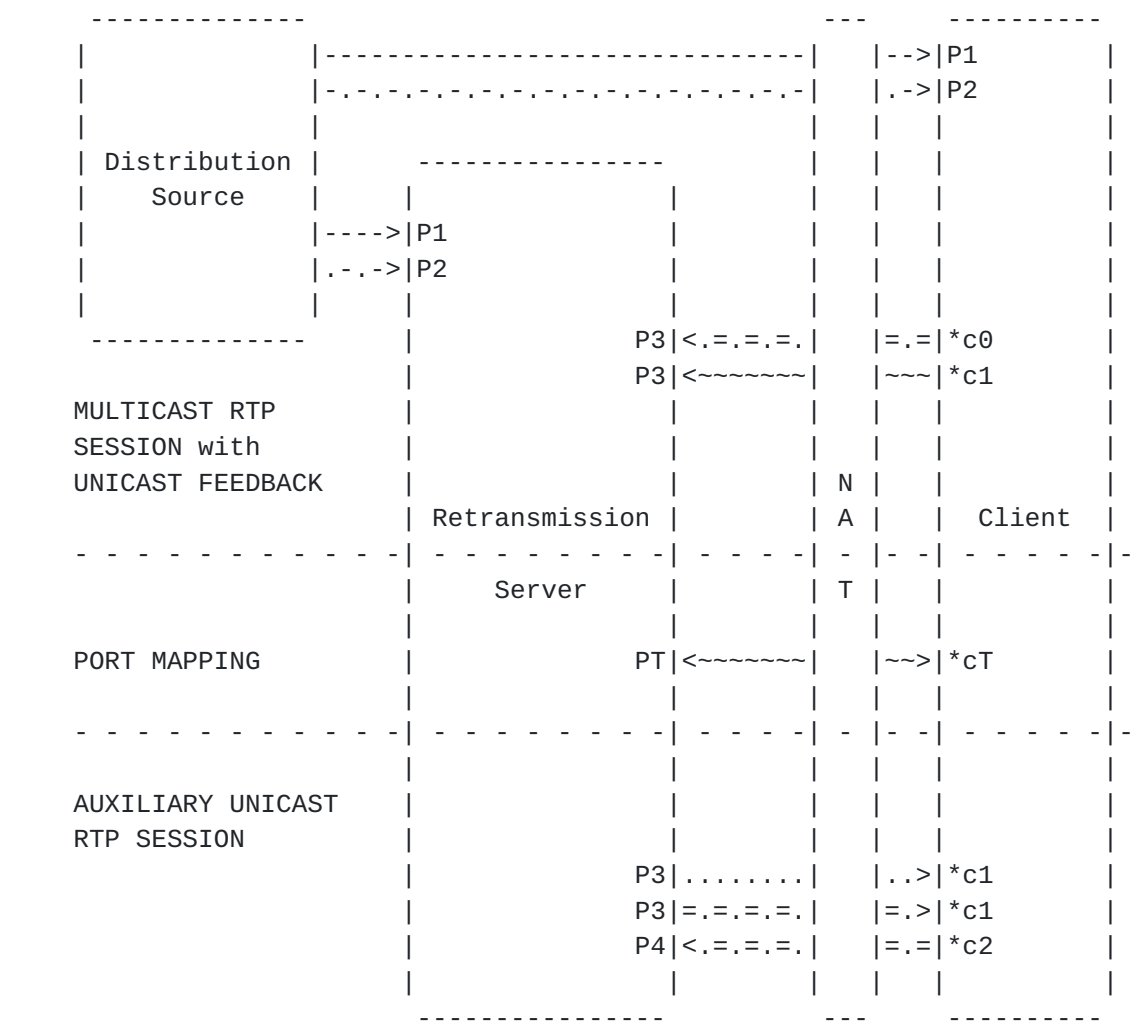

Figure 2: Example scenario showing an SSM distribution with support for retransmissions from a server

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

o Ports P1 and P2 denote the destination RTP and RTCP ports in the 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 RTCP receiver and extended reports from the clients in the multicast session. This is also the port that the retransmission server uses to send the RTP packets and RTCP sender reports in the unicast session. Port P3 is defined declaratively.

- Port P4 denotes the RTCP port on the retransmission server used to collect the RTCP receiver and extended reports for the unicast session. Port P4 is defined declaratively and MUST be different from port P3.

- Ports *c0, *c1 and *c2 are chosen by the client. *c0 denotes the port on the client used to send the RTCP reports for the multicast session. *c1 denotes the port on the client used to send the unicast RTCP feedback messages in the multicast session and to receive the RTP packets and RTCP sender reports in the unicast session. *c2 denotes the port on the client used to send the RTCP receiver and extended reports in the unicast session. Ports c0, c1 and c2 MAY be the same port or different ports. However, there are two advantages of using the same port for both c0 and c1:

  1. Some NATs only keep bindings active when a packet goes from the inside to the outside of the NAT (See REQ-6 of Section 4.3 of [RFC4787]). When the retransmission server sends unicast packets for a long period of time, this can exceed that timeout. If c0=c1, the occasional (periodic) RTCP receiver reports sent from port c0 (for the multicast session) will ensure the NAT does not time out the public port associated with the incoming unicast traffic to port c1.

  2. Having c0=c1 conserves NAT port bindings.

  Thus, it is strongly RECOMMENDED that the client uses the same port for c0 and c1.

- Ports PT and cT denote the ports through which the Token request and retrieval occur at the server and client sides, respectively. Port PT is declared on a per unicast session basis, although its value MAY be the same for all unicast sessions sourced by the server. This way, a Token once requested and retrieved by a client from port PT remains valid across different unicast sessions. Port PT MAY be equal to port P3. Port cT MAY also be equal to ports c0 and c1.

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.

The following steps summarize the Token-based solution:

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

2. The client determines its port numbers (*c0, *c1 and *c2).

3. If the client does not have a valid Token:

   A. The client first sends a message to the server via a new RTCP message, called Port Mapping Request to port PT. This message is sent from port cT on the client side. The server learns client's public IP address (C') from the received message. The client can send this message anytime it wants (e.g., during initialization), and does not normally ever need to re-send this message (See Section 7).

   B. The server generates an opaque encapsulation (i.e., the Token) that conveys client's IP address information using a reversible transform only known to the server. For details, see Section 6.

   C. The server sends the Token back to the client using a new RTCP message, called Port Mapping Response. This message MUST be sent from port PT to port cT.

4. The client provides the Token to the server using a new RTCP message, called Token Verification, whenever the client sends an RTCP feedback message for triggering or controlling a unicast session. Note that the unicast session is only established after the server has received a feedback message (along with a valid Token) from the client for which it needs to react by sending unicast data. Until a unicast session is established, neither

the server nor the client needs to send RTCP reports for the unicast session.

5. Normal flows ensue as shown in Figure 2. Note that in the unicast session the RTP and RTCP packets MUST be multiplexed on the (same) port c1. If the client uses the same port for both c0 and c1, the RTCP reports sent for the multicast session keep the P3->c1(=c0) binding alive. If the client uses different ports for c0 and c1, the client needs to periodically send an explicit keep-alive message [I-D.ietf-avt-app-rtp-keepalive] to keep the P3->c1 binding alive during the lifetime of the unicast session if the unicast session's lifetime is likely to exceed the NAT's timeout value.

4. The portmapping-req Attribute

This new SDP attribute is used declaratively to indicate the port for obtaining a Token. Its presence indicates that a Token MUST be included in the feedback messages sent to the server triggering or controlling a unicast session.

The formal description of the 'portmapping-req' attribute is defined by the following ABNF [RFC5234] syntax:

```
portmapping-req-attribute = "a=portmapping-req:" port CRLF
```

Here, 'port' is defined as specified in Section 9 of [RFC4566]. The 'portmapping-req' attribute is used as a session-level or media-level attribute.

5. Message Formats

This section defines the formats of the RTCP transport-layer feedback messages that are exchanged between a server and a client for the purpose of Token-based port mapping. Three RTCP messages are defined:

1. Port Mapping Request

2. Port Mapping Response

3. Token Verification

These are all payload-independent RTCP feedback messages with a common format defined in Section 6.1 of [RFC4585], also sketched in Figure 3.

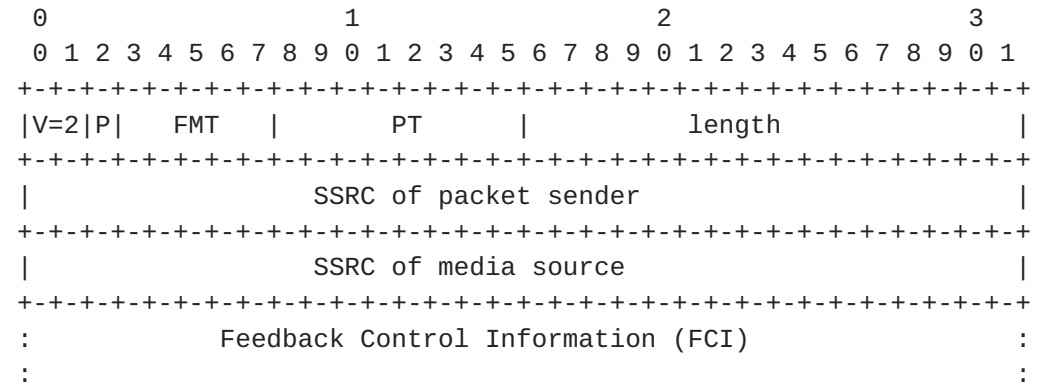

Figure 3: The common packet format for the RTCP feedback messages

Each feedback message has a fixed-length field for version, padding, feedback message type (FMT), packet type (PT), length, SSRC of packet sender, SSRC of media source as well as a variable-length field for feedback control information (FCI).

In the new messages defined in this section, the PT field is set to RTPFB (205) and the FMT field is set to Port Mapping (7). Individual Port Mapping messages are identified by a sub-field called Sub Feedback Message Type (SFMT). Any Reserved field SHALL be set to zero and ignored.

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 stated, numeric constants are in decimal (base 10).

5.1. Port Mapping Request

The Port Mapping Request message is identified by SFMT=1. This message is a unicast feedback message transmitted by the client to a dedicated server port to request a Token. In the Port Mapping Request message, the client MUST set both the packet sender SSRC and media source SSRC fields to its own SSRC since the Port Mapping Request message is not necessarily linked to any specific media source. The FCI field has the structure depicted in Figure 4.

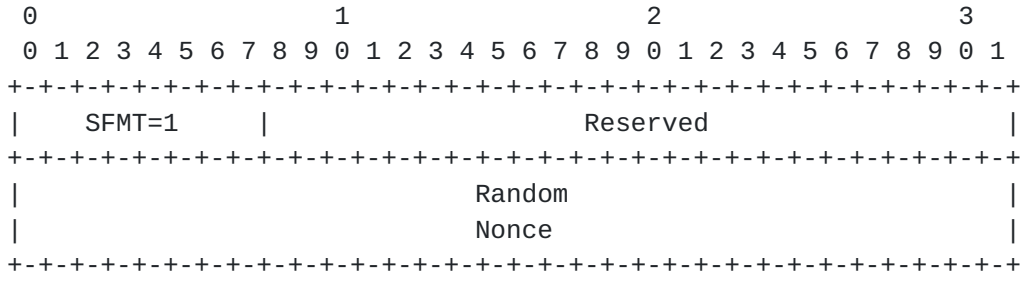

Figure 4: The FCI field of Port Mapping Request message

o Random Nonce (64 bits): Mandatory field that contains a random nonce value generated by the client following the procedures of [RFC4086]. This nonce MUST be taken into account by the server when generating a Token for the client to enable better security for clients that share the same IP address. If the Port Mapping Request message is transmitted multiple times for redundancy reasons, the random nonce value MUST remain the same in these duplicated messages.

5.2. Port Mapping Response

The Port Mapping Response message is identified by SFMT=2. This message is sent by the server and delivers the Token to the client. In the Port Mapping Response message, the packet sender SSRC and media sender SSRC fields are both set to the client's SSRC since the Port Mapping Response message is not necessarily linked to any specific media source. The FCI field has the structure depicted in Figure 5.

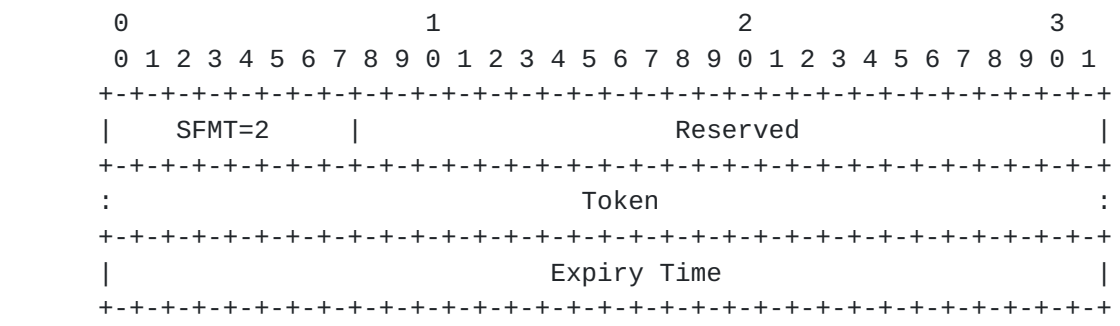

Figure 5: FCI field syntax for the Port Mapping Response message

- Token (128 bits): Mandatory element that contains the Token generated by the server.

- Expiry Time (32 bits): Mandatory element that contains the expiry time of the Token. The expiry time is expressed in seconds from the time the Token was generated. An expiry time of zero indicates that the accompanying Token is not valid.

5.3. Token Verification

The Token Verification message is identified by SFMT=3. This message contains the Token and MUST accompany any other RTCP feedback message sent by the client to trigger or control a unicast session. Examples include the RAMS-R and RAMS-T messages [I-D.ietf-avt-rapid-acquisition-for-rtp] as well as the NACK messages [RFC4585]. In the Token Verification message, the client MUST set both the packet sender SSRC and media source SSRC fields to its own SSRC since the media source SSRC may not be known. The client MUST NOT send a Token Verification message with a Token that has expired. The FCI field has the structure depicted in Figure 6.

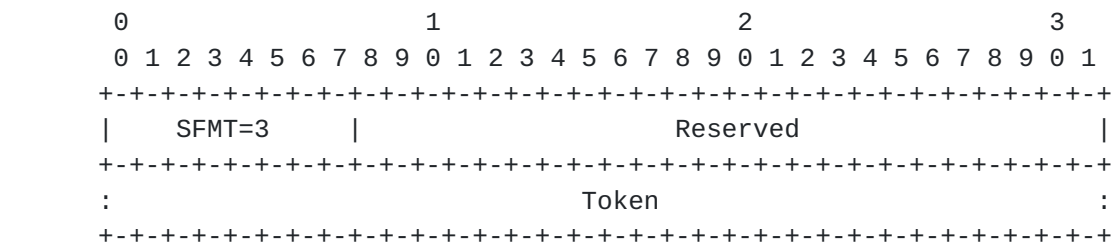

Figure 6: FCI field syntax for the Token Verification message

- Token (128 bits): Mandatory element that contains the Token previously acquired by the client.

6. Procedures for Token Construction

Editor's notes:

The Token SHOULD be calculated by the server by taking into account:

- Client's IP address as seen by the server
- The nonce generated by the client and inserted in the Port Mapping Request message
- A timestamp to protect against replay attacks
- HMAC [RFC2104] of the above information (where only the server knows the HMAC secret)

The server conveys the expiration time in the clear to the client in the Port Mapping Response message. Thus, the client can request a new Token before the current one expires.

Details are TBC.

7. Validating Tokens

Upon receipt of an RTCP feedback message along with the Token Verification message that contains a Token, the server MUST validate the Token. The server considers a Token valid if the source IP address of the RTCP feedback message matches the IP address in the Token and the Token has not expired yet.

The IP address is encoded into the Token by the server, using an algorithm known only to the server. This, combined with the expiration time provides protection against DoS attacks so that a client using a certain IP address cannot cause one or more RTP packets to be sent to another client with a different IP address.

When the server detects that the Token is invalid, it SHOULD NOT silently discard client's message since this adds an undesired delay. Instead, it is RECOMMENDED that applications define an application-specific error response. In applications that have not defined an error response, the server MUST reply back to the client with a Port Mapping Response message (that goes from port P3 on the server to port c1 on the client) where the Token field carries the invalid Token sent by the client and the Expiry Time field is set to zero (indicating that the Token is invalid).

For applications using [I-D.ietf-avt-rapid-acquisition-for-rtp], this draft defines a new 4xx-level response code in the RAMS Response Code Space Registry.

8. SDP Example

The declarative SDP describing the scenario given in Figure 2 is 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=Multicast Stream
c=IN IP4 233.252.0.2/255
a=source-filter:incl IN IP4 233.252.0.2 198.51.100.1   ; Note 1
a=rtpmap:98 MP2T/90000s
a=multicast-rtcp:41500                                 ; Note 1
a=rtcp:42000 IN IP4 192.0.2.1                          ; Note 2
a=rtcp-fb:98 nack                                      ; Note 2
a=mid:1
m=video 42000 RTP/AVPF 99                              ; Note 3
i=Unicast Retransmission Stream
c=IN IP4 192.0.2.1
a=sendonly
a=rtpmap:99 rtx/90000
a=rtcp-mux                                             ; Note 4
a=rtcp:42500                                           ; Note 5
a=fmtp:99 apt=98; rtx-time=5000
a=portmapping-req:30000                                ; Note 6
a=mid:2
```

Figure 7: SDP describing an SSM distribution with support for retransmissions from a local server

In this description, we highlight the following notes:

Note 1: The source stream is multicast from a distribution source with a source IP address of 198.51.100.1 (DS) 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 41500 (P2).

Note 2: A retransmission server including feedback target functionality with an IP address of 192.0.2.1 (S) and port of 42000 (P3) is specified with the 'rtcp' attribute. The feedback functionality is enabled for the RTP stream with payload type 98 through the 'rtcp-fb' attribute [RFC4585].

Note 3: The port specified in the second "m" line (for the unicast stream) does not mean anything in this scenario as the client does not send any RTP traffic back to the server.

Note 4: The server multiplexes RTP and RTCP packets on the same port (c1 in Figure 2).

Note 5: The server uses port 42500 (P4) for the unicast sessions.

Note 6: The "a=portmapping-req" line indicates that a Token needs to be retrieved first before a unicast session associated to the multicast session can be established and that the Port Mapping Request message needs to be sent to port 30000 (PT).

9. Address Pooling NATs

Large-scale NAT (LSN) devices have a pool of public IPv4 addresses and map internal hosts to one of those public IPv4 addresses. As long as an internal host maintains an active mapping in the NAT, the same IPv4 address is assigned to new connections. However, once all of the host's mappings have been deleted (e.g., because of timeout), it is possible that a new connection from that same host will be assigned a different IPv4 address from the pool. When that occurs, the Token will be considered invalid by the server, causing an additional round trip for the client to acquire a fresh Token.

Any traffic from the host which traverses the NAT will prevent this problem. As the host is sending RTCP receiver reports at least every 5 seconds (Section 6.2 of [RFC3550]) for the multicast session it is receiving, those RTCP messages will be sufficient to prevent this problem.

10. Security Considerations

The Token, which is generated based on a client's IP address and expiration date, provides protection against denial-of-service (DoS) attacks. An attacker using a certain IP address cannot cause one or more RTP packets to be sent to a victim client who has a different IP address. However, if the attacker acquires a valid Token for a victim and can spoof the victim's source address, this approach becomes vulnerable to replay attacks.

In networks where multicast services are enabled, however, source address validation is already a necessary requirement to prevent an attacker from spoofing a multicast (IGMP/MLD) Join message and attracting multicast traffic to a victim. As such an attack is even easier to execute, it is expected that source address validation will be available. Yet, if the source address validation capability is known not to perform properly, the server can reduce the probability of a Token being stolen and used to initiate an attack by expiring the Token in a short amount of time. In the extreme case, the server can expire the token after its first use.

11. IANA Considerations

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

Ali Begen
abegen@cisco.com

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

11.1. Registration of SDP Attributes

This document registers a new attribute name in SDP.

```
SDP Attribute ("att-field"):
Attribute name:     portmapping-req
Long form:          Port for requesting Token
Type of name:       att-field
Type of attribute:  Either session or media level
Subject to charset: No
Purpose:            See this document
Reference:          [RFCXXXX]
Values:             See this document
```

11.2. Registration of FMT Values

Within the RTPFB range, the following format (FMT) value is registered:

```
Name:       Port Mapping
Long name:  Port Mapping Between Unicast and Multicast RTP Sessions
Value:      7
Reference:  [RFCXXXX]
```

11.3. SFMT Values for Port Mapping Messages Registry

This document creates a new sub-registry for the sub-feedback message type (SFMT) values to be used with the FMT value registered for Port Mapping messages. The registry is called the SFMT Values for Port Mapping Messages Registry. This registry is to be managed by the IANA according to the Specification Required policy of [RFC5226].

The length of the SFMT field in the Port Mapping messages is a single octet, allowing 256 values. The registry is initialized with the

following entries:

Value	Name	Reference
0	Reserved	[RFCXXXX]
1	Port Mapping Request	[RFCXXXX]
2	Port Mapping Response	[RFCXXXX]
3	Token Verification	[RFCXXXX]
4-254	Assignable - Specification Required	
255	Reserved	[RFCXXXX]

The SFMT values 0 and 255 are reserved for future use.

Any registration for an unassigned SFMT value needs to contain the following information:

- Contact information of the one doing the registration, including at least name, address, and email.

- A detailed description of what the new SFMT represents and how it shall be interpreted.

11.4. RAMS Response Code Space Registry

This document adds the following entry to the RAMS Response Code Space Registry.

Code	Description	Reference
405	Invalid Token	[RFCXXXX]

This response code is used when the Token included by the RTP_Rx in the RAMS-R message is invalid.

12. Acknowledgments

The approach presented in this document came out after discussions with various individuals in the AVT and MMUSIC WGs, and the breakout session held in the Anaheim meeting. We thank each of these individuals.

13. References

13.1. 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.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

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

13.2. Informative References

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

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[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-16 (work in progress), October 2010.

[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.

[I-D.ietf-avt-app-rtp-keepalive] Marjou, X. and A. Sollaud, "Application Mechanism for keeping alive the Network Address Translator (NAT) mappings associated to RTP flows.", draft-ietf-avt-app-rtp-keepalive-09 (work in progress), September 2010.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
USA

Email: dwing@cisco.com

Tom VanCaenegem
Alcatel-Lucent
Copernicuslaan 50
Antwerpen, 2018
Belgium

Email: Tom.Van_Caenegem@alcatel-lucent.com