

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 1, 2012

M. Ramalho, Ed.
D. Wing
M. Perumal
Cisco Systems
N. Harada
NTT
H. Kaplan
Acme Packet
February 29, 2012

G.711.0 Compression Segments
draft-ramalho-g7110-segments-00

Abstract

This document describes use cases for ITU-T Recommendation G.711.0 compression of ITU-T Recommendation G.711 payloads when deployed in transport system segments using the Real-Time Transport Protocol (RTP).

ITU-T Rec. G.711.0 defines a lossless and stateless compression for G.711 packet payloads typically used in IP networks. Although the use of ITU-T Rec. G.711.0 can be negotiated end-to-end, being lossless and stateless it can also be applied as a compression mechanism "in-the-middle" of an end-to-end ITU-T G.711 negotiated session. These "in-the-middle" applications of ITU-T Rec. G.711.0 are called "G.711.0 Compression Segments" in this document.

This document outlines considerations and best practices (a.k.a. use cases) for these "G.711.0 compression segments".

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 September 1, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Requirements Language	4
3.	ITU-T Rec. G.711.0 Background and Use in RTP	5
3.1.	G.711.0 Codec Background	5
3.2.	G.711.0 RTP Background	6
4.	G.711.0 "In The Middle" - Media Issues	8
4.1.	G.711.0 "In The Middle" - No RTP Header Compression	8
4.2.	G.711.0 "In The Middle" - With RTP Header Compression	11
4.3.	G.711.0 "In The Middle" - Implications for Voice Quality and Added Delay	12
4.4.	G.711.0 "In The Middle" - Multiplexing Multiple G.711 Flows	12
5.	G.711.0 "In The Middle" - Signaling Issues	14
6.	Acknowledgements	17
7.	Contributors	18
8.	IANA Considerations	19
9.	Security Considerations	20
10.	References	21
10.1.	Normative References	21
10.2.	Informative References	22
	Authors' Addresses	23

1. Introduction

The ITU-T (ITU-T) Recommendation G.711.0 [[G.711.0](#)] specifies a stateless and lossless compression for G.711 packet payloads typically used in Voice-over-IP (VoIP) networks.

This document outlines considerations and best practices for when ITU-T Rec. G.711.0 is used as a compression mechanism on one or more segments of an end-to-end ITU-T Rec. G.711 [[G.711](#)] negotiated session using the Real-Time Transport Protocol (RTP, [[RFC3550](#)]). Because RTP payload types (PT) for G.711 PCMU (0) and PCMA (8) are static PTs and because G.711.0 is both lossless and stateless, G.711.0-based compression can often times be employed on intermediate segments without access to session signaling. These properties allow G.711.0-based bandwidth savings without modifications to G.711 endpoints or G.711 call processing systems. Additionally, due to the lossless property of G.711.0, it may be employed multiple times on an end-to-end G.711 session with no loss of voice quality relative to G.711.

ITU-T Rec. G.711.0 and ITU-T Rec. G.711 may be referred to in this document simply as G.711.0 and G.711, respectively.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

3. ITU-T Rec. G.711.0 Background and Use in RTP

This document describes the use of ITU-T Rec. G.711.0 when it is employed as a lossless compression mechanism somewhere in between end systems where:

- 1) at least one of the media processing end systems has negotiated use of G.711, and
- 2) RTP (see [RFC 3550](#) [[RFC3550](#)] and [RFC 3551](#) [[RFC3551](#)]) is employed.

When used in this way, the G.711 payloads resulting from decompression and the corresponding G.711 RTP headers should "appear" to the media processing end systems that have negotiated G.711 as having been transported transparently (more detail later in [Section 4.2](#) ([Section 4.2](#))). This use case is referred to herein as "G.711.0 in-the-middle".

This section briefly describes ITU-T Rec. G.711.0 and its use in RTP.

3.1. G.711.0 Codec Background

ITU-T Rec. G.711.0 is a lossless and stateless compression mechanism for ITU-T Recommendation G.711 [[G.711](#)] and thus is not a "codec" in the sense of "lossy" codecs typically carried by RTP. When ITU-T Rec. G.711.0 is negotiated end-to-end as if it were a codec, the understanding is that ITU-T Rec. G.711.0 is losslessly encoding the underlying (lossy) ITU-T Rec. G.711 pulse code modulation (PCM) sample representation of an audio signal. For this reason ITU-T Rec. G.711.0 will be interchangeably referred to in this document as a "lossless data compression algorithm" or a "codec", depending on context. ITU-T Rec. G.711 and ITU-T Rec. G.711.0 will be referred to as G.711 and G.711.0, respectively. Likewise, within this document, individual G.711 PCM samples will be referred to as "G.711 symbols" or just "symbols" or "samples" for brevity.

When ITU-T Rec. G.711.0 is negotiated end-to-end as a codec, it is negotiated similarly to ITU-T Rec. G.711 and its RTP payload format specification is nearly identical to ITU-T Rec. G.711. This end-to-end use of ITU-T Rec. G.711.0 and the payload format for it is documented in the G.711.0 RTP payload format Internet Draft [[I-D.ramalho-payload-g7110](#)].

The fundamental design of G.711.0 resulted from the desire to losslessly encode and compress frames of G.711 symbols independent of what types of signals those G.711 frames contained. The primary G.711.0 use case is for G.711 encoded, zero-mean, acoustic signals

(such as speech and music), and for virtually every real-world signal significant compression is obtained. However, a significant property for G.711.0 is that it is lossless for any valid G.711 payload - even if the payload consisted of random G.711 symbols (many G.711-encoded modem or FAX payloads appear random). For this reason G.711.0 can be applied as a compression mechanism for any VoIP payload containing G.711 symbols without special consideration if those G.711 symbols came from FAX, modem, TTY or other "non-voice" applications. For detailed properties of the G.711.0 codec, see Section 3.2 of [\[I-D.ramalho-payload-g7110\]](#).

G.711.0, being both lossless and stateless, can be employed multiple times (e.g., on multiple, individual hops or series of hops) of a given flow with no degradation of quality relative to end-to-end G.711. Stated another way, multiple "lossless transcodes" from/to G.711.0/G.711 do not negatively affect voice quality as usually occurs with lossy transcodes to/from dissimilar codecs.

3.2. G.711.0 RTP Background

We note here that ITU-T Rec. G.711 is virtually always negotiated in RTP with a Payload Type (PT) of 0 (PCMU) or 8 (PCMA) because G.711 has a static payload type assignment and use of that static assignment is generally preferred. Payload types 0 and 8 should not be dynamically assigned to other codecs unless all dynamic payload types and unassigned payload types are already in use. Therefore, we have not seen RTP payload type 0 or 8 used in any network administrative domain for carrying other than a G.711 payload.

For this reason RTP packets with payload type 0 or 8 can usually be assumed to be PCMU or PCMA in most RTP settings. Thus compression from a G.711 payload to a G.711.0 payload can occur by:

- 1) noting payload type 0 or 8 is in use (input is assumed to be G.711), and
- 2) checking the input payload size in octets to ensure that it is an integer multiple of 40 (G.711.0 compression requires an integer multiple of 40 G.711 samples).

Because the lossless property of G.711.0, even if the payload presented to the G.711.0 encoder was not G.711, the G.711.0 lossless decoding process would produce the same exact payload as the payload input to the G.711.0 encoder.

Thus, due to the lossless property of G.711.0, network elements MAY apply G.711.0 compression to RTP payloads with payload type 0 or 8 (after check #2 above) and transport the payload as a G.711.0 payload

- knowing that upon decompression the same G.711 input payload would be output from the G.711.0 decoder.

Because G.711.0 may be employed (as a payload compression mechanism) on any hop or hops of an end-to-end G.711 flow and payload types of 0 or 8 can reasonably be assumed to be G.711, neither Session Description Protocol (SDP, [RFC 4566](#)) [[RFC4566](#)] signaling elements nor specific G.711.0 negotiation mechanisms will be mandated by this RFC. While this is true, SDP descriptions in the G.711.0 RTP Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)] MAY be used for a G.711.0 "in-the-middle" negotiation such as may occur in Session Border Controllers (SBCs) and the like; these cases are described below.

The following section describes media issues for these "G.711.0 in-the-middle" use cases. The section following that section, [Section 5](#) ([Section 5](#)), describes signaling implications for these "G.711.0 in-the-middle" use cases.

4. G.711.0 "In The Middle" - Media Issues

When G.711 has been negotiated end-to-end, G.711.0 can be employed by entities in the middle of the end-to-end G.711 flow as a compression mechanism. When used in this manner, this payload compression may be used with or without compression of the RTP header (e.g., cRTP [[RFC2508](#)], [[RFC5795](#)]). In either case, the G.711 payloads AND the corresponding G.711 RTP headers MUST appear to the end systems as having been transported transparently.

4.1. G.711.0 "In The Middle" - No RTP Header Compression

This figure below illustrates how the compression could be accomplished without RTP header compression.

G.711.0 Compression "In The Middle" - No RTP Header Compression Case

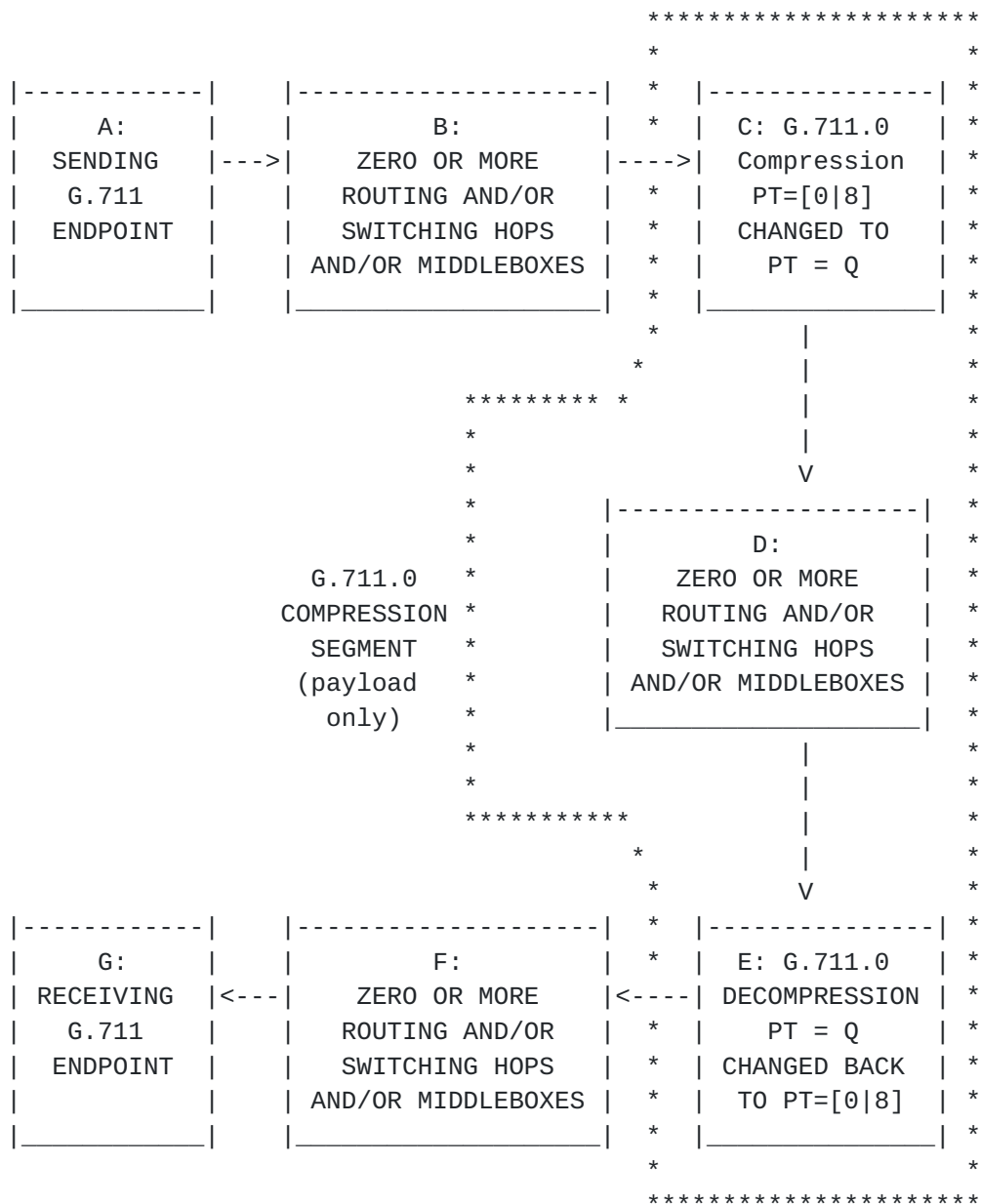


Figure 1

Figure 1 depicts G.711.0 compression in Box C and G.711.0 decompression back to G.711 in Box E. This figure depicts the case where only compression of the G.711 payload is desired; the RTP header (including any extensions) is simply copied with the exception that the G.711 payload type (the usual static PT of 0 or 8 is shown)

is replaced by a PT negotiated between Box C and Box E (depicted above as PT = Q).

Note that if there are no hops between Box C and E (i.e, no Box D), Figure 1 is equivalent to compression over a single link.

The compression segment represented by Box C, Box E and Box F is labeled a "G.711.0 compression segment" in the above figure.

Since G.711.0 is a lossless and stateless compression, there can be multiple such segments between the sending and receiving endpoints (not shown).

The G.711.0 compression and decompression (Box C and E) may reside in a variety of network elements such as, but not limited to, switches, routers, middleboxes (NATs/PATs, firewalls, session border controllers, transport acceleration devices) and is purposely not specified here.

In IP routed networks one cannot guarantee that the same physical element represented by Box C (the compressor) or Box E (the decompressor) will stay in the same IP routed path between Sending Endpoints A and Receiving Endpoint G. While this is true, we note that G.711.0 is a stateless compression and that as long as it is assured that some element in the topology can provide the functionality represented by Box C and Box E, the identical physical element need not be in the path. For example, if all ingress and egress routers in an enterprise WAN administrative domain had such functionality, it need not be the case that the same ingress or egress routers be traversed for every packet in the flow due to the statelessness of G.711.0-based compression.

In one possible design the payload type isn't changed at all (i.e., Q = original PCUM or PCMA PT) because a G.711.0 payload in number of octets is guaranteed to be different than the original G.711 payload size; this allows a RTP decoding process knowing the number of G.711 symbols to expect in the payload to infer that a G.711.0 representation of a G.711 payload is present. The interested reader will note that while a G.711.0 payload representation is usually much smaller than the uncompressed G.711 payload, an "uncompressible payload" is actually one octet larger (see ITU-T Rec. G.711.0 [[G.711.0](#)] specification).

If the RTP payload type to use (Q) is negotiated via session signaling, the method by which G.711.0 compression segment endpoints negotiate that payload type is not mandated in this document. However, the SDP descriptions in the G.711.0 RTP Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)] MAY be used for such a

negotiation and this point is addressed in [Section 5](#).

In designs where Q is other than the original PCMU or PCMA PT and is not negotiated via session signaling, Q MUST be outside of the range of dynamic PT assignment and it is RECOMMENDED that Q be chosen from a static PT that is known to never be assigned within the scope of the G.711.0 compression segment or from the range of unassigned PTs [[RFC3551](#)] that are otherwise known to be free from conflict within the system design. We note here that the PT Q should never be seen by the end systems nor by any element outside of the G.711.0 compression segment; the specification for the choice of Q here reflects an abundance of caution for the case where a rogue RTP packet is not successfully processed by Box E functionality.

If the RTP payload type to use (Q) is configured, the method by which the G.711.0 compression segment endpoints are configured is outside the scope of this document.

In another possible design, Box C and Box E might be configured to be tunnel endpoints in a design where functionality represented by Box C (and possibly Box E) are known to be in the end-to-end path. Such functionality is also outside the scope of this document.

There may be many "potential G.711.0 compression/decompression points" along the end-to-end G.711 flow; the mechanisms by which certain entities determine that they should perform G.711.0-based compression and decompression are outside the scope of this document.

G.711.0 payloads, like G.711 payloads, may be encrypted by encryption protocols such as the Secure Real-time Transport Protocol (SRTP) [[RFC3711](#)], however the mechanisms by which the keys are exchanged or negotiated are outside the scope of this document. Mechanisms such as those used when G.711 encryption is employed MAY be used. Additionally, the security considerations of using G.711.0 SHOULD be considered in these "G.711.0 in-the-middle" applications (see Internet Draft [[I-D.ramalho-payload-g7110](#)]).

Network elements such as firewalls, NATs, SBCs, etc. that may exist in the path of the G.711.0 packets (Box D). These elements may drop packets containing unexpected payload types; therefore these elements may need additional configuration and/or signaling knowledge to let the compressed G.711.0 packets through. Mechanisms to do this are also outside the scope of this document.

[4.2.](#) G.711.0 "In The Middle" - With RTP Header Compression

When it is desired to compress the G.711 header as well, the G.711.0 compression segment endpoints of the previous section have further

functionality by which they also compress the headers. However, this functionality and the negotiation of same is outside of the scope of this document.

We note here that if such RTP header compression functionality is employed, that the G.711 payloads AND the corresponding G.711 RTP headers MUST appear to the end systems as having been transported transparently. That is, RTP header fields such as sequence numbers and timestamps need not necessarily be identical, but the differences between the input and output fields should be such that the receiving end system "cannot tell" that they were modified (differences by a constant in modulo send timestamp units for example).

Any RTP header compression functionality SHOULD be stateless so as to minimize error propagation for lost packets to be consistent with the G.711.0 design goal of no error propagation due to lost packets (see G.711.0 RTP Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)] Attribute A3).

4.3. G.711.0 "In The Middle" - Implications for Voice Quality and Added Delay

G.711.0, being both lossless and stateless, can be employed multiple times on an end-to-end G.711 flow (e.g., on multiple, individual hops or series of hops). If RTP headers are not compressed or stateless RTP header compression is employed (as recommended in [Section 4.2](#) ([Section 4.2](#))), then there is no error propagation owing to a loss of a G.711.0 packet. That is, the impact of an individual packet drop of a G.711.0 RTP packet is identical to the impact of a drop of the corresponding G.711 RTP packet.

Stated another way, multiple "lossless transcodes" from/to G.711.0/ G.711 do not negatively affect voice quality as may occur with lossy transcodes to/from dissimilar codecs.

G.711.0 provides over 50% reduction in average payload size with exactly 0.0000% quality loss relative to G.711 [[ICASSP](#)].

For completeness we note that a G.711.0 encode/decode average complexity is 1 WMOPS (see Internet Draft [[I-D.ramalho-payload-g7110](#)] [Section 3.2](#), Attribute A8). Given such low complexity, less than 1 ms of compression/decompression of additional delay per each G.711.0 compression segment is expected in most implementations.

4.4. G.711.0 "In The Middle" - Multiplexing Multiple G.711 Flows

G.711.0 may also be desired to multiplex the payloads of many G.711 channels into one "G.711.0 payload" in a multiplex RTP packet.

If all the G.711 channels to be multiplexed have the same number of G.711 symbols in each individual source G.711 payload, as is the case in many "G.711 VoIP trunks", a straightforward way to parse the G.711.0 payload into individual G.711 payloads would be the methodology described in [Section 4.2.2](#) in the G.711.0 Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)]. While this is possible, there are subtleties to such an approach such as what to do when the *i*th channel is unavailable due to an input packet drop. A straightforward way to address this issue is to have a dynamic mapping carried in side information, such as a RTP header extension, which has the capability to add or drop channels "on-the-fly".

Alternatively, specialized tunneling mechanisms, such as WAN optimization tunneling, can be used to convey such dynamic mapping between input and output G.711 channels.

Owing to these architectural options, the specification of such mechanisms is outside the scope of this document.

Similarly to [Section 4.2](#) ([Section 4.2](#)) above, the de-multiplexing process producing individual G.711 output RTP packets MUST produce G.711 RTP headers that appear to the end systems as having been transported transparently. The mechanisms used are also outside the scope of this document.

Any RTP header multiplexing functionality SHOULD be stateless so as to minimize error propagation for lost packets to be consistent with the G.711.0 design goal of no error propagation due to lost packets (G.711.0 RTP Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)] Attribute A3).

5. G.711.0 "In The Middle" - Signaling Issues

This section describes a G.711.0 use case in which G.711 is negotiated end-to-end and:

- 1) there exists one or more places in the end-to-end path where the media is terminated and re-initiated, and
- 2) one or more of these "media segments" contained in the end to end path desires to compress the G.711 payloads to G.711.0 format (or vice versa).

Session Border Controllers (SBCs) and Media Termination Points (MTPs) are two examples of places where this could occur.

Figure 2 provides an illustration for such a case where SBCs are used as an example. This figure also assumes, as an example, that the Session Initiation Protocol (SIP, [[RFC3261](#)]) is used to set up the session and SDP was employed to negotiate the end-to-end codec.

G.711.0 Signaling Post End-to-End G.711 Negotiation

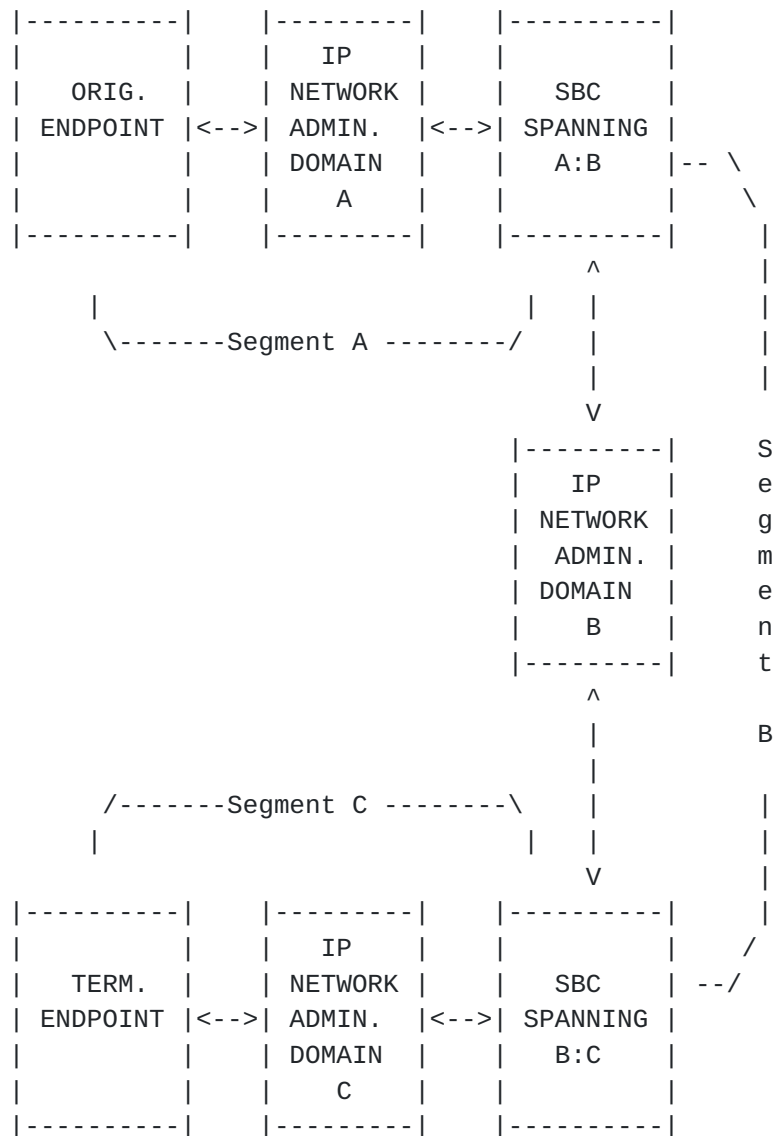


Figure 2

This figure represents an end-to-end session between the originating and terminating endpoints where there are two SBCs in the end-to-end path. The end-to-end path is depicted to be in three segments, labeled A, B and C, and without loss of generality the IP Networks in these segments are labeled administrative domain A, B and C (although A, B or C may be part of the same administrative domain).

Here we assume that G.711 is the preferred codec end-to-end. If all

points where media could be terminated had G.711.0 capability, it is highly likely that G.711.0 would have been negotiated from the originating endpoint to the terminating endpoint. The reason for this is that G.711.0 losslessly conveys G.711 and G.711.0 would likely have been preferred over G.711 in the original negotiation.

However, if one or more points where media could be terminated did not have G.711.0 capability, the end-to-end call would likely have been negotiated as G.711. This is the case depicted here. In this case there is an opportunity for any segment to renegotiate G.711.0 on its segment, and compressing to/from G.711 on the segments that remain G.711.

The RTP payload format for G.711.0 is in Internet Draft [[I-D.ramalho-payload-g7110](#)]. Specifically we note that the payload format for G.711.0 is nearly identical to G.711 in that the timestamp units are identical and most other elements are also identically assigned (the major exception is the PT). Thus, the G.711.0 payload format makes it trivial simply to change the PT and the payload of a G.711 RTP packet to a G.711.0 packet and the converse. In other words, the compression of the payload and the translation of the RTP headers to/from G.711/G.711.0 may be performed "on-the-fly" in the middle of an end-to-end G.711 session with no voice quality degradation (relative to G.711) and concurrently obtaining all the compression benefits of G.711.0.

For this case, the SDP parameters defined in the G.711 Payload Format specification MAY be used for renegotiation to G.711.0 by any SIP UA facing one of these segments without signaling these changes end-to-end. We note here that SBCs also have signaling functionality and are typically implemented as SIP Back-to-Back User Agents (B2BUAs). From the perspective of the Originating Endpoint, the SIP signaling termination is the UA at the first SBC (i.e., not the Terminating Endpoint). Thus, for the case where G.711.0 is renegotiated only on Segment A, the signaling for that codec change is not propagated to the Terminating Endpoint. The end result is that the terminating endpoint "believes" it is participating in an end-to-end G.711 session and the resulting voice quality is identical to that of an end-to-end G.711 session (i.e., there is no need for it to know that a lossless compression had taken place).

6. Acknowledgements

There have been many people contributing to G.711.0 in the course of its development. The people listed here deserve special mention: Takehiro Moriya, Claude Lamblin, Herve Taddei, Simao Campos, Yusuke Hiwasaki, Jacek Stachurski, Lorin Netsch, Paul Coverdale, Patrick Luthi, Lei Miao, Paul Barrett, Jari Hagqvist, Pengjun (Jeff) Huang, and Jon Gibbs.

7. Contributors

The authors thank everyone who have contributed to this document.
The people listed here deserve special mention: Paul, Jones, Ali
Begen, and Roni Even.

8. IANA Considerations

This RFC is informational and contains no elements requiring IANA registration.

9. Security Considerations

This RFC is informational. The security considerations surrounding the use of G.711.0 (including uses described in this document) are described in the security considerations section of the G.711.0 RTP Payload Format Internet Draft [[I-D.ramalho-payload-g7110](#)].

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", [RFC 2508](#), February 1999.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, [RFC 3551](#), July 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", [RFC 3711](#), March 2004.
- [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.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObusT Header Compression (ROHC) Framework", [RFC 5795](#), March 2010.
- [I-D.ramalho-payload-g7110] Ramalho, M., Jones, P., Harada, N., Perumal, M., and M. Lei, "RTP Payload Format for G.711.0", [draft-ramalho-payload-g7110-00](#) (work in progress), June 2011.
- [G.711.0] ITU-T G.711.0, "Recommendation ITU-T G.711.0 - Lossless Compression of G.711 Pulse Code Modulation", September 2009.
- [G.711] ITU-T G.711.0, "Recommendation ITU-T G.711 - Pulse Code Modulation (PCM) of Voice Frequencies", November 1988.

10.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", [RFC 2629](#), June 1999.
- [ICASSP] N. Harada, Y. Yamamoto, T. Moriya, Y. Hiwasaki, M. A. Ramalho, L. Netsch, Y. Stachurski, Miao Lei, H. Taddei, and Q. Fengyan, "Emerging ITU-T Standard G.711.0 - Lossless Compression of G.711 Pulse Code Modulation, International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, ISBN 978-1-4244-4244-4295-9", March 2010.

Authors' Addresses

Michael A. Ramalho (editor)
Cisco Systems, Inc.
4563 Tuscana Drive
Sarasota, FL 34241
USA

Phone: +1 919 476 2038
Email: mramalho@cisco.com

Dan Wing
Cisco Systems, Inc.
821 Alder Drive
Milpitas, CA 95035
USA

Phone: +1 408 853 4197
Email: dwing@cisco.com

Muthu Arul Mozhi Perumal
Cisco Systems, Inc.
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Phone: +91 9449288768
Email: mperumal@cisco.com

Noboru Harada
NTT Communications Science Labs
3-1 Morinosato-Wakamiya
Atsugi, Kanagawa 243-0198
JAPAN

Email: harada.noboru@lab.ntt.co.jp

Hadriel Kaplan
Acme Packet
71 Thirs Avenue
Burlington, VT 01803
USA

Email: hkaplan@acme.com