

Network Working Group	G. Zorn, Ed.	
Internet-Draft	Network Zen	
Intended status: Standards Track	Y. Wang	
Expires: June 13, 2011	Huawei Technologies	
	A. Lakaniemi	
	Nokia	
	December 10, 2010	

[TOC](#)

RTP Payload Format for G.718 Speech/Audio draft-ietf-avt-rtp-g718-05.txt

Abstract

This document specifies the Real-Time Transport Protocol (RTP) payload format for the Embedded Variable Bit-Rate (EV-VBR) speech/audio codec, specified in ITU-T G.718. A media type registration for this RTP payload format is also included.

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 June 13, 2011.

Copyright Notice

Copyright (c) 2010 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

- [1.](#) Introduction
- [2.](#) Requirements Language
- [3.](#) Background
 - [3.1.](#) The G.718 Codec
 - [3.2.](#) Benefits of Layered Design
 - [3.3.](#) Transmitting Layered Data
 - [3.4.](#) Scaling Scenarios and Rate Control
- [4.](#) G.718 RTP Payload Format
 - [4.1.](#) Payload Structure
 - [4.1.1.](#) Payload Header
 - [4.1.2.](#) G.718 Transport Blocks
 - [4.2.](#) Handling The Encoded Data
 - [4.3.](#) G.718 Scaling
 - [4.4.](#) CRC Verification
 - [4.5.](#) G.718 Session
 - [4.6.](#) Cross-stream/Cross-layer Timing Synchronization
 - [4.7.](#) RTP Header Usage
- [5.](#) Payload Format Parameters
 - [5.1.](#) Media Type Registration
 - [5.2.](#) Mapping to SDP Parameters
 - [5.3.](#) Offer/Answer Considerations
 - [5.4.](#) Declarative Usage of SDP
 - [5.5.](#) SDP Examples
 - [5.5.1.](#) Example 1
 - [5.5.2.](#) Example 2
 - [5.5.3.](#) Example 3
- [6.](#) Congestion Control
- [7.](#) Security Considerations
- [8.](#) IANA Considerations
- [9.](#) Acknowledgements
- [10.](#) References
 - [10.1.](#) Normative References
 - [10.2.](#) Informative References
- [Appendix A.](#) Payload Examples
 - [A.1.](#) Simple Payload Examples
 - [A.1.1.](#) All The Layers in The Same Payload
 - [A.1.2.](#) Layers in Separate RTP Streams
 - [A.2.](#) Advanced Examples
 - [A.2.1.](#) Different Update Rate for Subset of Layers
 - [A.2.2.](#) Redundant Frames With Limited Set of Layers

1. Introduction

[TOC](#)

The International Telecommunication Union (ITU-T) Recommendation G.718 [[ITU.G718.2008](#)] ([International Telecommunications Union, "Frame Error Robust Narrowband and Wideband Embedded Variable Bit-Rate Coding of Speech and Audio from 8-32 Kbit/s," May 2008.](#)) specifies the Embedded Variable Bit Rate (EV-VBR) speech/audio codec. This document specifies the Real-time Transport Protocol (RTP) [[RFC3550](#)] ([Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.](#)) payload format for this codec.

2. Requirements Language

[TOC](#)

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](#)] ([Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.](#)).

3. Background

[TOC](#)

3.1. The G.718 Codec

[TOC](#)

G.718 is an embedded variable rate speech codec having a layered design. The bitstream of the G.718 core codec consists of a core layer, denoted as L1, and four enhancement layers, denoted as L2-L5. The bit-rates of the G.718 core codec range from 8 kbit/s (core layer only) to 32 kbit/s (with all layers up to L5). Furthermore, the G.718 codec also supports discontinuous transmission (DTX) and comfort noise generation (CNG) by sending Silence Descriptor (SID) frames during periods of non-active input signal, resulting in a reduced bit-rate. The sampling frequency of the core codec is 16 kHz and the codec operates on 20 ms frames. The G.718 codec is also capable of narrowband operation with audio input and/or output at 8 kHz sampling frequency.

While transmitting/receiving the core layer L1 is enough for successful decoding of the audio content, each of the enhancement layers Ln (n being 2 to 5, inclusive) provides an improvement to reconstructed audio quality. Thus, the core layer ensures the basic

communication while the enhancement layers can be used to improve the perceptual quality. Furthermore, enhancement layers are dependent on all the lower layers in a sense that successful decoding of layer L_n requires also all the layers L_m with $m < n$ to be available.

The sizes, sampling rates and possible outputs of the G.718 core codec layers L1-L5 are summarized in Table 1 below, where the "Bytes" column indicates the number of bytes per encoded data unit for a layer. NB and WB denote narrowband and wideband, respectively. The "Bytes" column in other tables has the same meaning. Note that for layers L1 and L2, the corresponding output may either be NB or WB, depending on the rendering device and the application requirement, regardless of the sampling rate of the encoded data.

Table 1: G.718 Layers

Layer	Bytes	Cumulative bit-rate	Sampling rate	Output
L1'	32	12.8 kbit/s	16 kHz	WB
L3'	9	16.4 kbit/s	16 kHz	WB
L4	20	24.4 kbit/s	16 kHz	WB
L5	20	32.4 kbit/s	16 kHz	WB

The G.718 codec also includes an operating mode that is compatible with the Adaptive Multi-Rate Wideband (AMR-WB) codec [\[AMR-WB\] \(3GPP, "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband \(AMR-WB\) speech codec; General description," April 2001.\)](#), for which the RTP payload format is specified in [\[RFC4867\] \(Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate \(AMR\) and Adaptive Multi-Rate Wideband \(AMR-WB\) Audio Codecs," April 2007.\)](#). In this AMR-WB interoperable mode, layers L1 and L2 are replaced by L1' consisting of AMR-WB encoded data and L3' is used instead of L3. The usage of layers L4 and L5 is not affected by transmitting AMR-WB data in the lower layers. If layer L3' is present in the encoded bit-stream, the base layer L1' must use the AMR-WB mode 2 with a bit-rate of 12.65 kbits/s. Otherwise (the encoded bit-stream contains only the L1' layer), any of the 9 AMR-WB coding modes 0, 1, 2, 3, 4, 5, 6, 7, and 8 correspond to the bit-rates of 6.60, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, and 23.85 kbit/s, respectively, may be in use. Table 2 summarizes the AMR-WB interoperable mode when more than one layer may be present.

Table 2: G.718 layers in the AMR-WB interoperable mode

Layer	Bytes	Cumulative bit-rate	Sampling rate	Output
L1'	32	12.8 kbit/s	16 kHz	WB
L3'	9	16.4 kbit/s	16 kHz	WB
L4	20	24.4 kbit/s	16 kHz	WB
L5	20	32.4 kbit/s	16 kHz	WB

Note that the bit-rate for the raw bit-stream of AMR-WB mode 2 is 12.65 kbits/s. However, after counting the padding bits to make each encoded data unit byte-aligned, as in the octet-aligned mode specified in [\[RFC4867\] \(Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate \(AMR\) and Adaptive Multi-Rate Wideband \(AMR-WB\) Audio Codecs," April 2007.\)](#), the resulting bit-rate is then 12.8 kbits/s.

In the AMR-WB interoperable mode, when the base layer L1' is transported in its own RTP packet stream, the packetisation specified in [\[RFC4867\] \(Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate \(AMR\) and Adaptive Multi-Rate Wideband \(AMR-WB\) Audio Codecs," April 2007.\)](#) MUST be used, to enable legacy RFC4867 receivers to receive the base layer L1'.

ITU-T SG16 is currently working on a set of extension layers in order to provide so-called super-wideband (SWB) audio and stereophonic encoding extensions on top of the G.718 core codec. Further details and the usage of these layers are undetermined at this time.

The main application of the G.718 codec is telephony. Other expected applications include audio/video conferencing and streaming.

3.2. Benefits of Layered Design

[TOC](#)

Layered design enables simple scalability of the transmitted stream simply by conveying a suitable number of layers. The number of layers used in a session may be selected for example based on the capacity of the transmission channel, current transmission conditions, characteristics of the source signal or available processing capacity.

Another obvious benefit of the layered codec design is the possibility to exploit the scalability to support congestion control by transmitting/dropping some of the (higher) enhancement layers in order to alleviate congestion in the network. See more detailed discussion on the congestion control in section 6.

Furthermore, the layered design also implicitly provides possibility

for unequal error detection/protection by employing different levels of protection on core layer and enhancement layers.

3.3. Transmitting Layered Data

[TOC](#)

In principle there are two basic approaches to carry the data from a layered encoder:

1. All the layers are carried within a single RTP session
2. The encoded data is divided over multiple RTP sessions, each session carrying a subset of layers. This is also referred to as Multi-Session Transmission (MST)

The first choice is the most efficient in terms of exploitation of transmission bandwidth. Furthermore, using only one packet to carry all encoded data layers of a frame requires less resources also from the end-systems (and intermediate systems) since the number of packets is kept at minimum and only single RTP packet stream needs to be handled. However, this option requires any intermediate network element performing the scaling operation to be fully media-aware since removing encoded layers requires modification of the payload. Furthermore, the intermediate network element needs to be within the security context to enable the meaningful manipulation of the payload, in case secure transport is employed. This might not be feasible in all systems/scenarios, but some special-purpose devices such as e.g. media gateways in cellular telephone systems may be able to implement this kind of media-aware functionality.

The second alternative, transmitting selected subsets of layers in separate RTP sessions, facilitates simple scalability in intermediate network elements without the requirement of being fully media-aware. One use case of this alternative is layered multicast [[McCanne](#)] ([McCanne, S., Jacobson, V., and M. Vetterli, "Receiver-driven layered multicast," October 1996.](#)). On the other hand, this approach introduces separate packet header overhead for each subset of layers for those low-delay application scenarios wherein aggregation of data from multiple frames is not ideal. In this case, when the size of the encoded data block per single layer is in the range of 10 to 20 bytes, the packetisation may result in relatively high amount of protocol overhead, which might be an expensive solution on bandwidth-limited links. Another drawback of this approach is somewhat more complex session setup and the additional complexity associated with handling of several concurrent RTP sessions. However, this is a trade-off that enables simple scalability also by intermediate network elements that are not aware of the details of the transmitted media.

3.4. Scaling Scenarios and Rate Control

[TOC](#)

In principle there are three different ways to make use of the layered design to control the bandwidth usage:

1. A sender decides to change the number of layers it is transmitting (for example due to congestion control constraints)
2. A receiver or an intermediate network element instructs a sender to change the number of layers it is transmitting
3. An intermediate network element passes through only a subset of layers it receives

The most appropriate mechanism depends on the application and the employed network topology. For example point-to-point conversational audio connection can easily introduce rate control by changing the number of transmitted layers, while in centralized audio/video conferencing scenario the conference server is a more appropriate point to implement the rate control instead of transmitting end-point. Please refer to RFC 5117 for extensive discussion on the different topologies and their implications to the transmission. However, the fundamental difference between these choices is that method 1 does not necessarily need any feedback from the receiver(s), while methods 2 and 3 require a signaling mechanism to support rate control.

4. G.718 RTP Payload Format

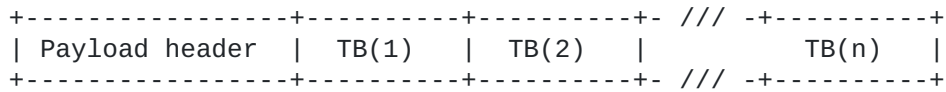
[TOC](#)

The basic G.718 source data unit is one layer of an encoded frame. Since generally the term layer refers to time series of data representing certain encoding layer, in this specification we use the term Encoded Data Unit (EDU) to refer to a single layer of data from single encoded frame. Thus, each EDU has a (conceptual) frame number indicating its location in encoding/decoding order and a layer number indicating the encoding layer the EDU represents.

4.1. Payload Structure

[TOC](#)

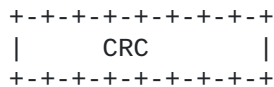
The G.718 payload format consists of a payload header, followed by one or more transport blocks (TB) forming the actual payload data.



4.1.1. Payload Header

[TOC](#)

The payload header consists of an 8-bit payload CRC checksum:



On the transmitting end the payload checksum is computed over the primary transport block (specified in [Section 4.1.2 \(G.718 Transport Blocks\)](#)) of the payload using the generator polynomial

$$C(z) = z^8 + z^4 + z^3 + z^2 + 1$$

Subsequent transport blocks are prepared in such a way that the payload checksum is valid for any integer number of contiguous transport blocks within one RTP packet starting from the beginning of the primary transport block.

On the receiving end the payload CRC checksum can be used to verify the correct reception of any contiguous subset of transport blocks within one RTP packet starting from the beginning of the primary transport block (see [Section 4.4 \(CRC Verification\)](#) for a detailed description).

4.1.2. G.718 Transport Blocks

[TOC](#)

The basic building block of the G.718 RTP payload data is an G.718 transport block (TB). There are two types of transport blocks: primary and secondary.

The structure of the primary transport block is depicted below.



The structure of the secondary transport block is depicted below.



The layer ID (L-ID) and the NF fields form the transport block header. The L-ID field is used to identify the layer structure of the encoded data carried in this G.718 transport block, and the NF field indicates the number of encoded frames with this layer structure carried in the Encoded data part following the transport block header. The Tail field of the secondary transport block carries a modified 8-bit CRC checksum computed over the transport block, as specified below.

A G.718 RTP packet payload SHALL include exactly one primary transport block, which MAY be followed by one or more secondary transport blocks. The data fields of both transport block types are described below.

L-ID (6 bits) Identification of the encoded data carried in this transport block. Table 3 below specifies the mapping between L-ID and the encoded data. Note that L-ID is treated as an unsigned integer.

Table 3: Layer Identification (L-ID) Values

L-ID	Encoded data
0	Empty frame
1	L1
2	L1-L2
3	L1-L3
4	L1-L4
5	L1-L5
6	L2
7	L2-L3
8	L2-L4
9	L2-L5
10	L3
11	L3-L4
12	L3-L5
13	L4
14	L4-L5
15	L5
16	L1'
17	L1', L3'
18	L1', L3', L4
19	L1', L3', L4-L5
20	G.718 SID
21	AMR-WB SID
22-63	Reserved

NF (2 bits) Number of frames in this transport block (2 bits) decreased by one. The number of frames is equal to the value of NF incremented by one. For example, value NF=0 indicates

that the transport block carries one frame, and value NF=3 indicate that the transport block carries four frames. If the sender wants to encapsulate more than four frames per payload, several transport blocks need to be used.

Encoded Data (variable length) Encoded data consists of EDUs as specified by the values L-ID and NF fields, arranged according to the rules given in [Section 4.2 \(Handling The Encoded Data\)](#). When L-ID is equal to 0 (empty frame), the encoded data field is not present.

Tail (8 bits) The Tail field of the secondary transport block carries a bit field that is needed to modify the partial CRC checksum over the payload data up to the end of this TB to match the payload CRC field value carried in the payload header.

In the transmitter the Tail bits for a secondary TB(n) are computed by first computing the CRC checksum CRC(n) over the payload data from the beginning of the primary TB up to the end of TB(n) using the generator polynomial C(z) given above. The bits of the Tail field of TB(n) are set to zero value for the CRC computation. The transmitted value of the Tail field in TB(n) is obtained by bitwise XOR operation between the payload CRC field value carried in the payload header and the CRC(n) computed for TB(n).

4.2. Handling The Encoded Data

[TOC](#)

In order to provide unique mapping of EDUs to encoded frames, the following rules on sequence of frames and sequence of layers need to be followed when creating a payload:

*The frames within a payload MUST form a set of contiguous frames in decoding order, i.e. if a payload carries frames n and n+N, all frames between n and n+N in decoding order MUST also be present in the payload.

*The layers within a frame MUST form a contiguous set of layers, i.e. if layers Lx and Ly of a frame are included in the payload, all layers between Lx and Ly layers MUST also be present.

The EDUs within a transport block are arranged according to the following rules:

*The EDUs within a transport block MUST be arranged in increasing order of layer number

*The EDUs with the same layer number within a transport block MUST be arranged in decoding order

Explicit timing information for the transport blocks is not needed, since the ordering of EDUs in the payload and their mapping to transport blocks can be used to implicitly carry this information. The following rules apply:

*If the highest layer carried in transport block k is n, and the lowest layer carried by transport block k+1 is n+1, then the EDUs of transport block k and k+1 belong to the same encoded frame. Furthermore, if transport blocks k and k+1 carry EDUs belonging to the same encoded frame(s), these transport blocks MUST include the same number of EDUs

*If the highest layer carried in transport block k is n, and the lowest layer carried by transport block k+1 is smaller than or equal to n, the EDUs of transport block k and k+1 belong to the two separate encoded frames, which are contiguous in decoding order

*Multiple copies of an EDU MUST NOT be included in the payload

A set of EDUs can be allocated to transport blocks in several ways. For example each EDU can be encapsulated in its own transport block, all EDUs can be carried in single transport block, EDUs belonging to the same encoded frame can be encapsulated in dedicated transport block, or EDUs representing the same layer can be carried in their own transport blocks. Three examples on this with two frames with layers L1-L3 are given below. The first example illustrates the case using a single transport block for the whole payload, while the second payload example introduces separate transport blocks for each of the EDUs. The third example shows an approach where all layers are carried in dedicated transport blocks. The notation Fx-Ly is used to denote layer y of frame x.

Example 1: All EDUs in a single transport block

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| L-ID=3 |NF=1 | F1-L1 | F2-L1 | F1-L2 | F2-L2 | F1-L3 | F2-L3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Example 2: All EDUs in separate transport blocks

```

+-----+-----+-----+-----+-----+-----+
| L-ID=1 |NF=0 | F1-L1 | L-ID=1 |NF=0 | F2-L1 |
+-----+-----+-----+-----+-----+-----+
| L-ID=8 |NF=0 | F1-L2 | L-ID=8 |NF=0 | F2-L2 |
+-----+-----+-----+-----+-----+-----+
| L-ID=14 |NF=0 | F1-L3 | L-ID=14 |NF=0 | F2-L3 |
+-----+-----+-----+-----+-----+-----+

```

Example 3: Dedicated transport for EDUs of each layer

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| L-ID=1 |NF=1 | F1-L1 | F2-L1 | L-ID=6 |NF=1 | F1-L2 | F2-L2 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| L-ID=10 |NF=1 | F1-L3 | F2-L3 |
+-----+-----+-----+-----+-----+-----+

```

While the first example carrying data from all layers in the same transport block obviously consumes less bandwidth, the second example using separate transport block for each EDU, and the third example using dedicated transport blocks for each layer provide simple scaling possibility: while in the first case the removal of e.g. layer L3 (from each frame in the payload) would require changing the value of the L-ID in addition to removing the corresponding EDU(s), in the second and third options it is enough to just remove all transport blocks carrying L3 data and the remaining part of the payload can be left untouched (however the packet size information in high-layer protocol headers needs change).

4.3. G.718 Scaling

[TOC](#)

Some Media-Aware Network Elements (MANEs) MAY modify the G.718 bitstream by dropping some of the layers in case congestion control or e.g. access link bandwidth requires such scaling to take place. Such MANEs are RTP translators (with the topology Topo-Translator as described in [\[RFC5117\] \(Westerlund, M. and S. Wenger, "RTP Topologies," January 2008.\)](#), for which the rules for RTP translators specified in [\[RFC3550\] \(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\)](#) apply.

A payload can be either completely dropped or some of the transport blocks it carries can be discarded. In case full payloads are dropped to implement scaling, a packet containing the core layer L1 SHOULD NOT be discarded, since the decoding of higher layers of the same encoded frame is not possible without the core layer data being available. This means that payloads with L-ID values equal to 1 to 5, inclusive and 16 to 19, inclusive, SHOULD NOT be completely discarded.

Author's note: To be checked whether the case of dropping a subset of the transport blocks in one packet also strictly follows the topology Topo-Translator.

In case the payload is forwarded with modified content, at least the primary transport block MUST be preserved in the payload, while some of the secondary transport blocks at the end of the payload MAY be discarded.

4.4. CRC Verification

[TOC](#)

Both UDP-Lite [[RFC3828](#)] (Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)," July 2004.) and DCCP [[RFC4340](#)] (Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," March 2006.) provide partial checksum options, in which partially damaged payloads can be delivered to the application layer. In cases wherein such a transport layer operation is in use, and the partial checksum service by the transport layer protects up to the RTP header and the payload header, the CRC checksum provided in the payload header can be used to verify whether an RTP packet payload contains corrupt transport blocks.

On the receiving end the CRC verification is made in such a way that the CRC computation is started from the beginning of the primary TB, i.e. from the MSB of the first octet of the TB(1), and the computation is continued until the end of the payload data or until an erroneous TB is encountered. At the end of each TB a check MAY be performed: if the CRC value at the end of TB(n) matches the payload CRC value received in the payload header, the verification is successful and the data up TB(n) is valid. If the CRC value at the end of TB(n) does not match the payload CRC value received in the payload header, there is an error in the TB(n) and it MUST be discarded as corrupted. Furthermore, if the verification indicates corrupted TB(n), all subsequent transport blocks TB(m) with $m > n$ MUST also be discarded.

4.5. G.718 Session

[TOC](#)

A G.718 session consists of one or several RTP sessions carrying G.718 data encoded according to the payload format specified in [Section 4.1 \(Payload Structure\)](#).

4.6. Cross-stream/Cross-layer Timing Synchronization

[TOC](#)

In the case where a G.718 session consists of multiple RTP sessions, the RTP packets transmitted on separate RTP sessions need to be synchronized in order to enable reconstruction of the frames in the

receiving end [\[RFC6051\] \(Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows," November 2010.\)](#). Since each of the RTP sessions uses its own random initial value for the RTP timestamp, there is also a random offset between the RTP timestamps values carrying the EDUs belonging to the same encoded frame in different RTP sessions.

The receiver MUST use the traditional RTCP-based mechanism to synchronize streams by using the RTP and NTP timestamps of the RTCP Sender Reports (SR) it receives [\[RFC3550\] \(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\)](#).

4.7. RTP Header Usage

[TOC](#)

This section specifies the usage of some fields of the RTP header (specified in Section 5 of [\(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\) \[RFC3550\]](#)) with the G.718 RTP payload format. The settings for other RTP header fields are as specified in [\[RFC3550\] \(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\)](#).

The RTP timestamp corresponds to the sampling instant of the first encoded sample of the earliest frame in the payload. The timestamp clock frequency is 32 kHz.

The marker bit (M) of each of the RTP streams of the session SHALL be set to value 1 if the payload carries an EDU belonging to the first frame after an inactive period, i.e. an EDU from the first frame of a talkspurt. For all other packets the marker bit is set to value 0.

5. Payload Format Parameters

[TOC](#)

This section defines the parameters that may be used to configure optional features in the G.718 RTP transmission.

The parameters are defined here as part of the media subtype registration for the G.718 codec. Mapping of the parameters into the Session Description Protocol (SDP) [\[RFC4566\] \(Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol," July 2006.\)](#) is also provided for those applications that use SDP. In control protocols that do not use MIME or SDP, the media type parameters MUST be mapped to the format used with that control protocol.

5.1. Media Type Registration

[TOC](#)

This registration is done using the template defined in RFC 4288 [[RFC4288](#)] ([Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures," December 2005.](#)) and following RFC 4855 [[RFC4855](#)] ([Casner, S., "Media Type Registration of RTP Payload Formats," February 2007.](#)).

Type name: audio

Subtype name: G718

Required parameters: none

Optional parameters: This parameter MAY be used to indicate **mode:** whether the mode with layer L1 being present or the AMR-WB compatible mode (with layer L1' being present) is in use. If this parameter is not present or the value of this parameter is equal to 0, the mode with layer L1 being present is in use. Otherwise, the AMR-WB compatible mode is in use. When this parameter is present, the value MUST be either 0 or 1.

NOTE: When the upcoming stereo and SWB options are present, the semantics of this parameter may change.

layers: The numbers of the layers (in range from 1 to 5, denoting layers from L1 to L5, respectively) transmitted in this session, expressed as comma-separated list of layer numbers. If the parameter is present, at least layer L1 or L1' MUST be included in the list of layers in one of the RTP sessions included in the G.718 session. If the parameter is not present, all layers up to layer L5 MAY be used in the session.

NOTE: Why not use semantics similarly as L-ID?

ptime: The recommended length of time (in milliseconds) represented by the media in a packet. [See Section 6 of \(Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol," July 2006.\)](#) [[RFC4566](#)].

maxptime: The maximum length of time (in milliseconds) that can be encapsulated in a packet. [See Section 6 of \(Handley, M., Jacobson,](#)

[V., and C. Perkins, "SDP: Session Description Protocol," July 2006.](#)) [RFC4566].

NOTE: Some further study is needed to see if separate parameters for sending and receiving capabilities/preferences are needed -- especially for upcoming stereo and SWB options.

NOTE: Support for upcoming SWB and stereo options needs to be taken into account. Basically we can either 1) extend the parameter "layers" to cover also this aspect, or 2) define separate parameter(s) for these new options when more details on the stereo/SWB support are available.

Encoding considerations: This media type is framed and contains binary data; see Section 4.8 of [\[RFC4288\] \(Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures," December 2005.\)](#).

Security considerations: See [Section 7 \(Security Considerations\)](#) of RFC XXXX. [RFC Editor: Upon publication as an RFC, please "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: None.

Published specification: RFC XXX. [RFC Editor: Upon publication as an RFC, please "XXXX" with the number assigned to this document and remove this note.]

Applications which use this media type: For example: Voice over IP, audio and video conferencing, audio streaming and voice messaging.

Additional information: None.

Person & email address to contact for further information: Ari Lakaniemi, ari.lakaniemi@nokia.com

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP [\[RFC3550\] \(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\)](#).

Author: Ari Lakaniemi, ari.lakaniemi@nokia.com

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

5.2. Mapping to SDP Parameters

[TOC](#)

The information carried in the media type specification has a specific mapping to fields of the SDP [\[RFC4566\] \(Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol," July 2006.\)](#), which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the G.718 codec, the mapping is as follows:

*The media type ("audio") goes in SDP "m=" as the media name.

*The media subtype ("G718") goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be 32000 for G.718.

NOTE: The current choice for the RTP clock rate is a 'placeholder'. The clock rate needs to be set according to SWB sampling rate, which is still T.B.D. Since the core codec employs 16000 Hz sampling rate, an integer multiple of 16000 Hz seems to be a preferable choice.

*The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

*Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type string as a semicolon separated list of parameter=value pairs.

5.3. Offer/Answer Considerations

[TOC](#)

The following considerations apply when using the SDP offer/answer [\[RFC3264\] \(Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol \(SDP\)," June 2002.\)](#) mechanism to negotiate the G.718 transport. The parameter "layers" MAY be used to indicate the layer configuration for the each RTP session belonging to current G.718 session an end-point making the offer is ready to transmit and wishes to receive.

*In case the G.718 session consists of a single RTP session, it is RECOMMENDED not to impose any layer restrictions for the session but to use the rate control functionality to set possible restrictions on usage of the higher or highest layers. If the offer includes a layer configuration parameter, the answer MAY use different configuration, but the highest layer in the answer MUST NOT be higher than the highest layer of the offered configuration.

NOTE:

Support for answer modifying the layer configuration is FFS.

*In case the G.718 session consists of multiple RTP sessions, the answer MUST use the layer configurations provided in the offer for the sessions it accepts.

5.4. Declarative Usage of SDP

[TOC](#)

In declarative usage, such as SDP in RTSP [[RFC2326](#)] ([Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol \(RTSP\)," April 1998.](#)) or SAP [[RFC2974](#)] ([Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol," October 2000.](#)), the parameter "layers" SHALL be interpreted to provide a set of layers that the sender MAY use in the session.

5.5. SDP Examples

[TOC](#)

Some example SDP session descriptions utilizing G.718 encodings are provided below.

5.5.1. Example 1

[TOC](#)

The first example illustrates the simple case with the G.718 session employing a single RTP session and the AVPF profile is offered, and the answer accepts the offer without any changes.

Offer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
```

Answer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
```

[TOC](#)

5.5.2. Example 2

This example shows a bit more complex case where the G.718 session using a single RTP session and the AVPF profile is offered with the restriction to send/receive only with layers L1 and L2. The answer indicates that the other end-point is happy to receive (and send) layers up to L5.

Offer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
a=fmtp:97 layers=1,2
```

Answer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
a=fmtp:97 layers=1,2,3,4,5
```

5.5.3. Example 3

[TOC](#)

The third example shows an G.718 session using multiple RTP sessions with the AVPF profile. The answerer wishes to use only layers up to L3.

Offer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
a=fmtp:97 layers=1,2
a=mid=1
```

```
m=audio 49122 RTP/AVPF 98
a=rtpmap:98 G718/32000/1
a=fmtp:98 layers=3
a=mid=2
a=depend:lay 1
```

```
m=audio 49124 RTP/AVPF 99
a=rtpmap:99 G718/32000/1
a=fmtp:99 layers=4,5
a=mid=3
a=depend:lay 1 2
```

Answer:

```
m=audio 49120 RTP/AVPF 97
a=rtpmap:97 G718/32000/1
a=fmtp:97 layers=1,2
a=mid=1
```

```
m=audio 49120 RTP/AVPF 98
a=rtpmap:98 G718/32000/1
```

```
a=fmtp:98 layers=3
a=mid=2
a=depend:lay 1
```

Note that the dependency signaling described in [\[RFC5583\] \(Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol \(SDP\)," July 2009.\)](#) is used in the third example above to indicate the relationship between the layers distributed into separate RTP sessions.

6. Congestion Control

[TOC](#)

As a scalable codec, G.718 implicitly provides means for congestion control by providing a possibility for 'thinning' the bitstream. The RTP payload format according to this specification provides several different means for reducing the G.718 session bandwidth. The most appropriate mechanism (in terms of impact to the user experience) depends on the employed payload structure and also on the employed session configuration (single RTP session or multiple RTP sessions). The following means (in no particular order) can be used to assist congestion control procedures -- either by the sender or by the intermediate node.

*The payloads carrying the EDUs representing the highest layers in an G.718 session can be dropped, along with all associated transport blocks

*The transport blocks carrying the EDUs representing the highest layers within the payload can be dropped

*Transport blocks or payloads carrying EDUs belonging to redundant frames included in the payload can be dropped

7. Security Considerations

[TOC](#)

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [\[RFC3550\] \(Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.\)](#), and in any appropriate RTP profile (for example [\[RFC3551\] \(Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," July 2003.\)](#) or [\[RFC4585\] \(Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol \(RTCP\)-Based Feedback \(RTP/AVPF\)," July 2006.\)](#)). This implies that

confidentiality of the media streams is achieved by encryption; for example, through the application of SRTP [\[RFC3711\]](#) ([Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol \(SRTP\)," March 2004.](#)). Because the data compression used with this payload format is applied end-to-end, any encryption needs to be performed after compression.

A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream that will increase the processing load of the decoder and may cause the receiver to be overloaded. For example inserting additional EDUs representing the higher enhancement layers on top of the ones actually transmitted may increase the decoder load. However, the G.718 codec is not particularly vulnerable to such an attack, since the majority of the computational load in an G.718 session is associated to the encoder. Another form of possible attack might be forging of codec bit-rate control messages, which may result in encoder operating employing higher number of enhancement layers than originally intended and thereby requiring larger amount of computation resources. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED; for example, with SRTP [\[RFC3711\]](#) ([Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol \(SRTP\)," March 2004.](#)).

Note that the appropriate mechanism to ensure confidentiality and integrity of RTP packets and their payloads is very dependent on the application and on the transport and signaling protocols employed. Thus, although SRTP is given as an example above, other possible choices exist.

Note that end-to-end security with either authentication, integrity or confidentiality protection will prevent a network element not within the security context from performing media-aware operations other than discarding complete packets. To allow any (media-aware) intermediate network element to perform its operations, it is required to be a trusted entity which is included in the security context establishment.

8. IANA Considerations

[TOC](#)

IANA is kindly requested to register a media type for the G.718 codec for RTP transport, as specified in [Section 5.1 \(Media Type Registration\)](#) of this document.

9. Acknowledgements

[TOC](#)

Thanks to Qin Wu for useful review and commentary.

10. References

[TOC](#)

10.1. Normative References

[TOC](#)

[AMR-WB]	3GPP, " Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description ," 3GPP TS 26.171 5.0.0, April 2001.
[ITU.G718.2008]	International Telecommunications Union, "Frame Error Robust Narrowband and Wideband Embedded Variable Bit-Rate Coding of Speech and Audio from 8-32 Kbit/s," ITU-T Recommendation G.718, May 2008.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT , HTML , XML).
[RFC3264]	Rosenberg, J. and H. Schulzrinne, " An Offer/Answer Model with Session Description Protocol (SDP) ," RFC 3264, June 2002 (TXT).
[RFC3550]	Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, " RTP: A Transport Protocol for Real-Time Applications ," STD 64, RFC 3550, July 2003 (TXT , PS , PDF).
[RFC3551]	Schulzrinne, H. and S. Casner, " RTP Profile for Audio and Video Conferences with Minimal Control ," STD 65, RFC 3551, July 2003 (TXT , PS , PDF).
[RFC3711]	Baughner, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, " The Secure Real-time Transport Protocol (SRTP) ," RFC 3711, March 2004 (TXT).
[RFC4288]	Freed, N. and J. Klensin, " Media Type Specifications and Registration Procedures ," BCP 13, RFC 4288, December 2005 (TXT).
[RFC4566]	Handley, M., Jacobson, V., and C. Perkins, " SDP: Session Description Protocol ," RFC 4566, July 2006 (TXT).
[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 (TXT).
[RFC4855]	Casner, S., " Media Type Registration of RTP Payload Formats ," RFC 4855, February 2007 (TXT).
[RFC4867]	Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, " RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs ," RFC 4867, April 2007 (TXT).
[RFC5583]	Schierl, T. and S. Wenger, " Signaling Media Decoding Dependency in the Session Description Protocol (SDP) ," RFC 5583, July 2009 (TXT).

10.2. Informative References

[TOC](#)

[McCanne]	McCanne, S., Jacobson, V., and M. Vetterli, "Receiver-driven layered multicast," ACM SIGCOMM Computer Communication Review Volume 26 Issue 4, October 1996.
[RFC2326]	Schulzrinne, H., Rao, A., and R. Lanphier , "Real Time Streaming Protocol (RTSP)," RFC 2326, April 1998 (TXT).
[RFC2974]	Handley, M., Perkins, C., and E. Whelan, " Session Announcement Protocol ," RFC 2974, October 2000 (TXT).
[RFC3828]	Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, " The Lightweight User Datagram Protocol (UDP-Lite) ," RFC 3828, July 2004 (TXT).
[RFC4340]	Kohler, E., Handley, M., and S. Floyd, " Datagram Congestion Control Protocol (DCCP) ," RFC 4340, March 2006 (TXT).
[RFC5117]	Westerlund, M. and S. Wenger, " RTP Topologies ," RFC 5117, January 2008 (TXT).
[RFC6051]	Perkins, C. and T. Schierl, " Rapid Synchronisation of RTP Flows ," RFC 6051, November 2010 (TXT).

Appendix A. Payload Examples

[TOC](#)

The G.718 payload structure enables flexible transport either by carrying all layers in the same payload or separating the layers into separate payloads. The following subsections illustrate different possibilities for transport by simple examples. Note that examples do not show the full payload structure to keep the illustration simple.

A.1. Simple Payload Examples

[TOC](#)

A.1.1. All The Layers in The Same Payload

[TOC](#)

The illustration below shows layers L1-L3 from two encoded frames encapsulated into separate payloads using single transport block.

```
+-----+-----+-----+-----+-----+-----+
| RTP1  | L-ID=3 |NF=0 |F1-L1 |F1-L2 |F1-L3 |
+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+
| RTP2  | L-ID=3 |NF=0 |F2-L1 |F2-L2 |F2-L3 |
+-----+-----+-----+-----+-----+-----+
```

In the case where the same layers from two input frames are encapsulated into one payload using single transport block, the structure is as shown below.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP1  | L-ID=3 |NF=1 |F1-L1 |F2-L1 |F1-L2 |F2-L2 |F3-L3 |F2-L3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

The third example illustrates the case where the layers L1-L3 from two input frames are encapsulated into one payload using two separate transport blocks, the first one carrying L1 and the other one containing L2 and L3.

```
+-----+-----+-----+-----+-----+
| RTP1  | L-ID=1 |NF=1 |F1-L1 |F2-L1 |
+-----+-----+-----+-----+-----+
| L-ID=7 |NF=1 |F1-L2 |F2-L2 |F2-L2 |F2-L3 |
+-----+-----+-----+-----+-----+
```

A.1.2. Layers in Seperate RTP Streams

[TOC](#)

In this case the data for each layer is transmitted in its own payload.

In the first example each transport block including a single EDU is carried in its own RTP payload.

```
+-----+-----+-----+-----+         +-----+-----+-----+-----+
| RTP1a | L-ID=1 |NF=0 |F1-L1|         | RTP1b | L-ID=6 |NF=0 |F1-L2|
+-----+-----+-----+-----+         +-----+-----+-----+-----+

+-----+-----+-----+-----+         +-----+-----+-----+-----+
| RTP1c |L-ID=10 |NF=0 |F1-L3|         | RTP2a | L-ID=1 |NF=0 |F2-L1|
+-----+-----+-----+-----+         +-----+-----+-----+-----+

+-----+-----+-----+-----+         +-----+-----+-----+-----+
| RTP2b | L-ID=6 |NF=0 |F2-L2|         | RTP2c |L-ID=10 |NF=0 |F2-L3|
+-----+-----+-----+-----+         +-----+-----+-----+-----+
```

If the payloads carry data from two consecutive input frames, the same encoded data as in the previous example is arranged as follows.


```

+-----+-----+-----+-----+-----+
| RTP1a | L-ID=1 |NF=1 |F1-L1|F2-L1|
+-----+-----+-----+-----+

+-----+-----+-----+-----+
| RTP1b | L-ID=6 |NF=1 |F1-L2|F2-L2|
+-----+-----+-----+-----+

+-----+-----+-----+-----+
| RTP1c |L-ID=10 |NF=1 |F1-L3|F2-L3|
+-----+-----+-----+-----+

```

A.2. Advanced Examples

[TOC](#)

A.2.1. Different Update Rate for Subset of Layers

[TOC](#)

An example employing different update rates (i.e. different number of frames per packet) for selected subsets of layers. In these examples all core codec layers L1-L5 are shown.

```

+-----+-----+-----+-----+-----+-----+-----+
| RTP1  | L-ID=1 |NF=3 |F1-L1|F2-L1|F3-L1|F4-L1|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP2a | L-ID=7 |NF=1 |F1-L2|F2-L2|F1-L3|F2-L3|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP3a |L-ID=14 |NF=0 |F1-L4|F1-L5|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP3b |L-ID=14 |NF=0 |F2-L4|F2-L5|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP2b | L-ID=7 |NF=1 |F3-L2|F4-L2|F3-L3|F4-L3|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP3c |L-ID=14 |NF=0 |F3-L4|F3-L5|
+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+
| RTP3d |L-ID=14 |NF=0 |F4-L4|F4-L5|
+-----+-----+-----+-----+-----+-----+-----+

```

A.2.2. Redundant Frames With Limited Set of Layers

[TOC](#)

An example transmitting layers L1-L3 as primary data and L1 (of the previous frame) as redundant data is shown below. Each payload carries one primary (i.e. new) frame in one transport block and one redundant frame, which in this example is the frame preceding the primary frame, in another transport block.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP1  | L-ID=1 |NF=0 |F0-L1| L-ID=3 |NF=0 |F1-L1|F1-L2|F1-L3|
+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP2  | L-ID=1 |NF=0 |F1-L1| L-ID=3 |NF=0 |F2-L1|F2-L2|F2-L3|
+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP3  | L-ID=1 |NF=0 |F2-L1| L-ID=3 |NF=0 |F3-L1|F3-L2|F3-L3|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Alternatively, the payload carrying also redundant data for a subset of layers can be arranged differently, as shown in the example below.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP1  | L-ID=3 |NF=0 |F0-L1|F0-L2|F0-L3| L-ID=1 |NF=0 |F1-L1|
+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP2  | L-ID=3 |NF=0 |F1-L1|F1-L2|F1-L3| L-ID=1 |NF=0 |F2-L1|
+-----+-----+-----+-----+-----+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+-----+-----+
| RTP3  | L-ID=3 |NF=0 |F2-L1|F2-L2|F2-L3| L-ID=1 |NF=0 |F3-L1|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Now the first transport block carries the primary data and the second transport block carries the redundant data, which in this case covers the frame following the primary frame. The benefit of this approach is that the redundant data is included in the last (secondary) transport block of the payload, which might be beneficial for possible payload scaling operation within the network.

Authors' Addresses

[TOC](#)

Glen Zorn (editor)

	Network Zen
	227/358 Thanon Sanphawut
	Bang Na, Bangkok 10260
	Thailand
Phone:	+66 (0) 87-040-4617
EEmail:	gwz@net-zen.net
	Ye-Kui Wang
	Huawei Technologies
	400 Somerset Corp Blvd.
	Suite 402
	Bridgewater, NJ 08807
	USA
Phone:	+1 (908) 541-3518
EEmail:	yekuiwang@huawei.com
	Ari Lakaniemi
	Nokia
	P.O.Box 407
	FIN-00045 Nokia Group
	Finland
Phone:	+358-71-8008000
EEmail:	ari.lakaniemi@nokia.com