Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-rmt-pi-alc-03.ps

RMT WG
M.Luby/Digital Fountain
J.Gemmell/Microsoft
L.Vicisano/Cisco
L.Rizzo/ACIRI and Univ. Pisa
J. Crowcroft/UCL

18 October 2001
Expires: April 2002

Asynchronous Layered Coding: A massively scalable reliable content delivery protocol instantiation

Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are 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 a "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

This document is a product of the IETF RMT WG. Comments should be addressed to the authors, or the WG's mailing list at rmt@lbl.gov.

Abstract

This document describes the Asynchronous Layered Coding protocol, a massively scalable reliable content delivery protocol, hereafter referred to as ALC. ALC can be used for multi-rate reliable delivery of content to large sets of receivers. This specification of ALC uses version 0 of the LCT building block specified in [4] augmented with a multi-rate congestion control building block that is compliant with RFC2357 such as one of the congestion control building blocks described [6] and [1]. ALC achieves reliability based on FEC codecs as generally described in [2], and in particular uses the FEC building block described in [3].

Table of Contents

1. Introduction 3
1.1. Related Documents 4
1.2. Environmental Requirements and Considerations 4
2. General Architecture 5
2.1. Delivery service models 5
2.2. Congestion Control 6
3. Packet format used by the ALC protocol 6
3.1. Specific packet format 6
3.2. Packet header field definitions 7
4. Procedures 8
4.1. Sender Operation 8
4.2. Receiver Operation 8
5. Security Considerations 8
6. IANA Considerations 9
7. Intellectual Property Issues 9
8. Acknowledgments 9
9. References 9
10. Authors' Addresses 10
11. Full Copyright Statement 11

1. Introduction

This document describes a massively scalable reliable content delivery protocol, Asynchronous Layered Coding (ALC), for multi-rate reliable content delivery. ALC is a protocol instantiation as defined in RFC3048. This specification of ALC uses version 0 of the LCT building block specified in [4]. Like the LCT building block, the ALC protocol is primarily designed to be used with the IP multicast network service, but may also be used with other basic underlying network services such as unicast IP.

All ALC packets in a session use the default LCT header to convey, among others, session and layering information. ALC uses FEC codes to provide reliability as generally described in [2], and in particular ALC uses the FEC building block described in [3]. Thus, all packets in a session contain an FEC payload ID in a format that is compliant with the FEC building block. Overall, a packet includes the default LCT header, followed by the FEC Payload ID, followed by the packet payload.

Although ALC is a transport protocol, no IP protocol identifier is reserved for it and this specification defines ALC as UDP payload, in accordance with the RMT-WG guidelines. The ALC header hence follows an UDP header in an IP packet. Future versions of this specification, or companion documents may extend ALC to use the IP network layer service directly. Note that this specification does not make any assumption on the presence of UDP.

A crucial point is that ALC strongly relies on FEC codecs in the sense that there is no request for retransmission of individual packets from receivers that miss packets in order to assure reliable reception of an object, and the packets and their rate of transmission out of the sender can be independent of the number and the individual reception experiences of the receivers. In some delivery service models it is appropriate that receivers send out of band messages to the sender to provide guaranteed delivery of content. For example, in a push reliable delivery model receivers may send messages to a sender to continue the session if receivers have not yet received enough packets to recover the content or to terminate the session when receivers have received enough packets to recover the content. To be able to track usage of the system, receivers may also send messages out of band to the sender that contain statistics on their reception experience. ALC, however, does not require nor specify such messages, with the aim of avoiding unnecessary limitation to the scalability of the basic ALC protocol.

The LCT building block provides support for congestion control, and the ALC protocol uses this support. In particular, to take full advantage of the scalability features of ALC, the congestion control building block used by ALC must be a multi-rate congestion control scheme that requires no receiver feedback to the sender and must use the Congestion Control Information field within the LCT header to convey packet by packet congestion control information to receivers. Possible congestion control building blocks that could be used with ALC are described in [6] and [1]. A sender sends packets in the session to several LCT channels at potentially different rates. Then, individual receivers can adjust their reception rate within a session by adjusting which set of LCT channels they are joined to at each point in time depending on the available bandwidth between the receiver and the sender, but independent of other receivers.

Receivers may join and leave a session asynchronously at their discretion.

Note also that the congestion control building block used by ALC must conform with RFC2357.

ALC has the following properties when multicast is used to deliver packets:

- o To each receiver, it appears as if though there is a dedicated session from the sender to the receiver, where the reception rate adjusts to congestion along the path from sender to receiver.
- o To the sender, there is no difference in load or outgoing rate if one receiver is joined to the session or a million (or any number of) receivers are joined to the session, independent of when the receivers join and leave.
- o On each link in the network, the packet traffic from the session and its reaction to competing traffic is the same whether there is one receiver or a million receivers beyond the link.

Thus, ALC provides a massively scalable content delivery system that is network friendly.

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.

1.1. Related Documents

This specification of ALC must use version 0 of the LCT building block as specified in [4]. A more in-depth description of the use of FEC codecs in reliable content delivery protocols is given in [2]. All packets in a session must contain an FEC Payload ID in a format that is compliant with the FEC building block described in [3]. Implementors of ALC must also implement a multi-rate feedback-free congestion control building block that is in accordance to RFC2357, where the congestion control is over the entire session. Some possible schemes are specified in [6] and [1]. Congestion control must be integrated into ALC through the support provided in the LCT building block.

It is recommended that ALC implementors use some packet authentication scheme to protect the protocol from attacks. An example of a possibly suitable scheme is described in [5]. Packet authentication in ALC, if used, is to be integrated through the support provided in the LCT building block.

1.2. Environmental Requirements and Considerations

ALC is intended for mult-rate congestion controlled delivery of objects, i.e., reliable content delivery.

This specification defines ALC as payload of the UDP transport protocol (RFC768).

All of the environmental requirements and considerations that apply to the LCT building block and to the FEC building block also apply to ALC.

In particular, LCT requires the ability by receivers to uniquely identify and demultiplex packets associated with an LCT session, and ALC inherits this requirement. To this purpose, a Transport Session Identifier (TSI) must be associated with each session. The TSI is scoped by the IP address of the sender, and the IP address of the sender together with the TSI must uniquely identify the session. It is recommended that the TSI field in the LCT header is used to convey the ALC TSI. If the TSI is not present in the LCT header, then 16-bit UDP source port is used as TSI.

If Generic Router Assist (GRA) is being used then additional dependencies may be introduced by GRA on the TSI field. Generic Router Assist is a current topic of discussion within the RMT

working group. If a TSI value appears more than once in a packet then all occurrences of the value must be equal.

If, in future specification, there is no underlying TSI provided by the network, transport or any other layer, then the use of the TSI in the LCT header must be mandatory.

2. General Architecture

ALC protocol uses the FEC codecs in the form described in [3] combined with the LCT building block as described in [4]. Thus, the terminology used in the description of the LCT building block and the FEC building block are inherited by the ALC protocol and this terminology is used below. In particular, the definition of a session for the ALC protocol is the same as it is for the LCT building block.

There is a one-to-one mapping between ALC sessions and LCT sessions.

An ALC packet consists of the default LCT header, followed by the FEC Payload ID, followed by the packet payload. The packet payload is as described in [3]. If Generic Router Assist (GRA) is being used, then additional constraints may be introduced on the LCT header.

The ALC header immediately follows a UDP header.

The out of band information used by the ALC protocol consists of all the out of band information required for both the LCT building block and for the FEC building block. The possible means for acquiring this out of band information is the same as for the LCT building block and the FEC building block, and in particular is outside the scope of the ALC protocol.

2.1. Delivery service models

ALC can support several different delivery service models. Two examples are briefly described here.

Push service model.

One way a push service model can be used for reliable content delivery is to deliver a series of objects. For example, a receiver could join the session and dynamically adapt the number of LCT channels the receiver is joined to until enough packets have been received to reconstruct an object. After reconstructing the object the receiver may stay in the session and wait for the transmission of the next object.

The push model is particularly attractive in satellite networks and wireless networks. In these cases, a session may consist of one fixed rate LCT channel.

On-demand content delivery model.

For an on-demand content delivery service model, senders typically transmit for some given time period selected to be long enough to allow all the intended receivers to join the session and recover a single object. For example a popular software update might be transmitted using ALC for several days, even though a receiver may be able to complete the download in one hour total

of connection time, perhaps spread over several intervals of time.

In this case the receivers join the session at any point in time when it is active and dynamically adapt the number of LCT channels they subscribe to according to the available bandwidth using a multi-rate congestion control building block. Receivers leave the session when they have received enough packets to recover the object.

Other service models.

There may be other delivery service models that can be supported by ALC. The description of the potential applications, the appropriate delivery service model, and the additional mechanisms to support such functionalities when combined with ALC is beyond the scope of this document.

2.2. Congestion Control

A multi-rate congestion control building block that is feedback-free, that is suitable for reliable content delivery, and that conforms to RFC2357 is to be used by ALC. An example of such a congestion control building block is described in [6] and [1]. While the general behavior of the congestion control building block is to reduce transmission rate in the presence of congestion and gradually increase the rate in the absence of congestion, the actual dynamic behavior (e.g. response to single losses) can vary.

The congestion control building block must use the Congestion Control Information field carried in the LCT header of each packet. Congestion control must be applied to all packets within a session independently of which information about which object is carried in each packet.

3. Packet format used by the ALC protocol

The packet format used by the ALC protocol is the UDP header followed by the default LCT header followed by the FEC Payload ID followed by the packet payload. The packet payload is as described in [3], i.e., it contains encoding information about the object. If Generic Router Assist (GRA) is being used, then additional information may be required in the packet headers, and an additional specification may be needed.

The version number of ALC specified in this document is 0. This coincides with version 0 of the LCT building block [4] used in this specification. The LCT version number field should be interpreted as the ALC version number field.

Some instances of sessions may require the generation of feedback from the receivers to the sender. However, the mechanism for doing this is outside the scope of ALC.

3.1. Specific packet format

The packet format used for ALC protocol is depicted in Figure 1.

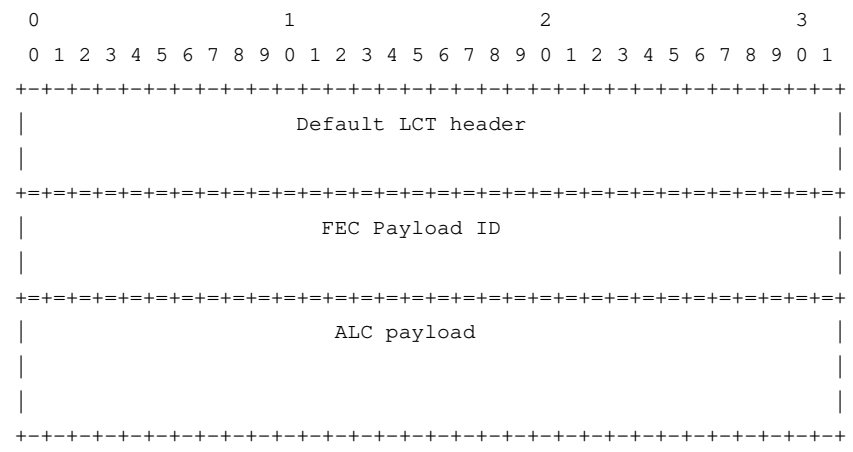

Figure 1 - Packet format for the ALC protocol

The length of the Default LCT header is explicitly encoded in the header itself (see [4]) , while the length of the FEC Payload ID is implicitly defined by the FEC Encoding ID (see [3]) , that is communicated out of band and, possibly, through the Codepoint field in the LCT header. The rest of the datagram is made of ALC payload.

In some special cases an ALC source may need to produce ALC packets that do not contain any payload. This may be required, for example, to signal the end of a session or to convey congestion control information. These data-less packets do not contain the FEC Payload ID either, but only the LCT header fields. The total datagram length, conveyed by outer protocol headers (e.g. the IP or UDP header), enables receivers to detect the absence of the ALC payload and FEC Payload ID.

3.2. Packet header field definitions

The description of the fields and their functions within the default LCT header can be found in [4]. The information that needs to be communicated out of band for the LCT building block and the possible mechanisms for achieving this are described in [4].

The description of the fields and their functions within the FEC Payload ID can be found in [3]. The information that needs to be communicated out of band for the FEC codecs and the possible mechanisms for achieving this are described in [3]. The mechanisms for communicating the out of band information needed for the LCT building block, including the mapping between the Codepoint field values in the default LCT header and the interpretations of the values, may be the same as those used for communicating the out of band information needed for the FEC building block, with the exception that portions of the information needed for FEC building blocks may be communicated through the use of the Codepoint field in the default LCT header.

Multiple objects may be transmitted in the same session. The Transport Object Identifier (TOI)

field in the LCT header must be used to identify the different objects, and the FEC Payload ID and the packet payload must correspond to the object identified by the TOI in the LCT header.

If the FEC encoding scheme varies among different objects transmitted in the same session, the Codepoint field in the LCT header must be used to convey, in each packet, the association between the ALC payload and the tuple <FEC Encoding ID, FEC Encoding Name>. This allows the receiver to parse the FEC Payload ID field and to select the appropriate decoder class without having to interpret the TOI field.

4. Procedures

4.1. Sender Operation

The sender operation when using the ALC protocol includes all the points made about the sender operation when using the LCT building block as described in [4], and the FEC building block as described in [3]. The following description highlights some of the additional considerations to be taken into account with respect to the ALC protocol.

A sender using the ALC protocol must make available all applicable information regarding the session. This information includes:

- Any information needed by the LCT building block;
- The congestion control building block being used within the LCT building block;
- The mapping between values of the Codepoint field in the default LCT header and interpretations of these values;
- Any information needed for the FEC building block, including the FEC Object Transmission Information;
- If used, the packet authentication scheme being used within the LCT building block, and all relevant information which is necessary for packet authentication purposes.

4.2. Receiver Operation

The receiver operation when using the ALC protocol includes all the points made about the receiver operation that pertain to reliable content delivery when using the LCT building block as described in [4], and that pertain to using the FEC building block as described in [3]. The following description highlights some of the additional considerations to be taken into account with respect to the ALC protocol.

When a receiver receives a packet, the receiver must process the default LCT header (excluding the Codepoint field) as described in [4] before processing any other fields of the packet.

5. Security Considerations

The same security consideration that apply to the LCT building block and to the FEC building block also apply to the ALC protocol. In addition, any security considerations that apply to the congestion control building block used by the ALC protocol within the LCT building block also apply.

6. IANA Considerations

No information in this specification is directly subject to IANA registration. However, building blocks components used by the ALC protocol may introduce additional IANA considerations. In particular, the FEC building block used by the ALC protocol does require IANA registration of the FEC codecs used.

7. Intellectual Property Issues

No specific reliability building block or congestion control building block is specified or referenced as mandatory in this document.

ALC may be used with congestion control building blocks and other building blocks which contain proprietary elements, or have pending or granted patents.

8. Acknowledgments

Thanks to Vincent Roca and Justin Chapweske for their detailed comments on this draft.

9. References

[1] Byers, J.W., Frumin, M., Horn, G., Luby, M., Mitzenmacher, M., Roetter, A., and Shaver, W., "FLID-DL: Congestion Control for Layered Multicast", Proceedings of Second International Workshop on Networked Group Communications (NGC 2000), Palo Alto, CA, November 2000.

[2] Luby, M., Gemmell, Vicisano, L., J., Rizzo, L., Handley, M., Crowcroft, J., "The use of Forward Error Correction in Reliable Multicast", Internet Draft draft-ietf-rmt-info-fec-01.txt, October 2001, a work in progress.

[3] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M., Crowcroft, J., "RMT BB Forward Error Correction Codes", Internet Draft draft-ietf-rmt-bb-fec-04.txt, October 2001.

[4] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M., Crowcroft, J., "Layered Coding Transport: A massively scalable content delivery transport", Internet Draft draft-ietf-rmt-bb-lct-02.txt, October 2001.

[5] Perrig, A., Canetti, R., Song, D., Tygar, J.D., "Efficient and Secure Source Authentication for Multicast", Network and Distributed System Security Symposium, NDSS 2001, pp. 35-46, February 2001.

[6] Vicisano, L., Rizzo, L., Crowcroft, J., "TCP-like Congestion Control for Layered Multicast Data Transfer", IEEE Infocom '98, San Francisco, CA, Mar.28-Apr.1 1998.

10. Authors' Addresses

Michael Luby
luby@digitalfountain.com
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA, USA, 94538

Jim Gemmell
jgemmell@microsoft.com
Microsoft Research
301 Howard St., #830
San Francisco, CA, USA, 94105

Lorenzo Vicisano
lorenzo@cisco.com
cisco Systems, Inc.
170 West Tasman Dr.,
San Jose, CA, USA, 95134

Luigi Rizzo
luigi@iet.unipi.it
Dip. Ing. dell'Informazione,
Univ. di Pisa
via Diotisalvi 2, 56126 Pisa, Italy

Jon Crowcroft
J.Crowcroft@cs.ucl.ac.uk
Department of Computer Science
University College London
Gower Street,
London WC1E 6BT, UK

11. Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."