Internet Engineering Task Force (IETF)
Request for Comments: 5775
Obsoletes: 3450
Category: Standards Track
ISSN: 2070-1721

M. Luby
M. Watson
L. Vicisano
Qualcomm, Inc.

April 2010

Asynchronous Layered Coding (ALC) Protocol Instantiation

Abstract

This document describes the Asynchronous Layered Coding (ALC) protocol, a massively scalable reliable content delivery protocol. Asynchronous Layered Coding combines the Layered Coding Transport (LCT) building block, a multiple rate congestion control building block and the Forward Error Correction (FEC) building block to provide congestion controlled reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender. This document obsoletes RFC 3450.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5775.

Table of Contents

1. Introduction

This document describes a massively scalable reliable content delivery protocol, Asynchronous Layered Coding (ALC), for multiple rate congestion controlled reliable content delivery. The protocol is specifically designed to provide massive scalability using IP multicast as the underlying network service. Massive scalability in this context means the number of concurrent receivers for an object is potentially in the millions, the aggregate size of objects to be delivered in a session ranges from hundreds of kilobytes to hundreds of gigabytes, each receiver can initiate reception of an object asynchronously, the reception rate of each receiver in the session is the maximum fair bandwidth available between that receiver and the sender, and all of this can be supported using a single sender.

Because ALC is focused on reliable content delivery, the goal is to deliver objects as quickly as possible to each receiver while at the same time remaining network friendly to competing traffic. Thus, the congestion control used in conjunction with ALC should strive to maximize use of available bandwidth between receivers and the sender while at the same time backing off aggressively in the face of competing traffic.

The sender side of ALC consists of generating packets based on objects to be delivered within the session and sending the appropriately formatted packets at the appropriate rates to the channels associated with the session. The receiver side of ALC consists of joining appropriate channels associated with the session, performing congestion control by adjusting the set of joined channels associated with the session in response to detected congestion, and using the packets to reliably reconstruct objects. All information flow in an ALC session is in the form of data packets sent by a single sender to channels that receivers join to receive data.

ALC does specify the Session Description needed by receivers before they join a session, but the mechanisms by which receivers obtain this required information is outside the scope of ALC. An application that uses ALC may require that receivers report statistics on their reception experience back to the sender, but the mechanisms by which receivers report back statistics is outside the scope of ALC. In general, ALC is designed to be a minimal protocol instantiation that provides reliable content delivery without unnecessary limitations to the scalability of the basic protocol.

This document is a product of the IETF RMT WG and follows the general guidelines provided in [RFC3269].

A previous version of this document was published in the "Experimental" category as [RFC3450] and is obsoleted by this document.

This Proposed Standard specification is thus based on and backwards compatible with the protocol defined in [RFC3450] updated according to accumulated experience and growing protocol maturity since its original publication. Said experience applies both to this specification itself and to congestion control strategies related to the use of this specification.

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 BCP 14, [RFC2119].

1.1. Delivery Service Models

ALC can support several different reliable content delivery service models as described in [RFC5651].

1.2. Scalability

Massive scalability is a primary design goal for ALC. IP multicast is inherently massively scalable, but the best effort service that it provides does not provide session management functionality, congestion control, or reliability. ALC provides all of this on top of IP multicast without sacrificing any of the inherent scalability of IP multicast. ALC has the following properties:

- To each receiver, it appears as if 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.
- To the sender, there is no difference in load or outgoing rate if one receiver or a million (or any number of) receivers are joined to the session, independent of when the receivers join and leave.
- No feedback packets are required from receivers to the sender.
- Almost all packets in the session that pass through a bottleneck link are utilized by downstream receivers, and the session shares the link with competing flows fairly in proportion to their utility.

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

ALC intentionally omits any application-specific features that could potentially limit its scalability. By doing so, ALC provides a minimal protocol that is massively scalable. Applications may be built on top of ALC to provide additional features that may limit the scalability of the application. Such applications are outside the scope of this document.

1.3. Environmental Requirements and Considerations

All of the environmental requirements and considerations that apply to the LCT building block [RFC5651], the FEC building block [RFC5052], the multiple rate congestion control building block, and to any additional building blocks that ALC uses also apply to ALC.

The IP multicast model defined in [RFC1112] is commonly known as Any-Source Multicast (ASM), in contrast to Source-Specific Multicast (SSM) which is defined in [RFC3569]. One issue that is specific to ALC with respect to ASM is the way the multiple rate congestion control building block interacts with ASM. The congestion control building block may use the measured difference in time between when a join to a channel is sent and when the first packet from the channel arrives in determining the receiver reception rate. The congestion control building block may also use packet sequence numbers per channel to measure losses, and this is also used to determine the receiver reception rate. These features raise two concerns with respect to ASM: The time difference between when the join to a channel is sent and when the first packet arrives can be significant due to the use of Rendezvous Points (RPs) and the Multicast Source Discovery Protocol (MSDP) [RFC3618] protocol, and packets can be lost in the switch over from the (*,G) join to the RP and the (S,G) join directly to the sender. Both of these issues could potentially substantially degrade the reception rate of receivers. To ameliorate these concerns, it is recommended during deployment to ensure that the RP be as close to the sender as possible. SSM does not share these same concerns. For a fuller consideration of these issues, consult the multiple rate congestion control building block.

2. Architecture Definition

ALC uses the LCT building block [RFC5651] to provide in-band session management functionality. ALC uses a multiple rate congestion control building block that is compliant with [RFC2357] to provide congestion control that is feedback free. Receivers adjust their reception rates individually by joining and leaving channels associated with the session. ALC uses the FEC building block [RFC5052] to provide reliability. The sender generates encoding symbols based on the object to be delivered using FEC codes and sends them in packets to channels associated with the session. Receivers

simply wait for enough packets to arrive in order to reliably reconstruct the object. Thus, 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.

The definition of a session for ALC is the same as it is for LCT. An ALC session comprises multiple channels originating at a single sender that are used for some period of time to carry packets pertaining to the transmission of one or more objects that can be of interest to receivers. Congestion control is performed over the aggregate of packets sent to channels belonging to a session. The fact that an ALC session is restricted to a single sender does not preclude the possibility of receiving packets for the same objects from multiple senders. However, each sender would be sending packets to a different session to which congestion control is individually applied. Although receiving concurrently from multiple sessions is allowed, how this is done at the application level is outside the scope of this document.

ALC is a protocol instantiation as defined in [RFC3048]. This document describes version 1 of ALC, which MUST use version 1 of LCT described in [RFC5651]. Like LCT, ALC is designed to be used with the IP multicast network service. This specification defines ALC as payload of the UDP transport protocol [RFC0768] that supports the IP multicast delivery of packets.

The ALC packet format is illustrated in Figure 1. An ALC packet header immediately follows the IP/UDP header and consists of the default LCT header that is described in [RFC5651] followed by the FEC Payload ID that is described in [RFC5052]. The Congestion Control Information field within the LCT header carries the required Congestion Control Information that is described in the multiple rate congestion control building block specified that is compliant with [RFC2357]. The packet payload that follows the ALC packet header consists of encoding symbols that are identified by the FEC Payload ID as described in [RFC5052].

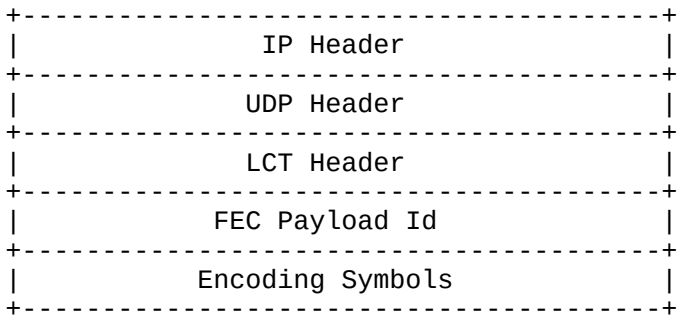

Figure 1: ALC Packet Format

Each receiver is required to obtain a Session Description before joining an ALC session. As described later, the Session Description includes out-of-band information required for the LCT, FEC, and the multiple rate congestion control building blocks. The FEC Object Transmission Information specified in the FEC building block [RFC5052] required for each object to be received by a receiver can be communicated to a receiver either out-of-band or in-band using a Header Extension. The means for communicating the Session Description and the FEC Object Transmission Information to a receiver is outside the scope of this document.

2.1. LCT Building Block

LCT requires receivers to be able to uniquely identify and demultiplex packets associated with an LCT session, and ALC inherits and strengthens this requirement. A Transport Session Identifier (TSI) MUST be associated with each session and MUST be carried in the LCT header of each ALC packet. The TSI is scoped by the sender IP address, and the (sender IP address, TSI) pair MUST uniquely identify the session.

The LCT header contains a Congestion Control Information (CCI) field that MUST be used to carry the Congestion Control Information from the specified multiple rate congestion control protocol. There is a field in the LCT header that specifies the length of the CCI field, and the multiple rate congestion control building block MUST uniquely identify a format of the CCI field that corresponds to this length.

The LCT header contains a Codepoint field that MAY be used to communicate to a receiver the settings for information that may vary during a session. If used, the mapping between settings and Codepoint values is to be communicated in the Session Description, and this mapping is outside the scope of this document. For example, the FEC Encoding ID that is part of the FEC Object Transmission

Information, as specified in the FEC building block [RFC5052], could vary for each object carried in the session, and the Codepoint value could be used to communicate the FEC Encoding ID to be used for each object. The mapping between FEC Encoding IDs and Codepoints could be, for example, the identity mapping.

If more than one object is to be carried within a session, then the Transmission Object Identifier (TOI) MUST be used in the LCT header to identify which packets are to be associated with which objects. In this case, the receiver MUST use the TOI to associate received packets with objects. The TOI is scoped by the IP address of the sender and the TSI, i.e., the TOI is scoped by the session. The TOI for each object is REQUIRED to be unique within a session, but is not required be unique across sessions. Furthermore, the same object MAY have a different TOI in different sessions. The mapping between TOIs and objects carried in a session is outside the scope of this document.

If only one object is carried within a session, then the TOI MAY be omitted from the LCT header.

The LCT header from version 1 of the LCT building block [RFC5651] MUST be used.

The LCT Header includes a two-bit Protocol Specific Indication (PSI) field in bits 6 and 7 of the first word of the LCT header. These two bits are used by ALC as follows:

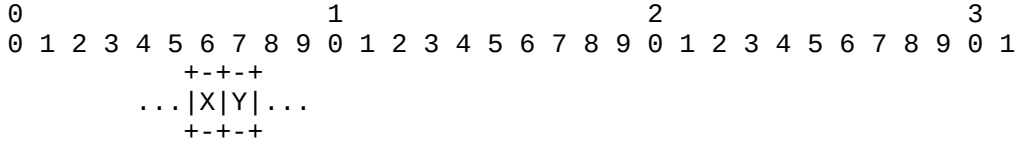

Figure 2: PSI Bits within LCT Header

PSI bit X - Source Packet Indicator (SPI)

PSI bit Y - Reserved

The Source Packet Indicator is used with systematic FEC Schemes which define a different FEC Payload ID format for packets containing only source data compared to the FEC Payload ID format for packets containing repair data. For such FEC Schemes, the SPI MUST be set to 1 when the FEC Payload ID format for packets containing only source data is used, and the SPI MUST be set to zero when the FEC Payload ID for packets containing repair data is used. In the case of FEC

Schemes that define only a single FEC Payload ID format, the SPI MUST be set to zero by the sender and MUST be ignored by the receiver.

Support of two FEC Payload ID formats allows FEC Payload ID information that is only of relevance when FEC decoding is to be performed to be provided in the FEC Payload ID format for packets containing repair data. This information need not be processed by receivers that do not perform FEC decoding (either because no FEC decoding is required or because the receiver does not support FEC decoding).

2.2. Multiple Rate Congestion Control Building Block

At a minimum, implementations of ALC MUST support [RFC3738]. Note that [RFC3738] has been published in the "Experimental" category and thus has lower maturity level than the present document. Caution should be used since it may be less stable than this document.

Congestion control MUST be applied to all packets within a session independently of which information about which object is carried in each packet. Multiple rate congestion control is specified because of its suitability to scale massively and because of its suitability for reliable content delivery. [RFC3738] specifies in-band Congestion Control Information (CCI) that MUST be carried in the CCI field of the LCT header.

Alternative multiple rate congestion control building blocks MAY be supported, but only a single congestion control building block may be used in a given ALC session. The congestion control building block to be used in an ALC session is specified in the Session Description (see Section 2.4). A multiple rate congestion control building block MAY specify more than one format for the CCI field, but it MUST specify at most one format for each of the possible lengths 32, 64, 96, or 128 bits. The value of C in the LCT header that determines the length of the CCI field MUST correspond to one of the lengths for the CCI defined in the multiple rate congestion control building block; this length MUST be the same for all packets sent to a session, and the CCI format that corresponds to the length as specified in the multiple rate congestion control building block MUST be the format used for the CCI field in the LCT header.

When using a multiple rate congestion control building block, a sender sends packets in the session to several channels at potentially different rates. Then, individual receivers adjust their reception rate within a session by adjusting to which set of channels they are joined at each point in time depending on the available bandwidth between the receiver and the sender, but independent of other receivers.

2.3. FEC Building Block

The FEC building block [RFC5052] provides reliable object delivery within an ALC session. Each object sent in the session is independently encoded using FEC codes as described in [RFC3453], which provide a more in-depth description of the use of FEC codes in reliable content delivery protocols. All packets in an ALC session MUST contain an FEC Payload ID in a format that is compliant with the FEC Scheme in use. The FEC Payload ID uniquely identifies the encoding symbols that constitute the payload of each packet, and the receiver MUST use the FEC Payload ID to determine how the encoding symbols carried in the payload of the packet were generated from the object as described in the FEC building block.

As described in [RFC5052], a receiver is REQUIRED to obtain the FEC Object Transmission Information for each object for which data packets are received from the session. In the context of ALC, the FEC Object Transmission Information includes:

- The FEC Encoding ID.
- If an Under-Specified FEC Encoding ID is used, then the FEC Instance ID associated with the FEC Encoding ID.
- For each object in the session, the transfer length of the object in bytes.

Additional FEC Object Transmission Information may be required depending on the FEC Scheme that is used (identified by the FEC Encoding ID).

Some of the FEC Object Transmission Information MAY be implicit based on the FEC Scheme and/or implementation. As an example, source block lengths may be derived by a fixed algorithm from the object length. As another example, it may be that all source blocks are the same length and this is what is passed out-of-band to the receiver. As another example, it could be that the full-sized source block length is provided, and this is the length used for all but the last source block, which is calculated based on the full source block length and the object length. As another example, it could be that the same FEC Encoding ID and FEC Instance ID are always used for a particular application, and thus the FEC Encoding ID and FEC Instance ID are implicitly defined.

Sometimes the objects that will be sent in a session are completely known before the receiver joins the session, in which case the FEC Object Transmission Information for all objects in the session can be communicated to receivers before they join the session. At other

times, the objects may not know when the session begins, receivers may join a session in progress and may not be interested in some objects for which transmission has finished, or receivers may leave a session before some objects are even available within the session. In these cases, the FEC Object Transmission Information for each object may be dynamically communicated to receivers at or before the time packets for the object are received from the session. This may be accomplished using an out-of-band mechanism, in-band using the Codepoint field or a Header Extension, or any combination of these methods. How the FEC Object Transmission Information is communicated to receivers is outside the scope of this document.

2.4. Session Description

Before a receiver can join an ALC session, the receiver needs to obtain a Session Description that contains the following information:

- The multiple rate congestion control building block to be used for the session;
- The sender IP address;
- The number of channels in the session;
- The address and port number used for each channel in the session;
- The Transport Session ID (TSI) to be used for the session;
- An indication of whether or not the session carries packets for more than one object;
- If Header Extensions are to be used, the format of these Header Extensions.
- Enough information to determine the packet authentication scheme being used, if one is being used.

How the Session Description is communicated to receivers is outside the scope of this document.

The Codepoint field within the LCT portion of the header CAN be used to communicate in-band some of the dynamically changing information within a session. To do this, a mapping between Codepoint values and the different dynamic settings MUST be included within the Session Description, and then settings to be used are communicated via the Codepoint value placed into each packet. For example, it is possible that multiple objects are delivered within the same session and that a different FEC encoding algorithm is used for different types of

objects. Then the Session Description could contain the mapping between Codepoint values and FEC Encoding IDs. As another example, it is possible that a different packet authentication scheme is used for different packets sent to the session. In this case, the mapping between the packet authentication scheme and Codepoint values could be provided in the Session Description. Combinations of settings can be mapped to Codepoint values as well. For example, a particular combination of a FEC Encoding ID and a packet authentication scheme could be associated with a Codepoint value.

The Session Description could also include, but is not limited to:

- The mappings between combinations of settings and Codepoint values;
- The data rates used for each channel;
- The length of the packet payload;
- Any information that is relevant to each object being transported, such as the Object Transmission Information for each object, when the object will be available within the session, and for how long.

The Session Description could be in a form such as the Session Description Protocol (SDP) as defined in [RFC4566], XML metadata as defined in [RFC3023], or HTTP/MIME headers as defined in [RFC2616], etc. It might be carried in a session announcement protocol such as SAP as defined in [RFC2974], obtained using a proprietary session control protocol, located on a web page with scheduling information, or conveyed via email or other out-of-band methods. Discussion of Session Description formats and methods for communication of Session Descriptions to receivers is beyond the scope of this document.

2.5. Packet Authentication Building Block

It is RECOMMENDED that implementors of ALC use some packet authentication scheme to protect the protocol from attacks. Suitable schemes are described in [RFC5776] and [RMT-SIMPLE].

3. Conformance Statement

This Protocol Instantiation document, in conjunction with the LCT building block [RFC5651], the FEC building block [RFC5052], and [RFC3738] completely specifies a working reliable multicast transport protocol that conforms to the requirements described in [RFC2357].

4. Functionality Definition

This section describes the format and functionality of the data packets carried in an ALC session as well as the sender and receiver operations for a session.

4.1. Packet Format Used by ALC

The packet format used by ALC is the UDP header followed by the LCT header followed by the FEC Payload ID followed by the packet payload. The LCT header is defined in the LCT building block [RFC5651] and the FEC Payload ID is described in the FEC building block [RFC5052]. The Congestion Control Information field in the LCT header contains the required Congestion Control Information that is described in the multiple rate congestion control building block used. The packet payload contains encoding symbols generated from an object. If more than one object is carried in the session, then the Transmission Object ID (TOI) within the LCT header MUST be used to identify from which object the encoding symbols are generated. Within the scope of an object, encoding symbols carried in the payload of the packet are identified by the FEC Payload ID as described in the FEC building block.

The version number of ALC specified in this document is 1. The version number field of the LCT header MUST be interpreted as the ALC version number field. This version of ALC implicitly makes use of version 1 of the LCT building block defined in [RFC5651].

The overall ALC packet format is depicted in Figure 3. The packet is an IP packet, either IPv4 or IPv6, and the IP header precedes the UDP header. The ALC packet format has no dependencies on the IP version number.

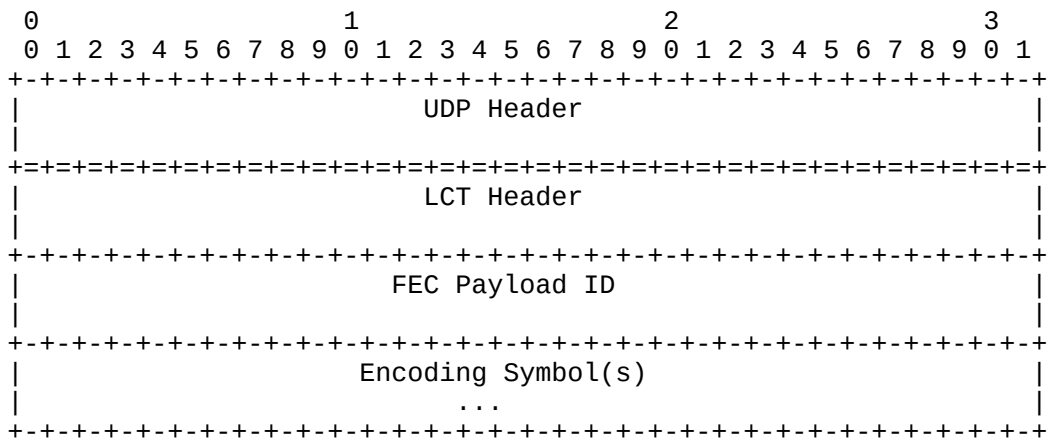

Figure 3: Overall ALC Packet Format

In some special cases an ALC sender 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.

For ALC, the length of the TSI field within the LCT header is REQUIRED to be non-zero. This implies that the sender MUST NOT set both the LCT flags S and H to zero.

4.2. LCT Header Extension Fields

This specification defines a new LCT Header Extension, "EXT_FTI", for the purpose of communicating the FEC Object Transmission Information in association with data packets of an object. The LCT Header Extension Type for EXT_FTI is 64.

The Header Extension Content (HEC) field of the EXT_FTI LCT Header Extension contains the encoded FEC Object Transmission Information as defined in [RFC5052]. The format of the encoded FEC Object Transmission Information is dependent on the FEC Scheme in use and so is outside the scope of this document.

4.3. Sender Operation

The sender operation, when using ALC, includes all the points made about the sender operation when using the LCT building block [RFC5651], the FEC building block [RFC5052], and the multiple rate congestion control building block.

A sender using ALC should make available the required Session Description as described in Section 2.4. A sender should also make available the required FEC Object Transmission Information as described in Section 2.3.

Within a session, a sender transmits a sequence of packets to the channels associated with the session. The ALC sender MUST obey the rules for filling in the CCI field in the packet headers, and it MUST send packets at the appropriate rates to the channels associated with the session as dictated by the multiple rate congestion control building block.

The ALC sender MUST use the same TSI for all packets in the session. Several objects MAY be delivered within the same ALC session. If more than one object is to be delivered within a session, then the sender MUST use the TOI field. Each object MUST be identified by a unique TOI within the session, and the sender MUST use corresponding TOI for all packets pertaining to the same object. The FEC Payload ID MUST correspond to the encoding symbol(s) for the object carried in the payload of the packet.

It is RECOMMENDED that packet authentication be used. If packet authentication is used, then the Header Extensions described in Section 4.2 MAY be used to carry the authentication.

4.4. Receiver Operation

The receiver operation, when using ALC, includes all the points made about the receiver operation when using the LCT building block [RFC5651], the FEC building block [RFC5052], and the multiple rate congestion control building block.

To be able to participate in a session, a receiver needs to obtain the required Session Description as listed in Section 2.4. How receivers obtain a Session Description is outside the scope of this document.

As described in Section 2.3, a receiver needs to obtain the required FEC Object Transmission Information for each object for which the receiver receives and processes packets.

Upon receipt of each packet, the receiver proceeds with the following steps in the order listed.

1. The receiver MUST parse the packet header and verify that it is a valid header. If it is not valid, then the packet MUST be discarded without further processing.

2. The receiver MUST verify that the sender IP address together with the TSI carried in the header matches one of the (sender IP address, TSI) pairs that was received in a Session Description and to which the receiver is currently joined. If there is not a match, then the packet MUST be silently discarded without further processing. The remaining steps are performed within the scope of the (sender IP address, TSI) session of the received packet.

3. The receiver MUST process and act on the CCI field in accordance with the multiple rate congestion control building block.

4. If more than one object is carried in the session, the receiver MUST verify that the TOI carried in the LCT header is valid. If the TOI is not valid, the packet MUST be discarded without further processing.

5. The receiver SHOULD process the remainder of the packet, including interpreting the other header fields appropriately, and using the FEC Payload ID and the encoding symbol(s) in the payload to reconstruct the corresponding object.

It is RECOMMENDED that packet authentication be used. If packet authentication is used, then it is RECOMMENDED that the receiver immediately check the authenticity of a packet before proceeding with step (3) above. If immediate checking is possible and if the packet fails the check, then the receiver MUST silently discard the packet.

5. Security Considerations

The same security considerations that apply to the LCT, FEC, and the multiple rate congestion control building blocks also apply to ALC.

ALC is especially vulnerable to denial-of-service attacks by attackers that try to send forged packets to the session, which would prevent successful reconstruction or cause inaccurate reconstruction of large portions of the object by receivers. ALC is also particularly affected by such an attack because many receivers may receive the same forged packet. There are two ways to protect against such attacks, one at the application level and one at the packet level. It is RECOMMENDED that prevention be provided at both levels.

At the application level, it is RECOMMENDED that an integrity check on the entire received object be done once the object is reconstructed to ensure it is the same as the sent object. Moreover, in order to obtain strong cryptographic integrity protection, a digital signature verifiable by the receiver SHOULD be used to provide this application-level integrity check. However, if even one corrupted or forged packet is used to reconstruct the object, it is likely that the received object will be reconstructed incorrectly. This will appropriately cause the integrity check to fail and in this case, the inaccurately reconstructed object SHOULD be discarded. Thus, the acceptance of a single forged packet can be an effective denial-of-service attack for distributing objects, but an object integrity check at least prevents inadvertent use of inaccurately reconstructed objects. The specification of an application-level integrity check of the received object is outside the scope of this document.

At the packet level, it is RECOMMENDED that a packet-level authentication be used to ensure that each received packet is an authentic and uncorrupted packet containing data for the object arriving from the specified sender. Packet-level authentication has the advantage that corrupt or forged packets can be discarded individually and the received authenticated packets can be used to accurately reconstruct the object. Thus, the effect of a denial-of-service attack that injects forged packets is proportional only to the number of forged packets, and not to the object size. [RMT-SIMPLE]and [RFC5776] described packet level authentication schemes that can be used with the ALC protocol.

In addition to providing protection against reconstruction of inaccurate objects, packet-level authentication can also provide some protection against denial-of-service attacks on the multiple rate congestion control. Attackers can try to inject forged packets with incorrect congestion control information into the multicast stream, thereby potentially adversely affecting network elements and receivers downstream of the attack, and much less significantly the rest of the network and other receivers. Thus, it is also RECOMMENDED that packet-level authentication be used to protect against such attacks. Timed Efficient Stream Loss-Tolerant Authentication (TESLA) [RFC5776] can also be used to some extent to limit the damage caused by such attacks. However, with TESLA, a receiver can only determine if a packet is authentic several seconds after it is received, and thus an attack against the congestion control protocol can be effective for several seconds before the receiver can react to slow down the session reception rate.

Some packet authentication schemes such as TESLA [RFC5776] do not allow an immediate authenticity check. In this case, the receiver

SHOULD check the authenticity of a packet as soon as possible, and if the packet fails the check, then it MUST be silently discarded before Step 5 above. It is RECOMMENDED that if receivers detect many packets that fail authentication checks within a session, the above procedure should be modified for this session so that Step 3 is delayed until after the authentication check and only performed if the check succeeds.

Reverse Path Forwarding checks SHOULD be enabled in all network routers and switches along the path from the sender to receivers to limit the possibility of a bad agent injecting forged packets into the multicast tree data path.

5.1. Baseline Secure ALC Operation

This section describes a baseline mode of secure ALC protocol operation based on application of the IPsec security protocol. This approach is documented here to provide a reference of an interoperable secure mode of operation. However, additional approaches to ALC security, including other forms of IPsec application, MAY be specified in the future. For example, the use of the EXT_AUTH Header Extension could enable ALC-specific authentication or security encapsulation headers similar to those of IPsec to be specified and inserted into the ALC/LCT protocol message headers. This would allow header compression techniques to be applied to IP and ALC protocol headers when needed in a similar fashion to that of RTP [RFC3550] and as preserved in the specification for Secure Real Time Protocol (SRTP) [RFC3711].

The baseline approach described is applicable to ALC operation configured for SSM (or SSM-like) operation where there is a single sender. The receiver set would maintain a single IPsec Security Association (SA) for each ALC sender.

5.1.1. IPsec Approach

To support this baseline form of secure ALC one-to-many SSM operation, each node SHALL be configured with a transport mode IPsec Security Association and corresponding Security Policy Database (SPD) entry. This entry will be used for sender-to-group multicast packet authentication and optionally encryption.

The ALC sender SHALL use an IPsec SA configured for Encapsulating Security Payload (ESP) protocol [RFC4303] operation with the option for data origination authentication enabled. It is also RECOMMENDED that this IPsec ESP SA be also configured to provide confidentiality protection for IP packets containing ALC protocol messages. This is suggested to make the realization of complex replay attacks much more

difficult. The encryption key for this SA SHALL be preplaced at the sender and receiver(s) prior to ALC protocol operation. Use of automated key management is RECOMMENDED as a rekey SHALL be required prior to expiration of the sequence space for the SA. This is necessary so that receivers may use the built-in IPsec replay attack protection possible for an IPsec SA with a single source (the ALC sender). Thus, the receivers SHALL enable replay attack protection for this SA used to secure ALC sender traffic. The sender IPsec SPD entry MUST be configured to process outbound packets to the destination address and UDP port number of the applicable ALC session.

The ALC receiver(s) MUST be configured with the SA and SPD entry to properly process the IPsec-secured packets from the sender. Note that use of ESP confidentiality, as RECOMMENDED, for secure ALC protocol operation makes it more difficult for adversaries to conduct effective replay attacks that may mislead receivers on content reception. This baseline approach can be used for ALC protocol sessions with multiple senders if a distinct SA is established for each sender.

In early deployments of this baseline approach to ALC security, it is anticipated that key management will be conducted out-of-band with respect to ALC protocol operation. For ALC unidirectional operation, it is possible that receivers may retrieve keying information from a central server via out-of-band (with respect to ALC) communication as needed or otherwise conduct group key updates with a similar centralized approach. However, it may be possible with some key management schemes for rekey messages to be transmitted to the group as a message or transport object within the ALC reliable transfer session. An additional specification is necessary to define an in-band key management scheme for ALC sessions perhaps using the mechanisms of the automated group key management specifications cited in this document.

5.1.2. IPsec Requirements

In order to implement this secure mode of ALC protocol operation, the following IPsec capabilities are required.

5.1.2.1. Selectors

The implementation MUST be able to use the source address, destination address, protocol (UDP), and UDP port numbers as selectors in the SPD.

5.1.2.2. Mode

IPsec in transport mode MUST be supported. The use of IPsec [RFC4301] processing for secure ALC traffic SHOULD also be REQUIRED such that unauthenticated packets are not received by the ALC protocol implementation.

5.1.2.3. Key Management

An automated key management scheme for group key distribution and rekeying such as the Group Domain of Interpretation (GDOI) [RFC3547], Group Secure Association Key Management Protocol (GSAKMP) [RFC4535], or Multimedia Internet KEYing (MIKEY) [RFC3830] SHOULD be used. Relatively short-lived ALC sessions MAY be able to use Manual Keying with a single, preplaced key, particularly if Extended Sequence Numbering (ESN) [RFC4303] is available in the IPsec implementation used. It should also be noted that it may be possible for key update messages (e.g., the GDOI GROUPKEY-PUSH message) to be included in the ALC application reliable data transmission as transport objects if appropriate interfaces were available between the ALC application and the key management daemon.

5.1.2.4. Security Policy

Receivers SHOULD accept connections only from the designated, authorized sender(s). It is expected that appropriate key management will provide encryption keys only to receivers authorized to participate in a designated session. The approach outlined here allows receiver sets to be controlled on a per-sender basis.

5.1.2.5. Authentication and Encryption

Large ALC group sizes may necessitate some form of key management that does rely upon shared secrets. The GDOI and GSAKMP protocols mentioned here allow for certificate-based authentication. These certificates SHOULD use IP addresses for authentication. However, it is likely that available group key management implementations will not be ALC-specific.

5.1.2.6. Availability

The IPsec requirements profile outlined here is commonly available on many potential ALC hosts. The principal issue is that configuration and operation of IPsec typically requires privileged user authorization. Automated key management implementations are typically configured with the privileges necessary to allow the needed system IPsec configuration.

6. IANA Considerations

This specification registers one value in the LCT Header Extensions Types namespace as follows:

Value	Name	Reference
64	EXT_FTI	This specification

7. Acknowledgments

This specification is substantially based on RFC 3450 [RFC3450] and thus credit for the authorship of this document is primarily due to the authors of RFC 3450: Mike Luby, Jim Gemmel, Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. Vincent Roca, Justin Chapweske, and Roger Kermode also contributed to RFC 3450. Additional thanks are due to Vincent Roca and Rod Walsh for contributions to this update to Proposed Standard.

8. Changes from RFC 3450

This section summarizes the changes that were made from the "Experimental" version of this specification published as RFC 3450 [RFC3450]:

- Updated all references to the obsoleted RFC 2068 to RFC 2616.
- Removed the 'Statement of Intent' from the introduction. (The Statement of Intent was meant to clarify the "Experimental" status of RFC 3450.)
- Removed the 'Intellectual Property Issues' Section and replaced with a standard IPR Statement.
- Removed material duplicated in LCT.
- Updated references in this document to new versions of the LCT Building Block and the FEC Building Block, and aligned this document with changes in the new version of the FEC Building Block.
- Split normative and informative references.
- Material applicable in a general LCT context, not just for ALC has been moved to LCT.

- The first bit of the "Protocol-Specific Indication" in the LCT Header is defined as a "Source Packet Indication". This is used in the case that an FEC Scheme defines two FEC Payload ID formats, one of which is for packets containing only source symbols that can be processed by receivers that do not support FEC Decoding.

- Definition and IANA registration of the EXT_FTI LCT Header Extension.

9. References

9.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3738] Luby, M. and V. Goyal, "Wave and Equation Based Rate Control (WEBRC) Building Block", RFC 3738, April 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

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

[RFC5052] Watson, M., Luby, M., and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, August 2007.

[RFC5651] Luby, M., Watson, M., and L. Vicisano, "Layered Coding Transport (LCT) Building Block", RFC 5651, October 2009.

9.2. Informative References

[RFC2357] Mankin, A., Romanov, A., Bradner, S., and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", RFC 2357, June 1998.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S., and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", RFC 3048, January 2001.

[RFC3269] Kermode, R. and L. Vicisano, "Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks and Protocol Instantiation documents", RFC 3269, April 2002.

[RFC3450] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., and J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 3450, December 2002.

[RFC3453] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

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

[RFC3569] Bhattacharyya, S., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, July 2003.

[RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[RFC5776] Roca, V., Francillon, A., and S. Faurite, "Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols", RFC 5776, April 2010.

[RMT-SIMPLE] Roca, V., "Simple Authentication Schemes for the ALC and NORM Protocols", Work in Progress, October 2009.

Authors' Addresses

Michael Luby
Qualcomm, Inc.
3165 Kifer Road
Santa Clara, CA 95051
US

EMail: luby@qualcomm.com

Mark Watson
Qualcomm, Inc.
3165 Kifer Road
Santa Clara, CA 95051
US

EMail: watson@qualcomm.com

Lorenzo Vicisano
Qualcomm, Inc.
3165 Kifer Road
Santa Clara, CA 95051
US

EMail: vicisano@qualcomm.com