

Intarea Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 13, 2014

C. Bormann
Universitaet Bremen TZI
September 09, 2013

Adaptation Layer Fragmentation Indication
draft-bormann-intarea-alfi-04

Abstract

IPv6 defines a minimum MTU of 1280 bytes. Many link layers are more limited in the maximum size of packets they can communicate. In order to enable the transport of IP packets that are too large for these link layers, typically their IP adaptation layers define a segmentation or fragmentation scheme to transport an IP packet in a sequence of multiple link layer packets.

Often, adaption layer fragmentation schemes reduce some performance metric, such as the packet delivery probability. Application or transport protocols may be able to reduce the maximum size of packets they send, e.g. by transport layer segmentation or choice of application layer data object size, which may have less of a performance impact. It would therefore be desirable for them to know about any adaptation layer fragmentation that is going on, so they can choose packet sizes that minimize adaptation layer fragmentation.

At the IP layer, fragmentation can be detected using a number of mechanisms used in Packetization Layer Path MTU Discovery [[RFC4821](#)]. However, adaptation layer fragmentation schemes are often designed to be "transparent", i.e. there is no way at higher layers to find out whether they had to be employed (except maybe by elaborate measurement schemes targeting one of the impacted performance metrics; this approach does not appear to be viable) [[WEI](#)].

The present specification defines two alternative mechanisms for IPv6 adaptation layers to indicate the presence of adaptation layer fragmentation on one or more hops on the path from an IP sender to an IP receiver, and to provide an indication of preferred (smaller) packet sizes on these hops.

One design is based on the the IPv6 design and probably doesn't work on the Internet. The other design goes strictly against the IPv6 design and probably works well on the Internet.

Comments are appreciated and should go to the intarea@ietf.org mailing list.

Internet-Draft Adaptation Layer Fragmentation Indication September 2013

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 March 13, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
1.1.	Terminology	3
2.	Objectives and Considerations	3
3.	The ALFI option	4
4.	Discussion	6
4.1.	Should this be done at all?	6
4.2.	Is this the right way to do it?	6
5.	Another way of doing it	6
5.1.	Discussion	7

6.	IANA Considerations	8
7.	Security Considerations	8
8.	Acknowledgements	8
9.	References	8
9.1.	Normative References	8

Internet-Draft Adaptation Layer Fragmentation Indication September 2013

9.2.	Informative References	8
Author's Address	10

[1.](#) Introduction

(To be written - for now please read the Abstract.)

[1.1.](#) Terminology

The following terms are used in this specification:

ALF: Adaptation Layer Fragmentation.

MUALTU: Maximum Unfragmented Adaptation Layer Transmission Unit, i.e. the largest piece of IPv6 packet (measured in bytes) that can be transferred by the adaptation layers on the path without invoking ALF.

IFMUALTU: Initial-Fragment MUALTU, the MUALTU for the initial adaptation layer fragment of an IP packet.

FFMUALTU: Following-Fragment MUALTU, the estimated minimum MUALTU for all but the initial adaptation layer fragments of an IP packet.

ALFI: Adaptation Layer Fragmentation Indication, i.e. indication that ALF was performed on a packet.

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](#)] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

The term "byte" is used in its now customary sense as a synonym for "octet".

2. Objectives and Considerations

This draft is shaped by the requirements of 6LoWPAN networks [[I-D.bormann-6lowpan-roadmap](#)], including variants such as Bluetooth/Low Energy [[I-D.ietf-6lowpan-btle](#)] or DECT/ULE [[I-D.mariager-6lowpan-v6over-dect-ule](#)]. However, it should be beneficial with any adaptation layer that requires the use of ALF.

One important consideration for ALFI is that the ALF scheme may not be able to provide a consistent MUALTU. E.g., header compression may cause variable overheads, and initial and following fragments are

likely to cause different MUALTUs. Header compression may be dependent on the specific characteristics of the packets employed, so indications will be most accurate if they can be made on the basis of actual packets as they are intended to be transferred. (E.g., for a 6LoWPAN with a physical layer maximum packet size of 127 bytes, the specific combination of MAC layer overheads, adaptation layer overheads, and header compression gains can turn the actual initial fragment size number into anything roughly between 70 and 160 bytes. Note that this is more than a factor of two, making it difficult to just guess a good MUALTU.)

Therefore, ALFI provides the ability to equip packets with a probe that collects any information for adaptation layer fragmentation that may be available on the path.

Note that probing for MUALTUs is likely to change the MUALTU. Implementations SHOULD attempt to indicate a MUALTU for an equivalent non-probe packet, i.e. the packet under consideration with the ALFI option (and its hop-by-hop header, if applicable) removed. If that is not possible, implementations SHOULD err towards indicating smaller MUALTUs, within reason.

Obviously, not all nodes will immediately implement ALFI. ALFI just "fails ignorant" (but see below).

An adaptation layer instance may want to manipulate ALFI for other reasons than to indicate ALF (which would be somewhat comparable to the widespread practice of TCP "MSS clamping"). (In particular, as long as it can be expected that some other nodes on the path don't

have ALFI yet, a border router such as a 6LBR [RFC6775] may want to provide some ALFI guessing.)

Generally speaking, ALFI can be used as a mechanism to indicate any significant, step function degradation of some performance metric based on packet size. However, as the mechanism can only collect a single value for the entire path (i.e., one IFMUALTU and one FFMUALTU), the performance degradation indicated SHOULD be significant. In other words, ALFI indications SHOULD NOT be set for segmentation implementations where segmentation causes limited performance impact. E.g., AAL5 implementations SHOULD NOT set ALFI.

3. The ALFI option

The ALFI option is an IPv6 option in the sense of [section 4.2 of \[RFC2460\]](#). It is only used in the hop-by-hop header.

The option type identifier is chosen to select the following behavior as detailed in [section 4.2 of \[RFC2460\]](#):

- o 00 - skip over this option and continue processing the header (this enables the "fail-ignorant" backwards compatibility behavior)
- o 1 - Option Data may change en-route (the option is used to record information en-route)

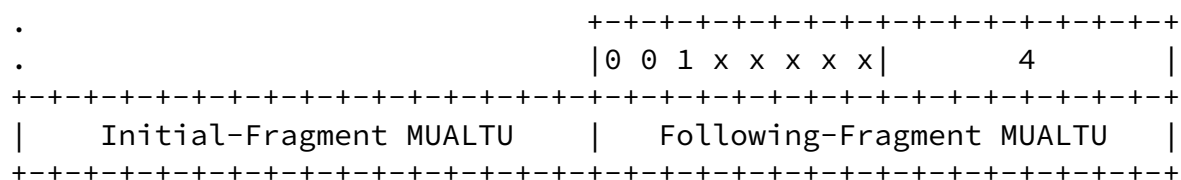


Figure 1: A hop-by-hoption for ALFI

In IFMUALTU and FFMUALTU, the value zero represents infinity. All other values are unsigned integers in network byte order, representing a MUALTU in bytes.

The originator of a packet MAY, for occasional probing, insert an ALFI option into packets where it can choose the packet size and the performance metrics of which are important to the application.

When generating the IP packet, the originator sets Initial-Fragment MUALTU (IFMUALTU) and Following-Fragment MUALTU (FFMUALTU) to zero. (Its own adaptation layer can then already update them as described in the following paragraphs before the packet even leaves the originator.)

Each instance of an adaptation layer that employs ALF and that implements this specification computes its own estimate of IFMUALTU and FFMUALTU for the type of packet that has this option, ignoring the option itself and, if the option was the only option in the hop-by-hop header, the hop-by-hop header. For each estimate, if it is below the value of the respective field encoded in the option (where zero represents infinity), the instance updates the field to the estimate.

The receiver of the packet relays the information in the ALFI option to the transport layer and/or application.

(TBD: How to ship this information through the IPv6 socket interface [[RFC3493](#)] [[RFC3542](#)]. Constrained implementations won't have this specific problem.)

The receiving transport layer and/or application can then make this information available back to the peer instance, which enables the latter to choose IPv6 packet sizes of IFMUALTU or lower, or, if this

cannot be achieved, at least below $IFMUALTU + n * FFMUALTU$ for a small n . For instance, in CoAP [[I-D.ietf-core-coap](#)], the receiver of an ALFI probe from a server can use the Block2 option [[I-D.ietf-core-block](#)] to negotiate a block size for further messages in a block-wise transfer accordingly.

[4.](#) Discussion

The discussion of this proposal should center around two questions.

[4.1.](#) Should this be done at all?

I.e., can't we just live with the inefficiency?

In ATM or DVB, the inefficiency caused by not knowing about

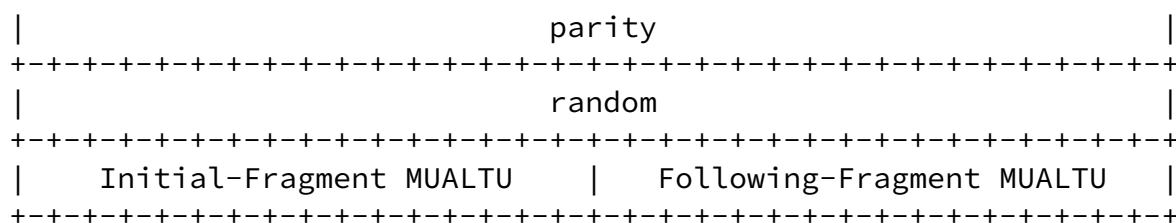


Figure 2: A STUN-like UDP payload for ALFI

The parity field is computed to make the 32-bit XOR of all five words zero (even parity). The random field is filled with a suitable pseudorandom number (with no requirement for cryptographic quality). All parameters of the IP and UDP header are chosen as for the data flow for which the MUALTU values are needed.

A router that implements ALFI needs to detect transiting UDP packets that match the structure of the ALFI probe. When updating the values, both the parity field in the ALFI payload and the UDP checksum need to be updated as well.

5.1. Discussion

Carrying ALFI probes in UDP data packets enables correspondent nodes on general purpose operating systems to send and receive these probes without any special interface such as that defined in [[RFC3542](#)].

For best quality of the header compression performance prediction, ALFI probes are required to look very similar to the actual data packets. This means this approach only works with protocols using UDP payloads. For use with CoAP [[I-D.ietf-core-coap](#)], this is not a problem.

The application protocol must be able to ignore packets that look like ALFI probes. Again, for use with CoAP [[I-D.ietf-core-coap](#)], this is not a problem. (The design might be refined to enable interoperability with other protocols as desired.)

by creating damage to the data transparency of the Internet. The conditions under which this is an acceptable trade-off must be carefully evaluated.

6. IANA Considerations

For the hop-by-hop option approach described in [Section 3](#), IANA needs to allocate an IPv6 option number for the ALFI option, "Destination Options and Hop-by-Hop Options" registry in "Internet Protocol Version 6 (IPv6) Parameters", with act=00 and chg=1 (i.e., similar to the Quick-Start option [[RFC4782](#)]).

For the STUN-related approach described in [Section 5](#), the message type 0b10_010_011_1000 = 0x938 should be reserved in the STUN Methods Registry.

7. Security Considerations

It is hard to like hop-by-hop options from a security point of view.

(This section will certainly grow as additional security considerations beyond those listed in the base specifications become known.)

8. Acknowledgements

Peter van der Stok prompted the author to finally write up this protocol, a couple of years after the need for it had been shown in [[WEI](#)]. He then also provided a number of editorial comments that improved the document.

Toerless Eckert pointed out that routers are today able to do on-path inspection, the approach that makes [Section 5](#) possible.

9. References

9.1. Normative References

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

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.

9.2. Informative References

[I-D.bormann-6lowpan-roadmap]

Bormann, C., "6LoWPAN Roadmap and Implementation Guide", [draft-bormann-6lowpan-roadmap-04](#) (work in progress), April 2013.

[I-D.ietf-6lowpan-btle]

Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH Low Energy", [draft-ietf-6lowpan-btle-12](#) (work in progress), February 2013.

[I-D.ietf-core-block]

Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP", [draft-ietf-core-block-12](#) (work in progress), June 2013.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", [draft-ietf-core-coap-18](#) (work in progress), June 2013.

[I-D.ietf-lwig-terminology]

Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained Node Networks", [draft-ietf-lwig-terminology-05](#) (work in progress), July 2013.

[I-D.mariager-6lowpan-v6over-dect-ule]

Mariager, P., Petersen, J., and Z. Shelby, "Transmission of IPv6 Packets over DECT Ultra Low Energy", [draft-mariager-6lowpan-v6over-dect-ule-03](#) (work in progress), July 2013.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", [RFC 3493](#), February 2003.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", [RFC 3542](#), May 2003.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", [RFC 4782](#), January 2007.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", [RFC 4821](#), March 2007.

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power

Wireless Personal Area Networks (6LoWPANs)", [RFC 6775](#),
November 2012.

Bormann

Expires March 13, 2014

[Page 9]

Internet-Draft Adaptation Layer Fragmentation Indication September 2013

[WEI] Shelby, Z. and C. Bormann, "6LoWPAN: the Wireless Embedded
Internet", ISBN 9780470747995, 2009.

Author's Address

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Bormann

Expires March 13, 2014

[Page 10]