6lo
Internet-Draft
Updates: 4944, 6282 (if approved)
Intended status: Standards Track
Expires: April 21, 2016

S. Chakrabarti
Ericsson
G. Montenegro
Microsoft
R. Droms
Cisco
J. Woodyatt
Nest

October 19, 2015

IANA Registry for 6lowpan ESC Dispatch Code points
draft-chairs-6lo-dispatch-iana-registry-01

Abstract

RFC4944 defines ESC dispatch type for additional dispatch bytes in the 6lowpan header. The value of ESC byte has been updated by RFC6282. However, the usage of ESC extension byte has not been defined in RFC6282 and RFC4944. The purpose of this document is to define the ESC extension byte code points and to request corresponding IANA actions.

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 April 21, 2016.

Copyright Notice

Table of Contents

1. Introduction

[RFC4944] section 5.1 defines the dispatch header and types. The ESC type is defined for using additional dispatch bytes in the 6lowpan header. RFC 6282 modifies the value of the ESC dispatch type and it is recorded in IANA registry [6LOWPAN-IANA]. However, the bytes and usage following the ESC byte are not defined in either [RFC4944] and [RFC6282]. However, in recent years with 6lowpan deployments, the implementations and Standards organizations have started using the ESC extension bytes and a co-ordination between the respective organizations and IETF/IANA are needed.

The following sections record the ITU-T specification for ESC dispatch byte code points as an existing known usage and propose the definition of ESC extension bytes for future applications. The document also requests IANA actions for the first extension byte following the ESC byte.

2. Terminology

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].

3. Usage of ESC dispatch bytes

RFC 4944 [RFC4944] first introduces this "ESC" dispatch header type for extension of dispatch bytes. RFC 6282 [RFC6282] subsequently modified its value to [01 000000].

This document specifies that the first octet following the ESC byte be used for extension type(extended dispatch values). Subsequent octets are left unstructured for the specific use of the extension type:

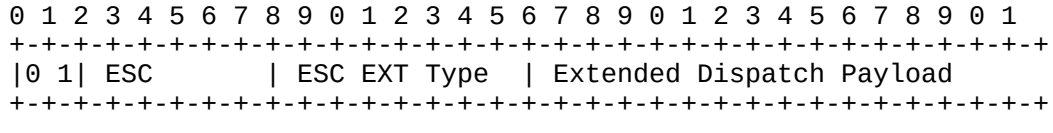


Figure 1: Frame Format with ESC Byte

ESC: The left-most byte is the ESC dispatch type containing ’0100000’

ESC Extension Type(EET): It is the first byte following the ESC byte. Extension type defines the payload for the additional dispatch bytes. The values are from 0 to 255. Value 0 and 255 are reserved for future use. These values are assigned by IANA. The EET values are similar to dispatch values in the 6lowpan header except they are preceeded by the ESC byte. Thus, ESC extension types and dispatch values are using orthogonal code spaces. Though not desirable, multiple ESC bytes MAY appear in a 6lowpan header. Section 3.1 describes how to handle unknown ESC dispatch type.

Extended Dispatch Payload(EDP): This part of frame format must be defined by the corresponding extension type. A specification is required to define each usage of extension type and its corresponding Extension Payload.

Note that section 5.1 in RFC4944 indicates that the Extension Type field may contain additional dispatch values larger than 63 [4944-ERRATA]. Note that the new dispatch type MUST NOT modify the behavior of existing dispatch types for the sake of interoperability.

3.1. Interaction with other RFC4944 implementations

It is expected that RFC4944 existing implementations are not capable of processing ESC extension data bytes as defined in this document. However, implementors have to assume that existing implementation that attempt to process an EET unknown to them will simply drop the packet or ignore the ESC dispatch bytes.

If an implementation following this document, during processing of the received packet reaches the ESC byte for which it does not understand the extension bytes (EET), it MUST drop that packet. However, it is important to clarify that a router node SHOULD forward a 6lowpan packet with the EET bytes as long as it does not attempt to process any ESC extension bytes.

Sequence Of dispatch bytes and ESC bytes: Multiple ESC extension bytes may appear in a packet. Could a 6lowpan packet start with a ESC dispatch type? In another word, should ESC extension always be preceeded by non-ESC dispatch bytes?

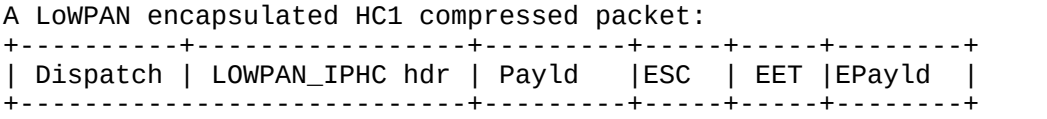
gab: I think the answer is no. But per the previous sentence, you have to assume that your packet will get dropped immediately by any node that does not understand the EET at the beginning of the packet. The closer to the end of the packet are the EET’s, the higher chance there is that a legacy node will recognize and successfully process some dispatch type before the EET and then ignore the EET instead of dropping the entire packet. Unless you know for sure that all nodes in your network understand a given EET (by definition a private and

non-standard deployment), placing it at the beginning is a good way to guarantee that the packet will get dropped.

3.2. ESC Extension Bytes Typical Sequence

The following diagram provides an example when ESC extension bytes might be used:

A LoWPAN encapsulated HC1 compressed packet:



A LoWPAN_IPHC Header, Mesh header and an ESC extenstion byte:

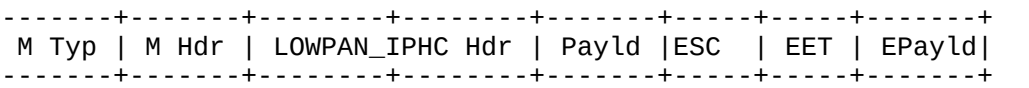


Figure 2: A 6lowpan packet with ESC Bytes

3.3. Example: ITU-T G.9903 ESC type usage

[G3-PLC] provides native mesh-under functionalities. The ESC dispatch type is used with the command frames specified in figure 9-12 and Table 9-35 in [G3-PLC] . The command ID values are 0x01 to 0x1F.

The frame format is defined as follows:

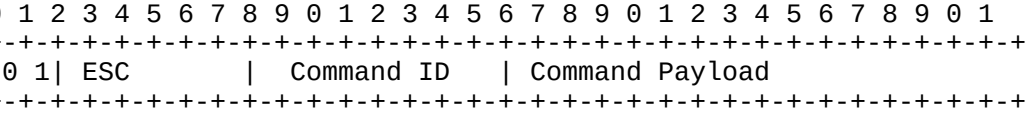


Figure 3: G.9903 Frame Format with ESC Byte

3.4. NALP Usage

There were several comments on 00 draft -- that this draft should provide guidance on NALP usage as there was no clear distinction between ITU-T command mode usage and NALP usage. In order to avaoid such confusion, a NALP usage guidance should be provided. This is a space holder section in order to decide whether NALP usage indeed should belong here.

gab: I don’t think we need to say anything beyond what we already say in 4944: it is not a 6lowpan frame. This was done recognizing that some SDO’s would also define their own frame structure, in particular, Zigbee. There was some effort to agree with them on some way for our definitions to not collide. So prescribing usage of NALP, beyond saying it is not 6lowpan nor the subject of any IETF document, would defeat the purpose.

4. IANA Considerations

This document requests IANA to register the ’ESC Extension Type’ values as per the policy ’Specification Required’[RFC5226] as specified in this document which follows the same policy as in the IANA section of [RFC4944]. For each Extension Type(except the Reserved values)the specification MUST define corresponding Extended Dispatch Payload frame bytes for the receiver implementation to read the ESC bytes with interoperability.

The initial values for the ’ESC Extension Type’ fields are:

Value	Description	Reference
0	Reserved for future use	This document
1-31	Used by ITU-T G.9903 and G.9905 Command IDs	ITU-T G.9903 & ITU-T G.9905
32-254	Unassigned (Reserved for future IANA Assignment-- Spec Required)	This document
255	Reserved for future use	This document

Figure 4: Initial Values for IANA Registry

5. Security Considerations

There is no additional security threats due to the assignments of ESC byte usage described in this document. However, this document forbids defining any extended dispatch values or extension types that modifies the behavior of existing Dispatch types.

6. Acknowledgements

The authors would like to thank the members of the 6lo WG members for the comments in the mailing list. Many thanks to Carsten Bormann, Ralph Droms, Thierry Lys, Cedric Lavenu, Pascal Thubert for their discussions regarding resolving the bits allocation issues which led to this document. Jonathan Hui and Robert Cregie provided extensive reviews and guidance for interoperability. The authors acknowledges the comments from the following people for shaping this document: Paul Duffy, Don Sturek, Michael Richardson, Xavier Vilajosana and Scott Mansfield.

7. References

7.1. Normative References

[4944-ERRATA]
"https://www.rfc-editor.org/errata_search.php?rfc=4944".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

7.2. Informative References

[6LOWPAN-IANA]
"https://www.iana.org/assignments/_6lowpan-parameters/_6lowpan-parameters.xhtml".

[G3-PLC] "http://www.itu.int/rec/T-REC-G.9903-201402-I".

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

Authors' Addresses

Samita Chakrabarti
Ericsson
300 Holger Way
San Jose, CA
US

Phone: +1 408 750 5843
Email: samita.chakrabarti@ericsson.com

Gabriel Montenegro
Microsoft
Seattle
US

Email: gabriel.montenegro@microsoft.com

Ralph Droms
Cisco
USA

Email: rdroms@cisco.com

James Woodyatt
Nest
Mountain View, CA
USA

Email: jhw@netstlabs.com