



OPEN NETWORKING
FOUNDATION

Multipart timeout errors Extension

Version 0.1

December 23, 2014



Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Without limitation, ONF disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and ONF disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

No license, express or implied, by estoppel or otherwise, to any Open Networking Foundation or Open Networking Foundation member intellectual property rights is granted herein.

Except that a license is hereby granted by ONF to copy and reproduce this specification for internal use only.

Contact the Open Networking Foundation at <http://www.opennetworking.org> for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

WITHOUT LIMITING THE DISCLAIMER ABOVE, THIS SPECIFICATION OF THE OPEN NETWORKING FOUNDATION ("ONF") IS SUBJECT TO THE ROYALTY FREE, REASONABLE AND NONDISCRIMINATORY ("RANDZ") LICENSING COMMITMENTS OF THE MEMBERS OF ONF PURSUANT TO THE ONF INTELLECTUAL PROPERTY RIGHTS POLICY. ONF DOES NOT WARRANT THAT ALL NECESSARY CLAIMS OF PATENT WHICH MAY BE IMPLICATED BY THE IMPLEMENTATION OF THIS SPECIFICATION ARE OWNED OR LICENSABLE BY ONF'S MEMBERS AND THEREFORE SUBJECT TO THE RANDZ COMMITMENT OF THE MEMBERS.

Contents

1	Introduction	3
2	How it works	3
3	Multipart timeout Experimenter ID	3
4	Multipart timeout errors	4

1 Introduction

This document describes an ONF extension for OpenFlow version 1.3.X that returns errors for multipart requests and reply taking too long.

2 How it works

Two new error code are defined for timeout of multipart messages.

Multipart request are composed of a sequence of multiple messages from the controller, and the last message has the `OFPMPPF_REQ_MORE` flag set to zero to indicate the end of the sequence. If a switch receives a multipart request message sequence that does not include a message with `OFPMPPF_REQ_MORE` flag set to zero, after a switch defined amount of time greater than 100 milliseconds from the last message, the switch must discard the incomplete multipart request, and may generate a `ONFERR_ET_MPART_REQUEST_TIMEOUT` error message to the controller.

Multipart replies are composed of a sequence of multiple messages from the switch, and the last message has the `OFPMPPF_REPLY_MORE` flag set to zero to indicate the end of the sequence. If a controller receives a multipart reply message sequence that does not include a message with `OFPMPPF_REPLY_MORE` flag set to zero, after a controller defined amount of time greater than 1 second from the last message, the controller must discard the incomplete multipart reply, and may generate a `ONFERR_ET_MPART_REPLY_TIMEOUT` error message to the switch. The controller may retry the failed switch operation as needed as a new multipart request.

3 Multipart timeout Experimenter ID

The Experimenter ID of this extension is:

`ONF_EXPERIMENTER_ID = 0x4F4E4600`

4 Multipart timeout errors

The following errors are defined by this extension:

```
/* Error codes */
enum onf_error_exp_type {
    ONFERR_ET_MPART_REQUEST_TIMEOUT = 2640, /* Timeout during multipart request. */
    ONFERR_ET_MPART_REPLY_TIMEOUT = 2641,   /* Timeout during multipart reply. */
};
```

The error `ONFERR_ET_MPART_REQUEST_TIMEOUT` and `ONFERR_ET_MPART_REPLY_TIMEOUT` use the following structure:

```
/* Message structure for all errors. */
struct onf_error_msg {
    struct ofp_header header;
    uint16_t type;           /* OFPET_EXPERIMENTER. */
    uint16_t exp_type;       /* One of ONFERR_ET_* above. */
    uint32_t experimenter;    /* ONF_EXPERIMENTER_ID. */
    uint8_t data[0];         /* Up to 64 bytes of failed request. */
};
OFP_ASSERT(sizeof(struct onf_error_header) == sizeof(struct ofp_error_experimenter_msg));
```

The `type` field must be set to `OFPET_EXPERIMENTER`.

The `experimenter` field is the Experimenter ID (see 3).

The `data` fields contains a copy of the failed request message, truncated to 64 bytes.

The `exp_type` field is set to `ONFERR_ET_MPART_REQUEST_TIMEOUT` or `ONFERR_ET_MPART_REPLY_TIMEOUT`.