OPEN NETWORKING
FOUNDATION

Vacancy events Extension

Version 0.1

December 23, 2014

OpenFlow

Disclaimer

Contents

1 Introduction

This document describes an ONF extension for OpenFlow version 1.3.X that generate vacancy events for flow tables. When the vacancy of a flow table cross a threshold configured by the controller, an event is generated by the switch.

2 How it works

Using this extension, a controller may set two vacancy threshold for each flow table, each threshold is express as a percentage vacancy. Two thresholds are to avoid spurious events using an hysteresis mechanism.

The vacancy events are generated when the remaining space in the flow table changes and cross a thereshold, whatever causes the change in remaining space. This can be the result of OpenFlow operations on the table (flow insertion, expiry or removal), or the result of switch internal processing changing the amount of remaining space in the table.

When the remaining space in the flow table decreases to less than `vacancy_down`, and if vacancy down events are enabled, a vacancy down event must be generated to the controller using the `ONF_ET_VACANCY_STATUS` message type with reason `ONFVR_DOWN`. Further vacancy down events are disabled until a vacancy up event is generated.

When the remaining space in the flow table increases to more than `vacancy_up`, and if vacancy up events are enabled, a vacancy up event must be generated to the controller using the `ONF_ET_VACANCY_STATUS` message type with reason `ONFVR_UP`. Further vacancy up events are disabled until a vacancy down event is generated.

When vacancy thereshold are set on a table and `vacancy_down` is not zero, either the vacancy up or vacancy down event is enabled. When enabling events, if the current vacancy is less than `vacancy_up`, vacancy up events must be enabled, and vacancy down events must be disabled. When enabling events, if the current vacancy is greater or equal to `vacancy_up`, vacancy down events must be enabled, and vacancy up events must be disabled. When `vacancy_down` is set to zero, both vacancy up and vacancy down events must be disabled.

3 Vacancy events Experimenter ID

The Experimenter ID of this extension is:

`ONF_EXPERIMENTER_ID` = 0x4F4E4600

4 Vacancy events messages

The following message types are defined by this extension.

```
/* Message types */
enum onf_exp_type {
  ONF_ET_SET_VACANCY         = 1920, /* Set vacancy - Controller/switch message */
  ONF_ET_GET_VACANCY_REQUEST = 1921, /* Get vacancy - Controller/switch message */
  ONF_ET_GET_VACANCY_REPLY   = 1922, /* Get vacancy - Controller/switch message */
  ONF_ET_VACANCY_STATUS      = 1923, /* Vacancy event - Async message */
};
```

The `ONF_ET_SET_VACANCY` message is used by the controller to set the vacancy thresholds on a flow table, and uses the following message structure :

```
/* Set vacancy message. */
struct onf_message_set_vacancy {
  struct ofp_header     header;
  uint32_t              experimenter;      /* ONF_EXPERIMENTER_ID. */
  uint32_t              exp_type;          /* ONF_ET_SET_VACANCY. */
  uint8_t table_id;       /* ID of the table, OFPTT_ALL indicates all tables */
  uint8_t pad[1];         /* Pad to 16 bits */
  uint8_t vacancy_down;   /* Vacancy threshold when space decreases (%). */
  uint8_t vacancy_up;     /* Vacancy threshold when space increases (%). */
  uint8_t pad2[4];        /* Pad to 64 bits */
};
OFP_ASSERT(sizeof(struct onf_message_set_vacancy) == sizeof(struct ofp_experimenter_header) + 8);
```

The `table_id` field is the table which threshold should be changed.

The fields `vacancy_down` and `vacancy_up` are the threshold for generating vacancy events that should be configured on this flow table, expressed as a percent. If `vacancy_down` field is set to 0, vacancy events are disabled on the flow table.

The `ONF_ET_GET_VACANCY_REQUEST` message is used by the controller to request the current vacancy thresholds from a flow table, and uses the following message structure :

```
/* Get vacancy request message. */
struct onf_message_get_vacancy_request {
  struct ofp_header     header;
  uint32_t              experimenter;      /* ONF_EXPERIMENTER_ID. */
  uint32_t              exp_type;          /* ONF_ET_GET_VACANCY_REQUEST. */
  uint8_t table_id;       /* ID of the table. */
  uint8_t pad[7];         /* Pad to 64 bits */
};
OFP_ASSERT(sizeof(struct onf_message_get_vacancy_request) == sizeof(struct ofp_experimenter_header) + 8);
```

The `table_id` field is the table which threshold are requested.

The `ONF_ET_GET_VACANCY_REPLY` message is used by the switch to reply to a `ONF_ET_GET_VACANCY_REQUEST` message from the controller and report the current vacancy thresholds of a flow table. It uses the following message structure :

```
/* Get vacancy reply message. */
struct onf_message_get_vacancy_reply {
  struct ofp_header     header;
  uint32_t              experimenter;      /* ONF_EXPERIMENTER_ID. */
  uint32_t              exp_type;          /* ONF_ET_GET_VACANCY_REPLY. */
  uint8_t table_id;       /* ID of the table, OFPTT_ALL indicates all tables */
  uint8_t vacancy;        /* Current vacancy (%). */
  uint8_t vacancy_down;   /* Vacancy threshold when space decreases (%). */
  uint8_t vacancy_up;     /* Vacancy threshold when space increases (%). */
  uint8_t pad2[4];        /* Pad to 64 bits */
};
OFP_ASSERT(sizeof(struct onf_message_get_vacancy_reply) == sizeof(struct ofp_experimenter_header) + 8);
```

The `table_id` field is the table which threshold are reported.

The field `vacancy` is the current vacancy of the table, expressed as a percent.

The fields `vacancy_down` and `vacancy_up` are the threshold for generating vacancy events configured on this flow table, expressed as a percent. If vacancy events are disabled on the table, the `vacancy_down` field must be set to 0.

The `ONF_ET_VACANCY_STATUS` message is generated by the switch when a vacancy event is crossed on a flow table. It uses the following message structure :

```
/* Vacancy event message. */
struct onf_message_vacancy_status {
  struct ofp_header     header;
  uint32_t              experimenter;      /* ONF_EXPERIMENTER_ID. */
  uint32_t              exp_type;          /* ONF_ET_SET_VACANCY. */
  uint8_t table_id;       /* ID of the table, OFPTT_ALL indicates all tables */
  uint8_t vacancy;        /* Current vacancy (%). */
  uint8_t vacancy_down;   /* Vacancy threshold when space decreases (%). */
  uint8_t vacancy_up;     /* Vacancy threshold when space increases (%). */
  uint8_t reason;         /* One of ONFVR_*. */
  uint8_t pad2[3];        /* Pad to 64 bits */
};
OFP_ASSERT(sizeof(struct onf_message_vacancy_status) == sizeof(struct ofp_experimenter_header) + 8);
```

The `table_id` field is the table which vacancy is reported.

The field `vacancy` is the current vacancy of the table, expressed as a percent.

The fields `vacancy_down` and `vacancy_up` are the threshold for generating vacancy events configured on this flow table, expressed as a percent.

The `reason` field can be one of the following values:

```
/* Reason for the vacancy event. */
enum onf_vacancy_reason {
  ONFVR_DOWN = 0,    /* Crossed vacancy_down threshold. */
  ONFVR_UP   = 1,    /* Crossed vacancy_up threshold. */
};
```

5 Vacancy errors

The following errors are defined by this extension:

```
/* Error codes */
enum onf_error_exp_type {
  ONFERR_ET_BAD_VACANCY     = 1920,   /* Bad value in ONF_ET_SET_VACANCY. */
};
```

If in a `ONF_ET_SET_VACANCY` message the value of `vacancy_down` is greater than the value of `vacancy_up`, the switch should reject the `ONF_ET_SET_VACANCY` message and send an `ofp_error_msg` with `ONFERR_ET_BAD_VACANCY` code.

The error `ONFERR_ET_BAD_VACANCY` uses the following structure:

```
/* Message structure for all errors. */
struct onf_error_msg {
  struct ofp_header header;
  uint16_t type;            /* OFPET_EXPERIMENTER. */
  uint16_t exp_code;        /* 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_code` field is set to `ONFERR_ET_BAD_VACANCY`.