Adding a new OVS action

Numan Siddique - @numansiddique

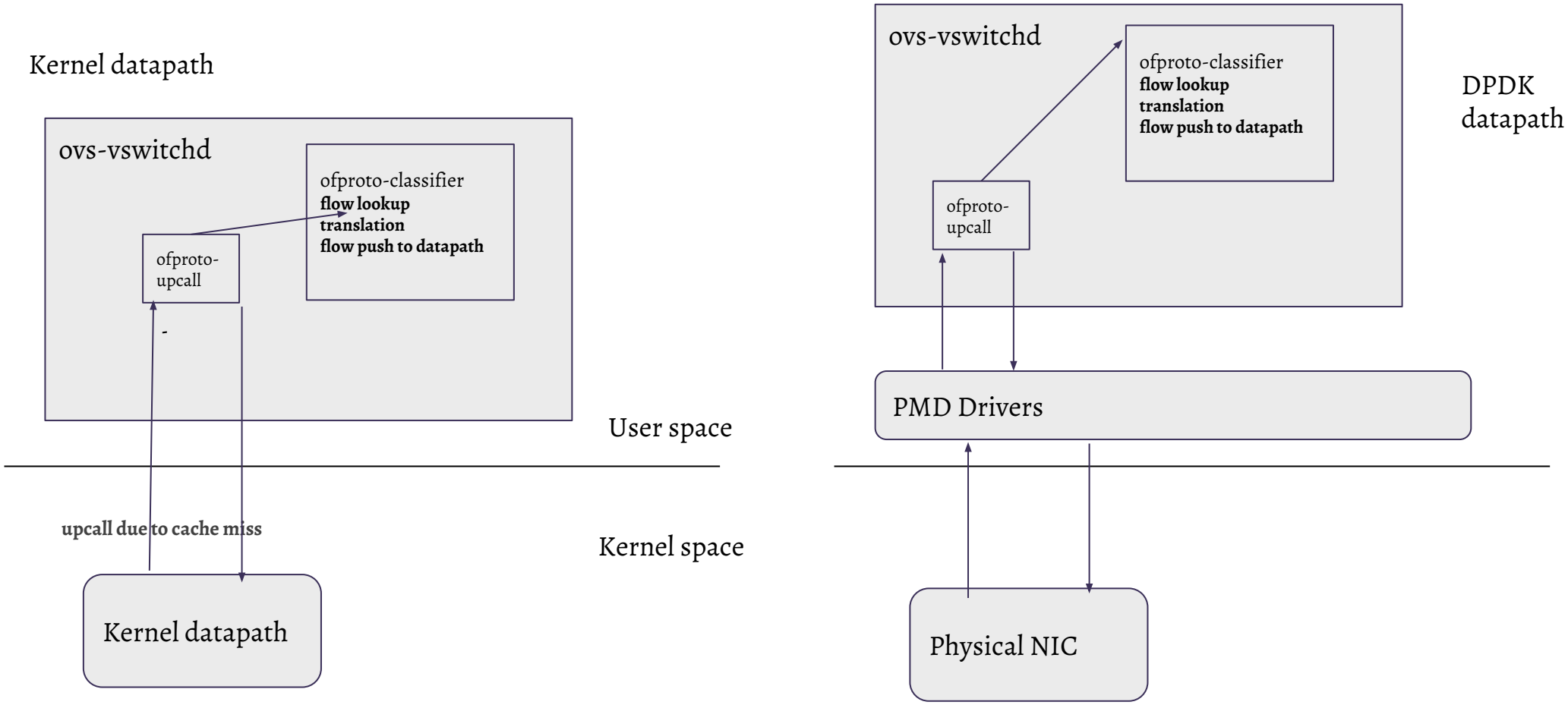
Yogananth Subramanian <ysubrama@redhat.com>

We will talk about

DPDK
DATA PLANE DEVELOPMENT KIT

- Basics of how a packet is handled in OVS (kernel datapath and dpdk)
- How to add a new action in OVS
- What code changes are required
- Will take an example

Packet flow in OVS



What is an action

- An OpenFlow flow has 2 parts
  - match
  - action

- Eg.
  - table=0, priority=100,icmp,in_port=1 actions=drop

  - table=15, priority=100,ip, nw_dst=192.168.0.0/24 actions=dec_ttl(),move:NXM_OF_IP_DST[]->NXM_NX_XXREG0[96..127],mod_dl_src:00:00:00:00:ff:01,load:0x1->NXM_NX_REG15[],resubmit(,16)

Existing OVS actions

OVS actions	Datapath actions
Output	
output	`OVS_ACTION_ATTR_OUTPUT`
controller	`OVS_ACTION_ATTR_USERSPACE`
bundle	
Header changes	
set_field	
mod_dl_src	`OVS_ACTION_ATTR_SET`
mod_dl_dst	
mod_nw_src	
mod_nw_dst	
mod_tp_src	
mod_tp_dst	
…	
conntrack	
ct	`OVS_ACTION_ATTR_CT`
ct_clear	`OVS_ACTION_ATTR_CT_CLEAR`

Why you want to add a new OVS action

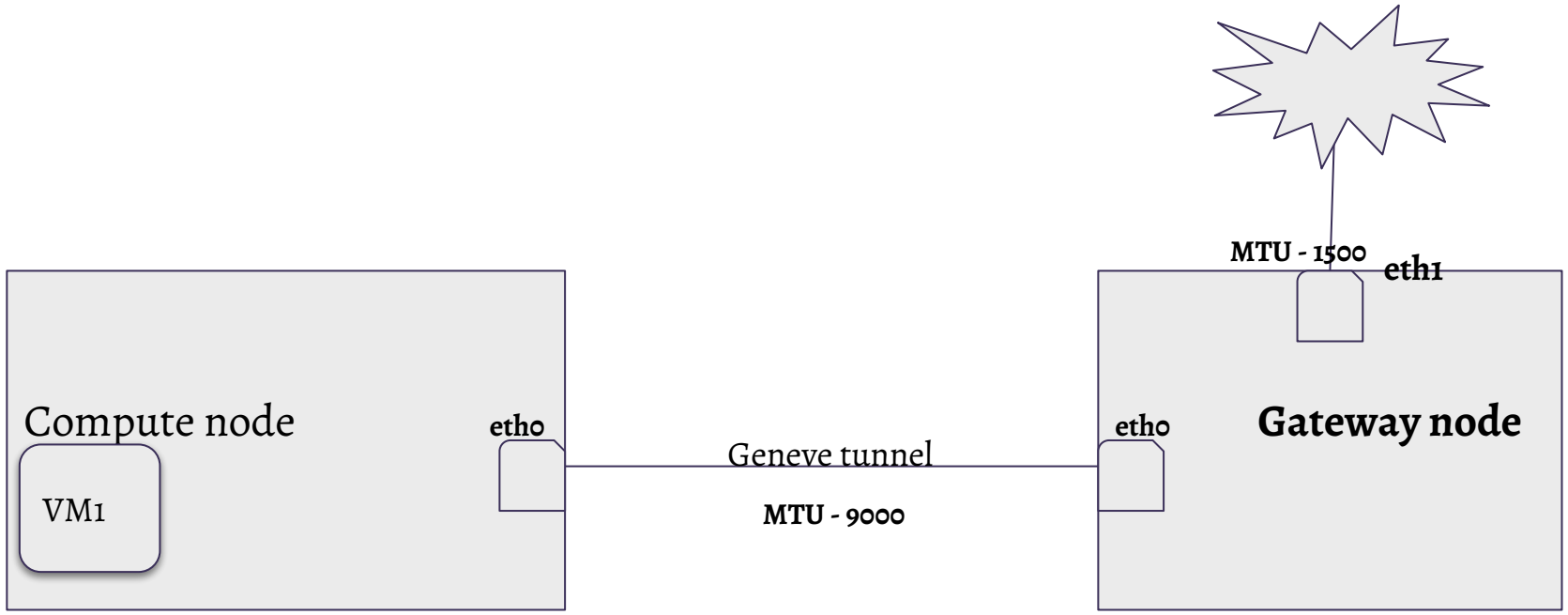
DPDK
DATA PLANE DEVELOPMENT KIT

- You have an use case and existing OVS actions can't address it.

Note:

- We are not talking about OpenFlow actions defined in the OpenFlow spec.
- We are talking about custom actions in OVS.

New OVS action - ‘check_pkt_larger’



- Action to check the packet length - “check_pkt_larger”
- Depending on the packet length take some action.
- Usage: ***check_pkt_larger(1500):-> NXM_NX_REG0[0]***
- In the case of OVN, if the packet length is greater then MTU of eth1 - generate ICMP packet to fragment the packet

Note: The proposed action is submitted as RFC and is still under review

Steps required

DPDK
DATA PLANE DEVELOPMENT KIT

- Check with OVS community and discuss first.
- Add a new action code in ofp-actions.c
- Add parse/encode/decode action code - so that ovs-ofctl, controllers can use your new action
- Implement your new action in Kernel datapath if required.
- Add a user space implementation of your action in ovs-vswitchd
- Action translation to datapath

Before starting

- Discuss with OVS community

How a flow is added

ovs-ofctl add-flow br-int "table=0,priority=100, ip actions=mod_nw_src:10.0.0.4, resubmit(,1)"

1. ovs-ofctl parses the actions and encode into ofpacts
2. It then encodes from ofpacts to OpenFlow actions
3. Sends a flow_mod message to ovs-vswitchd

ovs-vswitchd then

1. decodes the OpenFlow actions into ofpacts
2. Adds the flow into the classifier rule of the switch

- ofpacts is the generic representation of an action in OVS (independent of OpenFlow version)

Add a new action code in ofp-actions.c

How the action will be used by user

check_pkt_larger(length)->REG_BIT

Eg. **check_pkt_larger(1500)->NXM_NX_REG0[0]**

include/openvswitch/ofp-actions.h

```
/* List of OVS abstracted actions.
*
* This macro is used directly only internally by this header, but the list is
* still of interest to developers.
...
#define OFPACTS                                                    \
..
...
OFPACT(CLONE,           ofpact_nest,        actions, "clone")      \
OFPACT(CHECK_PKT_LARGER, ofpact_check_pkt_larger, ofpact,          \
       "check_pkt_larger")                                         \
```

```
/* OFPACT_CHECK_PKT_LARGER.
*
* Used for
NXAST_CHECK_PKT_LARGER. */
struct ofpact_check_pkt_larger
{
    struct ofpact ofpact;
    uint16_t pkt_len;
    struct mf_subfield dst;
};
```

Contd.

DPDK
DATA PLANE DEVELOPMENT KIT

lib/ofp-actions.c

```
/* Raw identifiers for OpenFlow actions.

    Decoding and encoding OpenFlow actions across multiple versions is difficult
* to do in a clean, consistent way.  This enumeration lays out all of the
* forms of actions that Open vSwitch supports.
… */
enum ofp_raw_action_type {
/* ## ----------------- ## */
/* ## Standard actions. ## */
/* ## ----------------- ## */
…
…
/* ## ------------------------ ## */
/* ## Nicira extension actions. ## */
/* ## ------------------------ ## */

/* NX1.0+(49): struct nx_action_check_pkt_larger, ... VLMFF */
    NXAST_RAW_CHECK_PKT_LARGER,
…
…
}
```

```
/* Check packet length action. */

struct nx_action_check_pkt_larger {
    ovs_be16 type;                /* OFPAT_VENDOR. */
    ovs_be16 len;                 /* 24. */
    ovs_be32 vendor;              /* NX_VENDOR_ID. */
    ovs_be16 subtype;             /* NXAST_OUTPUT_REG. */
    ovs_be16 pkt_len;             /* Length of the packet to
check. */
    ovs_be16 ofs_nbits;           /* (ofs << 6) | (n_bits -
1). */
    /* Followed by:
     * - 'dst', as an OXM/NXM header (either 4 or 8
bytes).
     * - Enough 0-bytes to pad the action out to 24 bytes.
*/
    uint8_t pad[10];
};
```

Parsing/Formatting the action

- Add code in ofp-actions.c to parse/format the action

ovs-ofctl add-flow br-int table=0,icmp actions=check_pkt_larger(1500)->NXM_NX_REG0[0]

```
static char * OVS_WARN_UNUSED_RESULT
parse_CHECK_PKT_LARGER(char *arg, const struct ofpact_parse_params  *pp)
{
  ...
   struct ofpact_check_pkt_larger  *check_pkt_larger  =
       ofpact_put_CHECK_PKT_LARGER(pp->ofpacts);
   error = str_to_u16(value, NULL, &check_pkt_larger ->pkt_len);
   if (error) {
       return error;
   }
   check_pkt_larger ->dst = dst;
}

static void
format_CHECK_PKT_LARGER(const struct ofpact_check_pkt_larger  *a,
                        const struct ofpact_format_params  *fp)
{
   ds_put_format(fp->s, "%scheck_pkt_larger( %s%"PRIu32")->",
                 colors .param, colors .end, a->pkt_len);
   mf_format_subfield(&a->dst, fp->s);
}
```

ovn-controller adding a flow with this action

```
...
struct ofpact_check_pkt_larger *check_pkt_larger =
       ofpact_put_CHECK_PKT_LARGER(ofpacts);
check_pkt_larger->pkt_len = 1500;
struct mf_subfield dst;
/* Set dst to proper value */
check_pkt_larger->dst =  dst;
...
```

controller/ovs-ofctl encodes the action using ofpacts

Encoding the ofpacts to OpenFlow actions

- Each ofpact is encoded into OpenFlow action

ofp-actions.c

```
static void
encode_ofpact(const struct ofpact *a, enum ofp_version ofp_version ,
              struct ofpbuf *out)
{
    switch (a->type) {
#define OFPACT(ENUM, STRUCT, MEMBER , NAME)                              \
        case OFPACT_##ENUM:                                              \
            encode_## ENUM(ofpact_get_## ENUM(a), ofp_version , out);    \
            return;
        OFPACTS
#undef OFPACT
    default:
        OVS_NOT_REACHED();
    }
}
```

```
/* Check packet length action. */

struct nx_action_check_pkt_larger {
    ovs_be16 type ;              /* OFPAT_VENDOR. */
    ovs_be16 len ;               /* 24. */
    ovs_be32 vendor ;            /* NX_VENDOR_ID. */
    ovs_be16 subtype ;           /* NXAST_OUTPUT_REG. */
    uint8_t pad[6];
    ovs_be16 pkt_len ;           /* Length of the packet to check. */
    ovs_be16 ofs_nbits ;         /* (ofs << 6) | (n_bits - 1). */
    ovs_be32 dst ;               /* Destination register. */
};
```

```
static void
encode_CHECK_PKT_LARGER(const struct ofpact_check_pkt_larger *check_pkt_larger ,
                        enum ofp_version ofp_version OVS_UNUSED ,
                        struct ofpbuf *out OVS_UNUSED )
{
    struct nx_action_check_pkt_larger *ncpl = put_OFPAT_CHECK_PKT_LARGER(out);
    ncpl->pkt_len = htons(check_pkt_larger->pkt_len);
    ncpl->ofs_nbits = nxm_encode_ofs_nbits(
        check_pkt_larger->dst.ofs, check_pkt_larger->dst.n_bits);
    if (check_pkt_larger->dst.field) {
        ncpl->dst = htonl(nxm_header_from_mff(check_pkt_larger->dst.field));
    }
}
```

Decoding the OpenFlow actions to ofpacts

- Each OpenFlow action is decoded back to ofpact

ofp-actions.c

```
static enum ofperr
decode_OFPAT_RAW_CHECK_PKT_LARGER(
    const struct nx_action_check_pkt_larger  *ncpl,
    enum ofp_version ofp_version OVS_UNUSED , struct ofpbuf *out)
{
    struct ofpact_check_pkt_larger  *check_pkt_larger  =
        ofpact_put_CHECK_PKT_LARGER(out);
    check_pkt_larger ->pkt_len  = ntohs(ncpl->pkt_len);
    check_pkt_larger ->dst.ofs  = nxm_decode_ofs(ncpl->ofs_nbits);
    check_pkt_larger ->dst.n_bits  = nxm_decode_n_bits(ncpl->ofs_nbits);;
    error  = mf_vl_mff_nx_pull_header(&b, vl_mff_map ,
                                      &check_pkt_larger ->dst.field,
                                      NULL, tlv_bitmap );
    return 0;
}

static enum ofperr
ofpacts_decode(const void *actions, size_t actions_len,
               …)
{
   while (openflow.size) {
       if (!error) {
           error = ofpact_decode(action, raw, ofp_version, arg, vl_mff_map,
                                 ofpacts_tlv_bitmap, ofpacts);
       }
      ...
   return 0;
}
```

Autogenerated code in ofp-actions.inc2

```
static enum ofperr
ofpact_decode(const struct ofp_action_header  *a, enum ofp_raw_action_type raw ,
              enum ofp_version version , uint64_t arg,
              const struct vl_mff_map *vl_mff_map ,
              uint64_t  *tlv_bitmap , struct ofpbuf  *out)
{
    switch (raw) {
    …
    …
    case OFPAT_RAW_CHECK_PKT_LARGER:
        return decode_OFPAT_RAW_CHECK_PKT_LARGER(ALIGNED_CAST(const struct
nx_action_check_pkt_larger  *, a), version, out);
    …
    …
}
```

Translating ofpact action- Kernel datapath

- check_pkt_larger OVS action is translated to “check_pkt_len” datapath action

- check_pkt_len(1500, gt(set of actions), less_eq(set of actions))

Eg.

OpenFlows

table=0, priority=100 in_port=1,ip,actions=check_pkt_larger:1500->NXM_NX_REG0[0],resubmit(,1)
table=1, priority=200,in_port=1,ip,reg0=0x1/0x1 actions=output:3
table=1, priority=100,in_port=1,ip,actions=output:4

Translated to datapath action as:

check_pkt_len(1500, gt(output:3), less_eq(output:4))

Translating ofpact action- Kernel datapath

In ofproto-dpif-xlate.c

```
static void
xlate_check_pkt_larger(struct xlate_ctx *ctx,
                       struct ofpact_check_pkt_larger *check_pkt_larger,
                       const struct ofpact *remaining_acts,
                       size_t remaining_acts_len)
{
    union mf_subvalue value;
    memset(&value, 0, sizeof value);
    if (!ctx->xbridge->support.check_pkt_len) {
    …
    …
        return;
    }
    …
    …
    size_t offset =
nl_msg_start_nested(ctx->odp_actions,

OVS_ACTION_ATTR_CHECK_PKT_LEN);
     nl_msg_put_u16(ctx->odp_actions,
OVS_CHECK_PKT_LEN_ATTR_PKT_LEN,
                    check_pkt_larger->pkt_len);
    …
    …
    …
    size_t offset_attr = nl_msg_start_nested(
         ctx->odp_actions, OVS_CHECK_PKT_LEN_ATTR_ACTIONS_IF_GREATER );
     value.u8_val = 1;
    mf_write_subfield_flow(&check_pkt_larger->dst, &value,
&ctx->xin->flow);
    do_xlate_actions(remaining_acts, remaining_acts_len, ctx, true,
false);
    nl_msg_end_nested(ctx->odp_actions, offset_attr);
    …
    …
    offset_attr = nl_msg_start_nested(
         ctx->odp_actions,
OVS_CHECK_PKT_LEN_ATTR_ACTIONS_IF_LESS_EQUAL );
    value.u8_val = 0;
    mf_write_subfield_flow(&check_pkt_larger->dst, &value,
&ctx->xin->flow);
    do_xlate_actions(remaining_acts, remaining_acts_len, ctx, true,
false);

     nl_msg_end_nested(ctx->odp_actions, offset_attr);
    nl_msg_end_nested(ctx->odp_actions, offset);
```

Translating ofpact action- DPDK datapath

DPDK
DATA PLANE DEVELOPMENT KIT

- Add userspace implementation of the action in lib/odp-execute.c

```
static void
odp_execute_check_pkt_len(void *dp, struct dp_packet *packet, bool steal,
                          const struct nlattr *action,
                          odp_execute_cb dp_execute_action)
{
   const struct nlattr *attrs[OVS_CHECK_PKT_LEN_ATTR_MAX  + 1];
  NL_NESTED_FOR_EACH_UNSAFE (a, left, action) {
       int type = nl_attr_type(a);
       …
  }
   is_greater = dp_packet_size(packet) > nl_attr_get_u16(a);
   if (is_greater) {
       a = attrs[OVS_CHECK_PKT_LEN_ATTR_ACTIONS_IF_GREATER ];
   } else {
       a = attrs[OVS_CHECK_PKT_LEN_ATTR_ACTIONS_IF_LESS_EQUAL ];
   }
   if (a) {
       subactions =  nl_attr_get(a);
       subactions_size  = nl_attr_get_size(a);
   }
   …
   odp_execute_actions(dp, &pb, true, subactions, subactions_size,
                       dp_execute_action );
 }
```

How OVN uses it

```
$ovn-nbctl show lr0
router 4dd784e3-94b1-4d6b-aba7-b7c78adde387 (lr0)
    port lrp0
        mac: "00:00:00:00:ff:01"
        networks: ["192.168.0.1/24"]
    port lrp1
        mac: "00:00:00:00:ff:02"
        networks: ["11.0.0.1/24"]
    port lr0-public
        mac: "00:00:20:20:12:13"
        networks: ["172.168.0.100/24"]
        gateway chassis: [chassis-1 chassis-2]
```

$ovn-nbctl set logical_router_port lr0-public **options:gateway_mtu=1500**

```
$ovn-nbctl show
switch cfabcafc-7386-4bab-9fb7-ddf4d414dfdb (public)
    port ln-public
        type: localnet
        addresses: ["unknown"]
    port public-lr0
        type: router
        router-port: lr0-public

switch 16d55b15-38d0-479d-936c-97af36495179 (sw0)
    port lrp0-attachment
        type: router
        addresses: ["00:00:00:00:ff:01"]
        router-port: lrp0
    port sw0-port1
        addresses: ["50:54:00:00:00:01 192.168.0.2"]

switch 6da419c5-2ea4-4e54-b763-516b916b196b (sw1)
    port sw1-port1
        addresses: ["50:54:00:00:00:03 11.0.0.2"]
    port lrp1-attachment
        type: router
        addresses: ["00:00:00:00:ff:02"]
        router-port: lrp1
```

How OVN uses it

$ovn-sbctl dump-flows lro

…

…

…

```
table=9 (lr_in_chk_pkt_len  ), priority=50   , match=(outport == "lr0-public" && ip4),
action=(reg9[2] = check_pkt_larger(1500); next;)
  table=9 (lr_in_chk_pkt_len  ), priority=0    , match=(1), action=(next;)

table=10(lr_in_larger_pkts  ), priority=50   , match=(inport == "lrp0" && outport == "lr0-public"
&& ip4 && reg9[2]),
 action=(icmp4 {icmp4.type = 3; /* Destination Unreachable. */ icmp4.code = 4; /* Frag Needed and
DF was Set. */ icmp4.frag_mtu = 1442; eth.dst = 00:00:00:00:ff:01; ip4.dst = ip4.src; ip4.src =
192.168.0.1; ip.ttl = 255; reg9[1] = 1; next(pipeline=ingress, table=0); };)

table=10(lr_in_larger_pkts  ), priority=50   , match=(inport == "lrp1" && outport == "lr0-public"
&& ip4 && reg9[2]), action=(icmp4 {icmp4.type = 3; /* Destination Unreachable. */ icmp4.code = 4;
/* Frag Needed and DF was Set. */ icmp4.frag_mtu = 1442; eth.dst = 00:00:00:00:ff:02; ip4.dst =
ip4.src; ip4.src = 11.0.0.1; ip.ttl = 255; reg9[1] = 1; next(pipeline=ingress, table=0); };)
  table=10(lr_in_larger_pkts  ), priority=0    , match=(1), action=(next;)
```

Summary

- Discussed how flow translation happens in OVS
- How to add a new action in OVS and steps required

Thank you

DPDK
DATA PLANE DEVELOPMENT KIT

Questions ?

Numan Siddique
twitter: @numansiddique
irc: #numans

Yogananth Subramanian
<ysubrama@redhat.com>