# Open vSwitch's
# Extensible Flow Match (NXM)

Justin Pettit

Nicira Networks

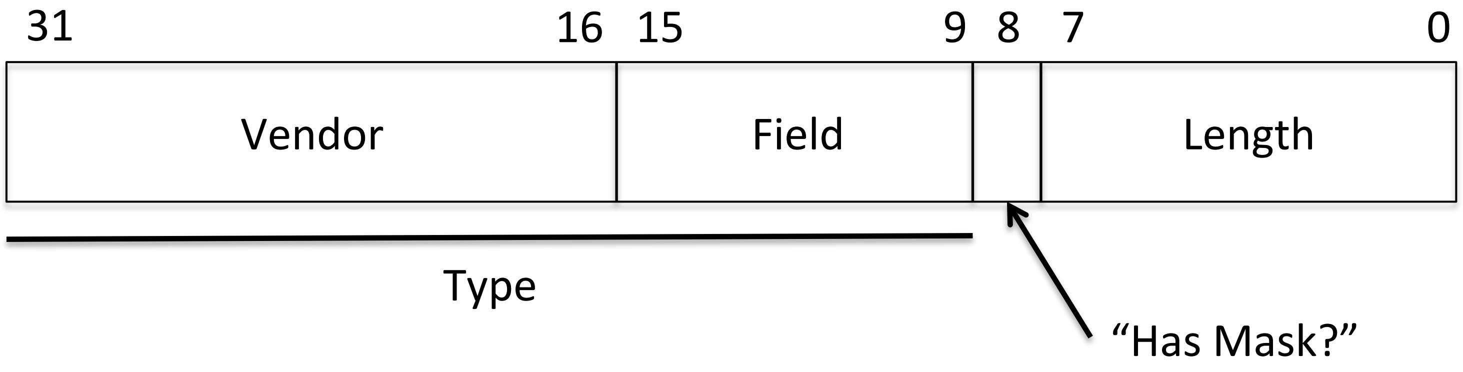
15 August 2011

# Overview

- Break dependence on ofp_match, so new matches can be defined without changing the wire protocol
- Defined new TLV to be compact
- Built on top of OpenFlow 1.0
- Introduced in Open vSwitch 1.1.0

# TLV Structure

- Variable length: 5 to 259 bytes long
- Not aligned or padded
- First four bytes are “header”, followed by “body”
- “Vendor” and “Field” define a “Type”



31 16 15 9 8 7 0

Vendor

Field

Length

Type

“Has Mask?”

# Semantics

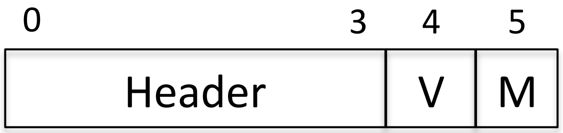
- Any field not specified is implicitly wildcarded
- Prerequisites may be defined that must be met (eg, NXM_OF_IP_TOS may only be matched if NXM_OF_ETH_TYPE==0x0800)
- Entries with prerequisites must appear after the prerequisite entries
- A given "type" must only appear once in a match

# Bit-Level Wildcarding

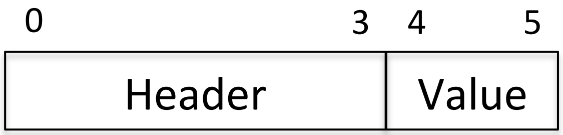
- The “Has Mask?” bit indicates whether the definition includes a mask
- A mask doubles the length of the “body”
- An unset bit in mask indicates that the bit is wildcarded (opposite of “wildcards” in ofp_match)
- Not all fields are maskable (eg, ingress port) and some support limited masking (eg, IPv4 CIDR masks)

# Entry Examples

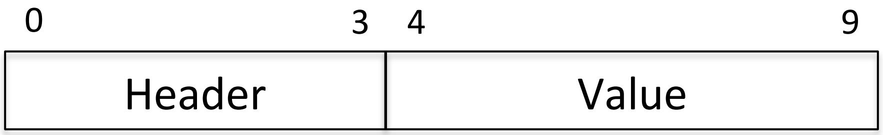
- 8-bit value, hasmask=1, length=2



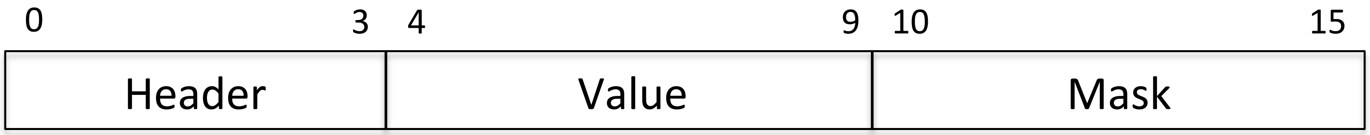
- 16-bit value, hasmask=0, length=2



- 48-bit value, hasmask=0, length=6



- 48-bit value, hasmask=1, length=12



# Definition Examples

Ethernet

```
/* Packet's Ethernet type.
 *
 * For an Ethernet II packet this is taken from the Ethernet header.  For an
 * 802.2 LLC+SNAP header with OUI 00-00-00 this is taken from the SNAP header.
 * A packet that has neither format has value 0x05ff
 * (OFP_DL_TYPE_NOT_ETH_TYPE).
 *
 * For a packet with an 802.1Q header, this is the type of the encapsulated
 * frame.
 *
 * Prereqs: None.
 *
 * Format: 16-bit integer in network byte order.
 *
 * Masking: Not maskable. */
#define NXM_OF_ETH_TYPE   NXM_HEADER  (0x0000,  3, 2)
```

IPv4

```
/* The source or destination address in the IP header.
 *
 * Prereqs: NXM_OF_ETH_TYPE must match 0x0800 exactly.
 *
 * Format: 32-bit integer in network byte order.
 *
 * Masking: Only CIDR masks are allowed, that is, masks that consist of N
 *   high-order bits set to 1 and the other 32-N bits set to 0. */
#define NXM_OF_IP_SRC     NXM_HEADER  (0x0000,  7, 4)
#define NXM_OF_IP_SRC_W   NXM_HEADER_W(0x0000,  7, 4)
#define NXM_OF_IP_DST     NXM_HEADER  (0x0000,  8, 4)
#define NXM_OF_IP_DST_W   NXM_HEADER_W(0x0000,  8, 4)
```

# Example Flows

- Match TCP port 80 traffic to 192.168.1.0/24:
  - NXM_OF_ETH_TYPE(0x0800)
  - NXM_OF_IP_PROTO(6)
  - NXM_OF_IP_DST_W(0xc0a80100, 0xffffff00)
  - NXM_OF_TCP_DST(80)
- Match traffic coming in port 3 with a particular IPv6 source address:
  - NXM_OF_IN_PORT(3)
  - NXM_OF_ETH_TYPE(0x86dd)
  - NXM_NX_IPV6_SRC(0xfe80…20c29fffec7374d)

# Changes to OpenFlow Messages

- Match moved to end of message
- New “match_len” field
- Messages changed:
  - Flow Mod
  - Flow Removed
  - Flow Stats Request/Response
  - Aggregate Stats Request/Response

# “Flow Removed” Example

Identical to original other than match description moved to the end and new “match_len” field

OpenFlow

```
struct ofp_flow_removed {
    struct ofp_header header;
    struct ofp_match match;   /* Description of fields. */
    ovs_be64 cookie;          /* Opaque controller-issued identifier. */

    ovs_be16 priority;        /* Priority level of flow entry. */
    uint8_t reason;           /* One of OFPRR_*. */
    uint8_t pad[1];           /* Align to 32-bits. */

    ovs_be32 duration_sec;    /* Time flow was alive in seconds. */
    ovs_be32 duration_nsec;   /* Time flow was alive in nanoseconds beyond
                                 duration_sec. */
    ovs_be16 idle_timeout;    /* Idle timeout from original flow mod. */
    uint8_t pad2[2];          /* Align to 64-bits. */
    ovs_be64 packet_count;
    ovs_be64 byte_count;
};
```

NXM

```
struct nx_flow_removed {
    struct nicira_header nxh;
    ovs_be64 cookie;          /* Opaque controller-issued identifier. */
    ovs_be16 priority;        /* Priority level of flow entry. */
    uint8_t reason;           /* One of OFPRR_*. */
    uint8_t pad[1];           /* Align to 32-bits. */
    ovs_be32 duration_sec;    /* Time flow was alive in seconds. */
    ovs_be32 duration_nsec;   /* Time flow was alive in nanoseconds beyond
                                 duration_sec. */
    ovs_be16 idle_timeout;    /* Idle timeout from original flow mod. */
    ovs_be16 match_len;       /* Size of nx_match. */
    ovs_be64 packet_count;
    ovs_be64 byte_count;
    /* Followed by:
     *   - Exactly match_len (possibly 0) bytes containing the nx_match, then
     *   - Exactly (match_len + 7)/8*8 - match_len (between 0 and 7) bytes of
     *     all-zero bytes. */
};
```

# Adding IPv6 Support

- NXM support committed Nov 10, 2010
- IPv6 support committed Feb 2, 2011
- No changes to underlying protocol—only seven new NXM fields

# Current OVS Match Extensions

- Metadata registers
- Tunnel ID (eg, GRE key)
- ARP target and source hardware addresses
- IPv6 source and destination addresses
- ICMPv6 type and code
- IPv6 neighbor discovery addresses (similar to IPv4 ARP)

# Conclusion

- New matches do not require modifying the wire protocol
- Fully supports OpenFlow 1.0 features
- Used in multiple controller products and many production environments