

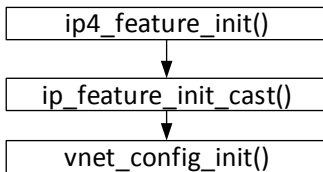
IP Features

Michael
2017

IP Features

VPP16.09

- input feature
- post-rewrite feature

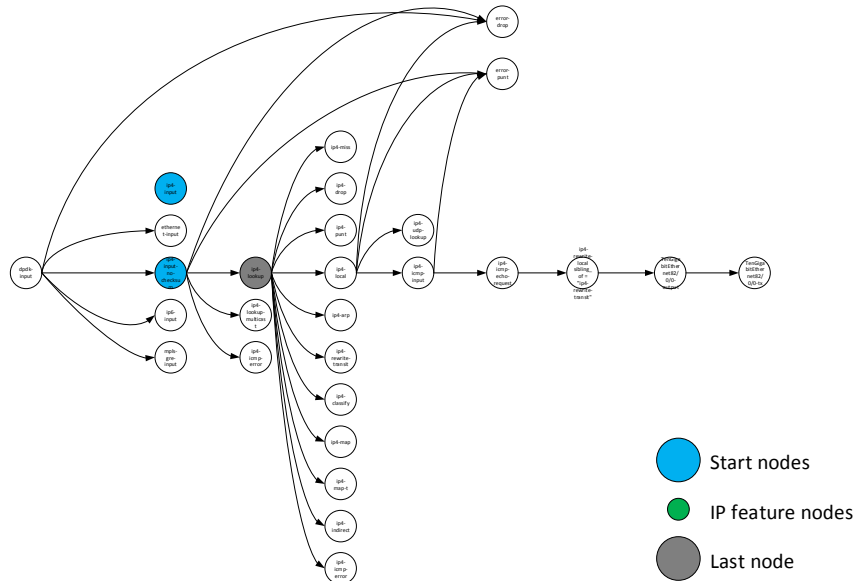


```
VNET_IP4_UNICAST_FEATURE_INIT()
VNET_IP4_MULTICAST_FEATURE_INIT()
VNET_IP6_UNICAST_FEATURE_INIT()
VNET_IP6_MULTICAST_FEATURE_INIT()
```

```
DBGvpp# show ip features
Available IP feature nodes
ip4 unicast:
  ip4-inacl
  ip4-source-check-via-rx
  ip4-source-check-via-any
  ip4-source-and-port-range-check
  ip4-policer-classify
  ipsec-input-ip4
  vpath-input-ip4
  ip4-lookup
ip4 multicast:
  vpath-input-ip4
  ip4-lookup-multicast
ip6 unicast:
  ip6-inacl
  ip6-policer-classify
  ipsec-input-ip6
  l2tp-decap
  vpath-input-ip6
  ip6-lookup
ip6 multicast:
  vpath-input-ip6
  ip6-lookup
```

Feature Arc

VPP16.09



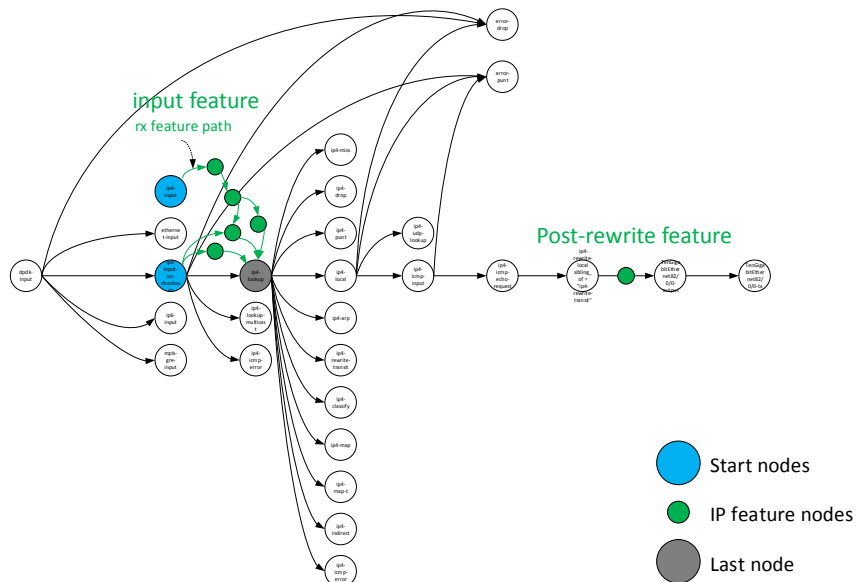
Copyright© 2017. All rights reserved.

Michael

3

Feature Arc(Cont')

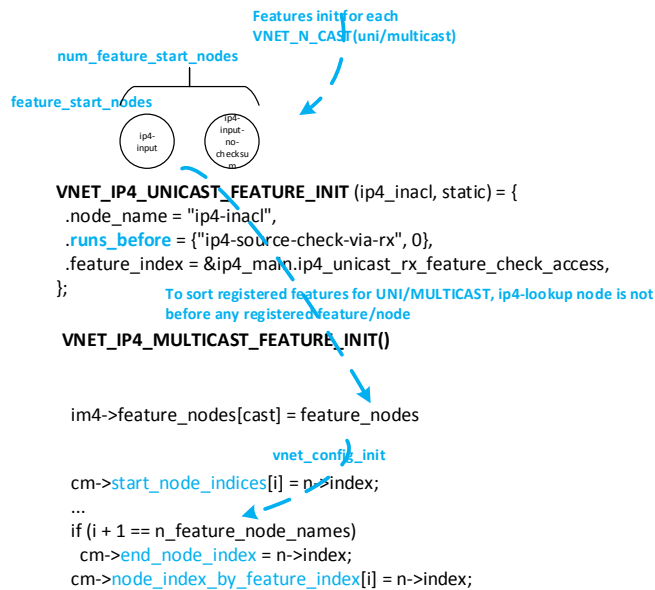
VPP16.09



Copyright© 2017. All rights reserved.

Michael

4



```

#set interface input acl intfc <int> [ip4-table <index>] [ip6-table <index>] [l2-table <index>] [del]
#show inacl type [ip4|ip6|l2]
#set interface acl output <interface> [disable]
  
```

```

VLIB_REGISTER_NODE (ip4_inacl_node) = {
    .function = ip4_inacl,
    .name = "ip4-inacl",
    .vector_size = sizeof (u32),
    .format_trace = format_ip_inacl_trace,
    .n_errors = ARRAY_LEN(ip_inacl_error_strings),
    .error_strings = ip_inacl_error_strings,

    .n_next_nodes = ACL_NEXT_INDEX_N_NEXT,
    .next_nodes = {
        [ACL_NEXT_INDEX_DENY] = "error-drop",
    },
};

VNET_IP4_UNICAST_FEATURE_INIT (ip4_inacl, static) = {
    .node_name = "ip4-inacl",
    .runs_before = {"ip4-source-check-via-rx", 0},
    .feature_index = &ip4_main.ip4_unicast_rx_feature_check_access,
};
  
```

ip_inacl_inline()

```

sw_if_index0 = vnet_buffer(b0)->sw_if_index[VLIB_RX];
tid = INPUT_ACL_TABLE_IP4;
table_index0 = am-
>classify_table_index_by_sw_if_index[tid][sw_if_index0];

t0 = pool_elt_at_index(vcm->tables, table_index0);
vnet_buffer(b0)->l2_classify.hash =
vnet_classify_hash_packet(t0, (u8 *) h0);

e0 = vnet_classify_find_entry(t0, (u8 *) h0, hash0, now);
If entry found
next0 = (e0->next_index < n_next_nodes)? e0->next_index:next0;

Put/forward the packet into next
frame/node as the entry indicated,
otherwise put/forward it as default

```

>>>

pick a feature arc and add a feature which classifies and either:

- sets `vnet_buffer(b0)->sw_if_index[VLIB_TX] = tap_port_sw_if_index`, rewinds the buffer, and enqueues it to 'interface-tx', or
- sends the packet to the next feature in the feature arc.
- is really simple; just enqueue the packet as directed by: `vnet_feature_next(sw_if_index0, &next0, b0);`

<<<