Connection Tracking

Usage and P4 mapping for PNA

Anjali Singhai Jain (Intel)

Namrata Limaye (Intel)

Hari Thantry (Google)

Nishant Bhat (Intel)

Venkata Suresh Kumar P (Intel)
Balachandher Sambasivam (Intel)

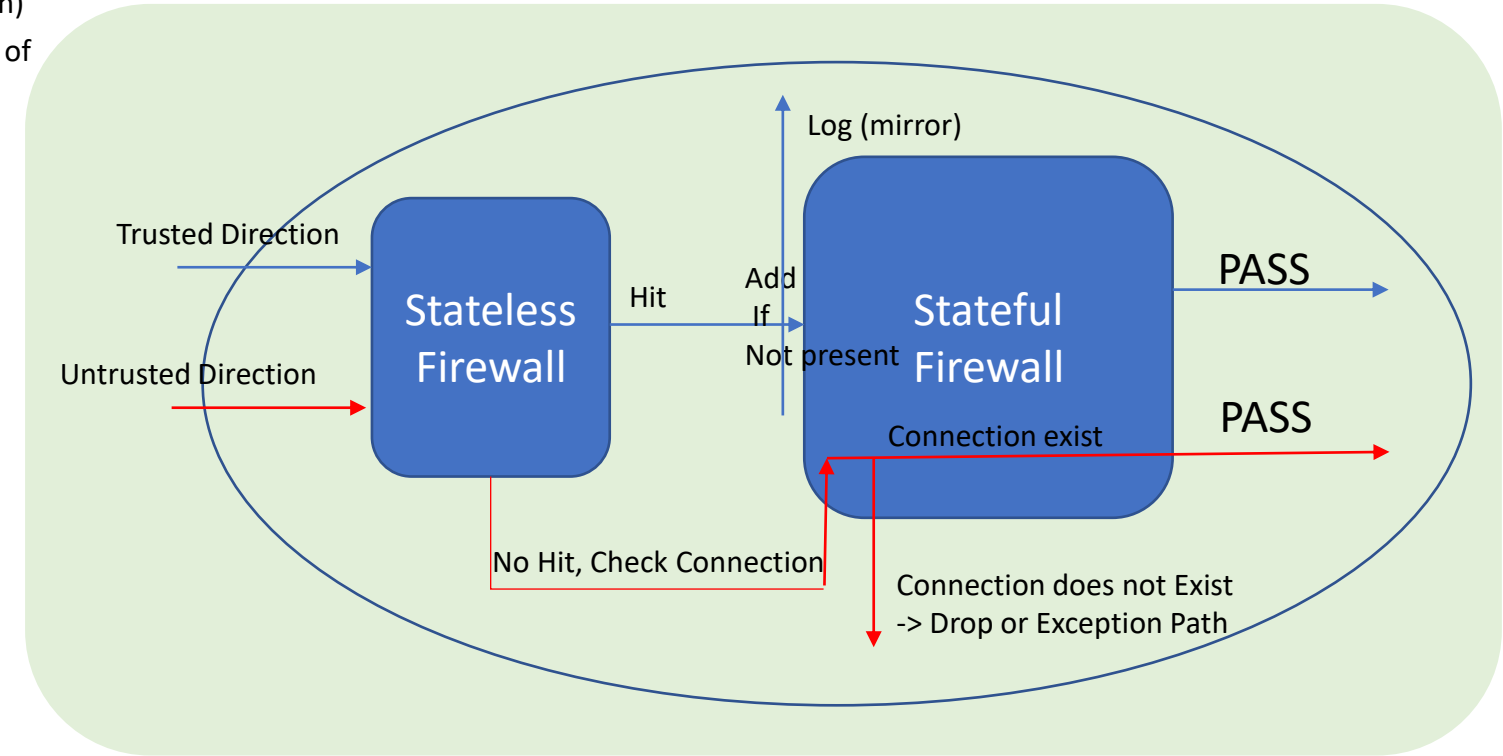
Deb Chatterjee (Intel)

What is CT and its Usage?

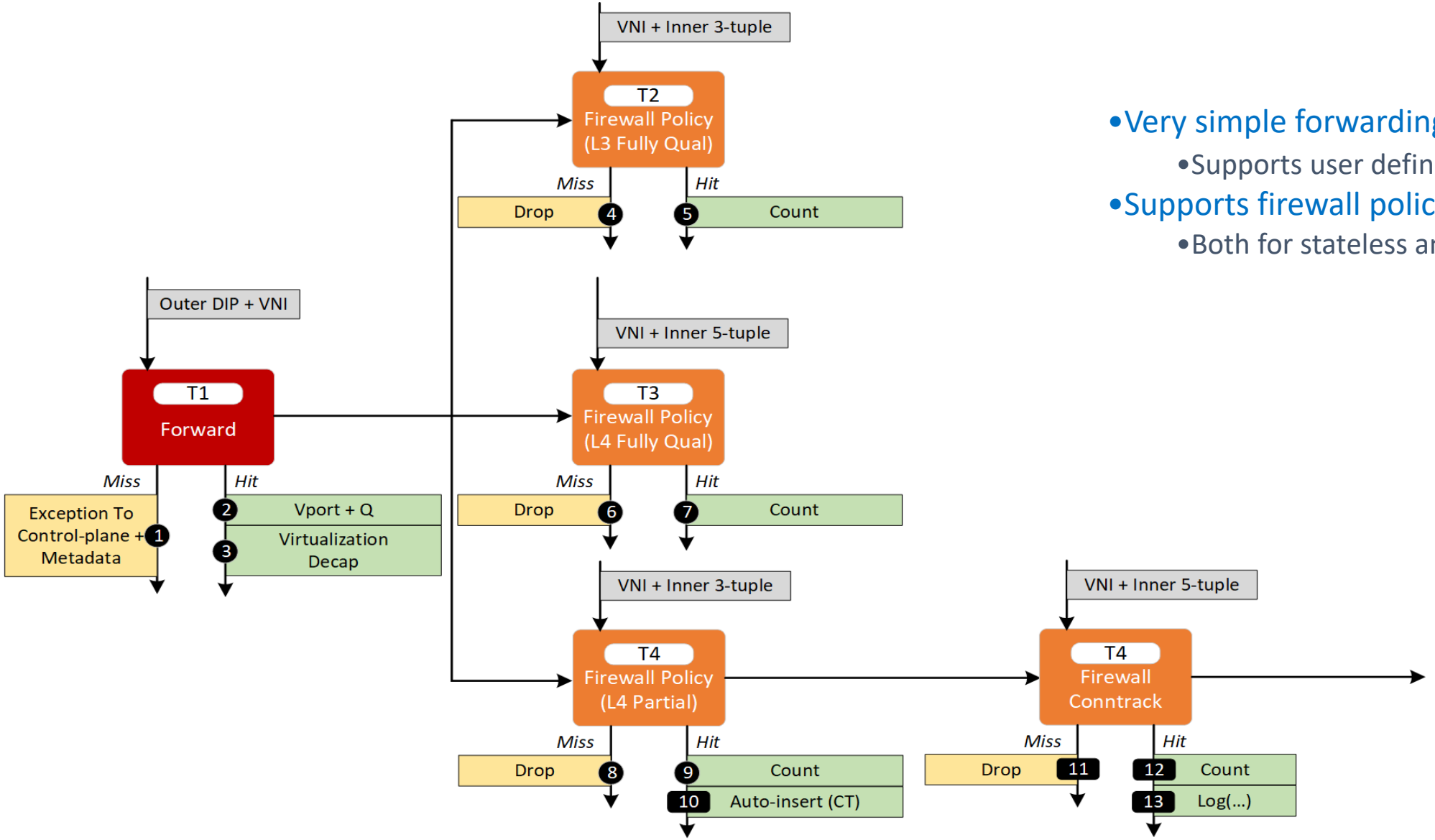
Usage:

- Stateful Firewalls
  - Stateless Firewall : Match = Remote IP, IP Protocol, Dest L4 port)
  - Stateful Firewall : Match = 5 tuple + CT Zone (Unique connection)
  - Packet Permitted **if permitted by Stateless firewall** rule OR part of existing connection.
- Firewall Logging
- Access Control for pay-per-use.
- SYN flood attack detection
- TCP Sequence Number Window checking

Connection tracking is a feature in a data plane to keep track of individual flows and allow for stateful operation on the flows.



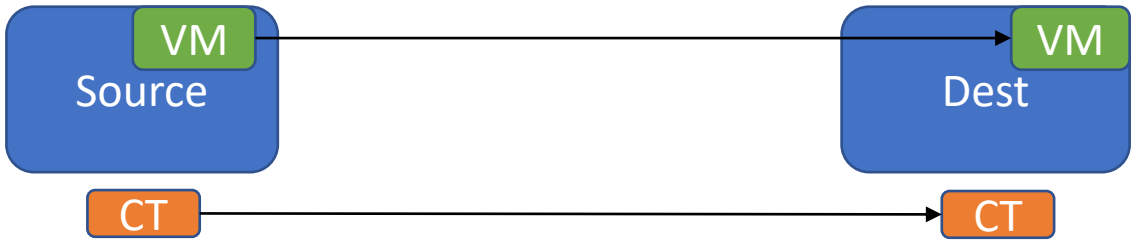
Virtual Forwarding + Stateful Firewall Example



- Very simple forwarding example
  - Supports user defined tunnels
- Supports firewall policy
  - Both for stateless and stateful rules

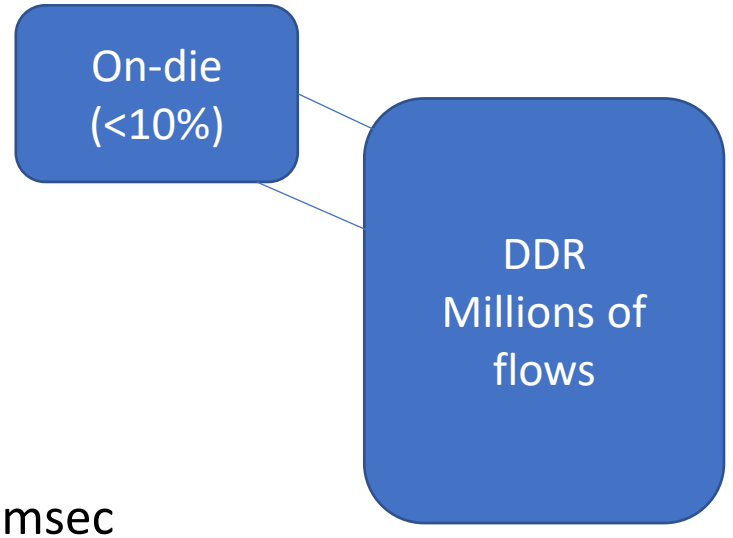
CT HW State retrieval/restore by CP SW: Black hole avoidance (Live Migration)

1. Support for Retrieving the Connection Tracker(CT) table entries from HW and finding the Migrating Virtual Machine(VM) related flows in Brown out phase.
2. Pre-install those on to Destination CT table, before the VM starts on Destination Machine.
3. Avoids black hole effect for most of the flows for a migrated VM.
4. Lazy deletion by SW of the CT flow rules related to migrated VM on the Source Machine.
   - (Disable the vPort first, and then cleanup the CT rules before reusing the vPort.

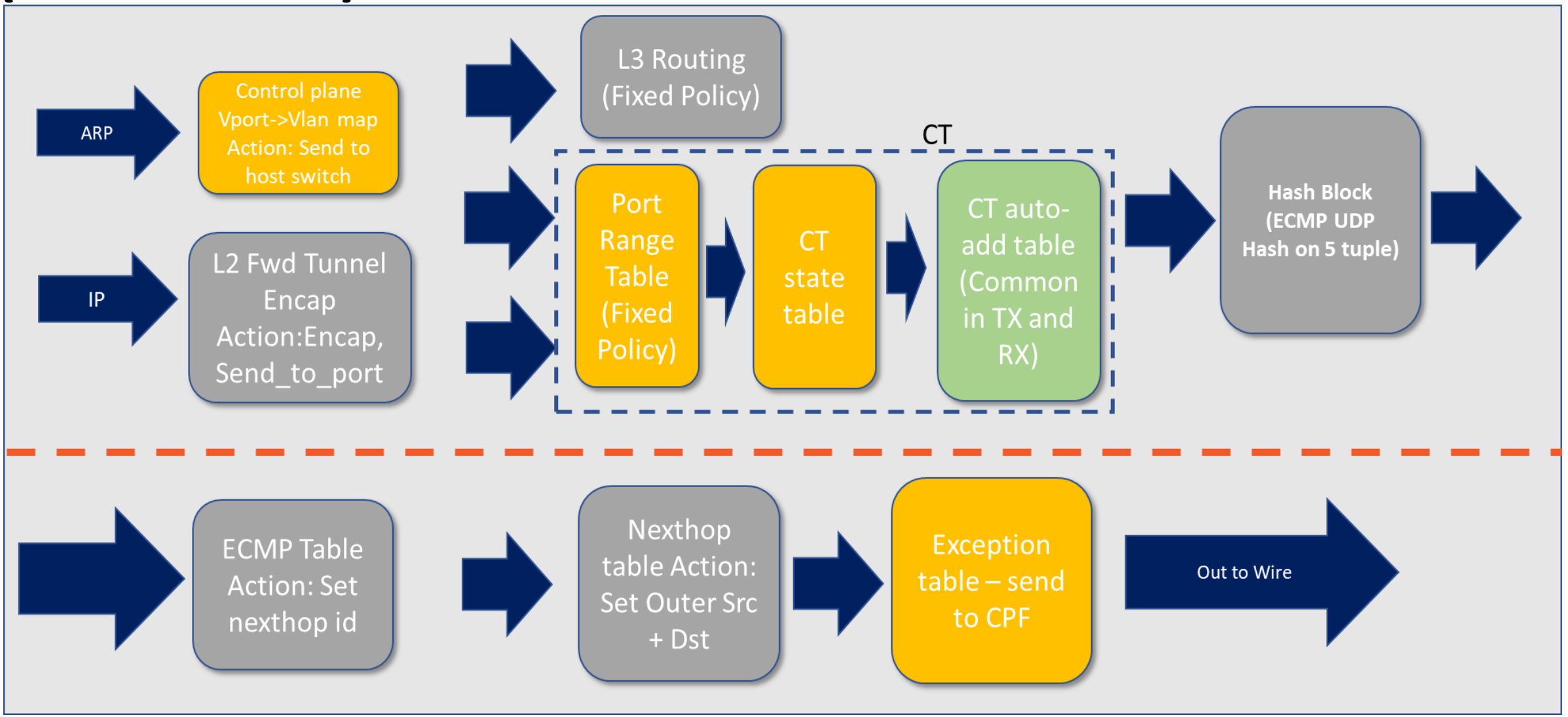


Scale of flows & State based Aging per flow

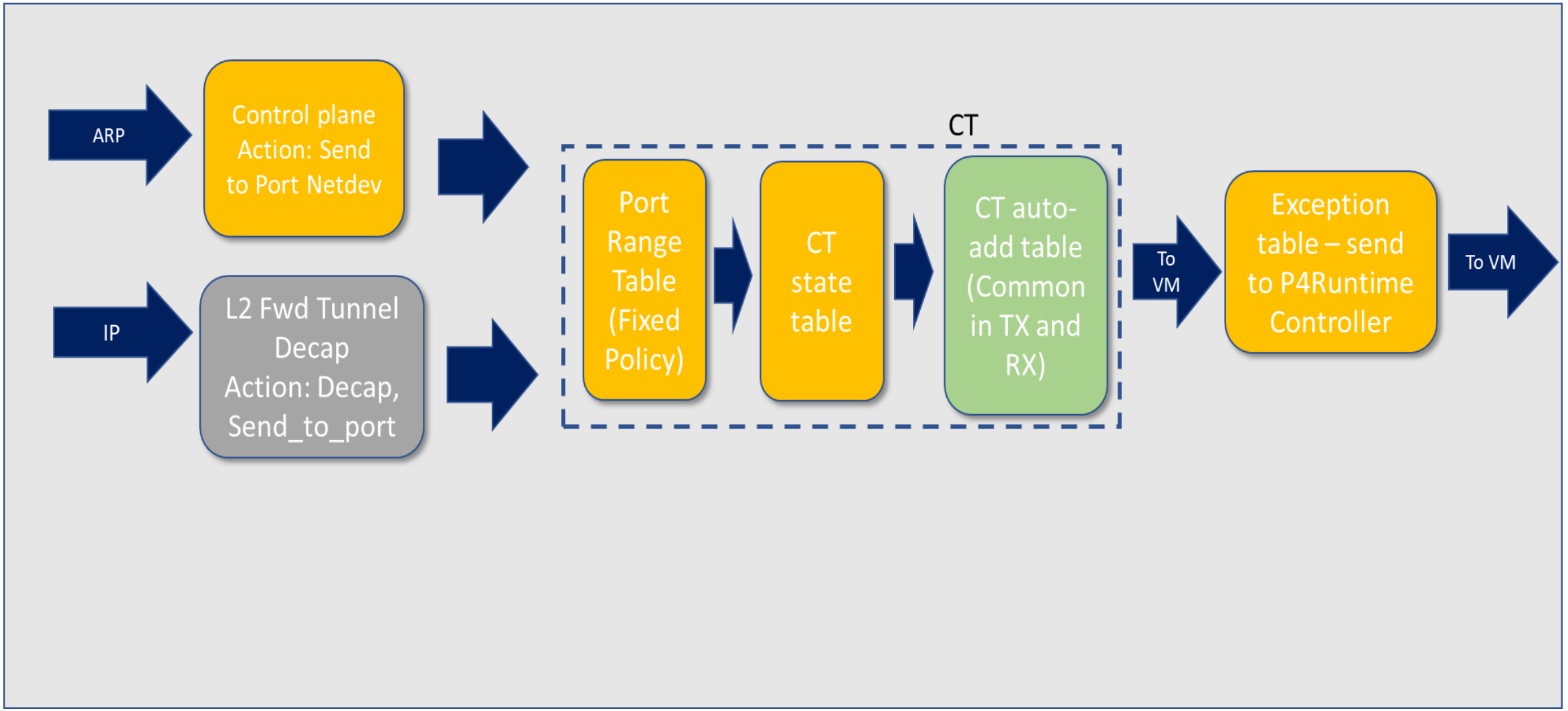
- Intel IPU ConnTrack State Table Scale (Millions of flows in L2 cache, <10% on-die L1 cache, LRU eviction)
- 4 value Aging Timer Select
  - No Aging
  - Getting Established (msecs to seconds) (State 1)
    - Absolute Mode
  - Established (10 mins… to hrs) (State 2)
    - Replenish mode
  - Tear down (msecs to seconds) (State 3)
    - Absolute Mode
- Example 1: @10M/sec Flow add rate. = 12.5% of stale flows in 200msec
- Example 2: @3M/sec Flow add rate = 3.75% of stale flows in 200msec
- Work in progress: Immediate removal flow option in P4 DP using Seq Number match



Infrastructure recipe pipeline – Egress (VM to wire)



Infrastructure recipe pipeline – Ingress (wire to VM)



Connection tracking P4 tables – CT State table

```
/* The basic firewall handles the following:
   - Firewall policy: permit / deny
   - Range based comparison for permit & deny
   - Connected state driven permit with auto-insertion - shown here
*/
action tcp_syn_packet () {
    meta.add_on_miss = true /* Insert into CT Table */
    meta.timer = EXPIRE_TIME_PROFILE_TCP_NEW;
}
action tcp_fin_or_rst_packet () {
    meta.timer = EXPIRE_TIME_PROFILE_TCP_NOW;
}
action tcp_other_packets () {
    meta.timer = EXPIRE_TIME_PROFILE_TCP_ESTABLISHED;
}

/* CT State transition table */
table set_ct_options {
    key = {
        meta.direction: exact;
        hdr.tcp.flags: exact;
    }
    actions = {
        tcp_syn_packet;
        tcp_fin_or_rst_packet;
        tcp_other_packets;
    }
    const default_action = tcp_other_packets;
}
```

Connection tracking P4 tables – CT TCP Table

```
table ct_tcp_table {
    /* add_on_miss table */
    key = {
        /* Pick fields based on direction */
        SelectByDirection(PNA_Direction_t.NET_TO_HOST,
                          hdr.ipv4.srcAddr,
                          hdr.ipv4.dstAddr):
                          exact @name("ipv4_addr_0");
        SelectByDirection(PNA_Direction_t.HOST_TO_NET,
                          hdr.ipv4.dstAddr,
                          hdr.ipv4.srcAddr):
                          exact @name("ipv4_addr_1");
        hdr.ipv4.protocol : exact;
        SelectByDirection(PNA_Direction_t.NET_TO_HOST,
                          hdr.tcp.srcPort,
                          hdr.tcp.dstPort):
                          exact @name("tcp_port_0");
        SelectByDirection(PNA_Direction_t.HOST_TO_NET,
                          hdr.tcp.dstPort,
                          hdr.tcp.srcPort):
                          exact @name("tcp_port_1");
      }
      actions = {
          @tableonly   ct_tcp_table_hit;
          @defaultonly ct_tcp_table_miss;
      }
      add_on_miss = true; /* PNA Table Property */
      idle_timeout_with_auto_delete = true; /* PNA */
      const default_action = ct_tcp_table_miss;
    }
```

```
action ct_tcp_table_miss() {
    if (meta.add_on_miss == true) {
        add_succeeded =
            add_entry(
               action_name = "ct_tcp_table_hit",
               action_params = ct_tcp_hit_params_t){});
    } else {
        drop_packet();
    }
}

action ct_tcp_table_hit () {
    if (user_meta.update_aging_info == 1) {
        if (update_expire_time) {
            set_entry_expire_time(new_expire_time);
            restart_expire_timer();
        } else {
            restart_expire_timer();
        }
    }
}
```

P4 Code Template and website information

1. PNA Connection Tracking info - https://github.com/p4lang/pna/blob/main/examples/pna-example-tcp-connection-tracking.p4

2. P4 Dataplane with CT and rest of Infrastructure Dataplane PNA compliant P4 program will be published in the github shortly here -

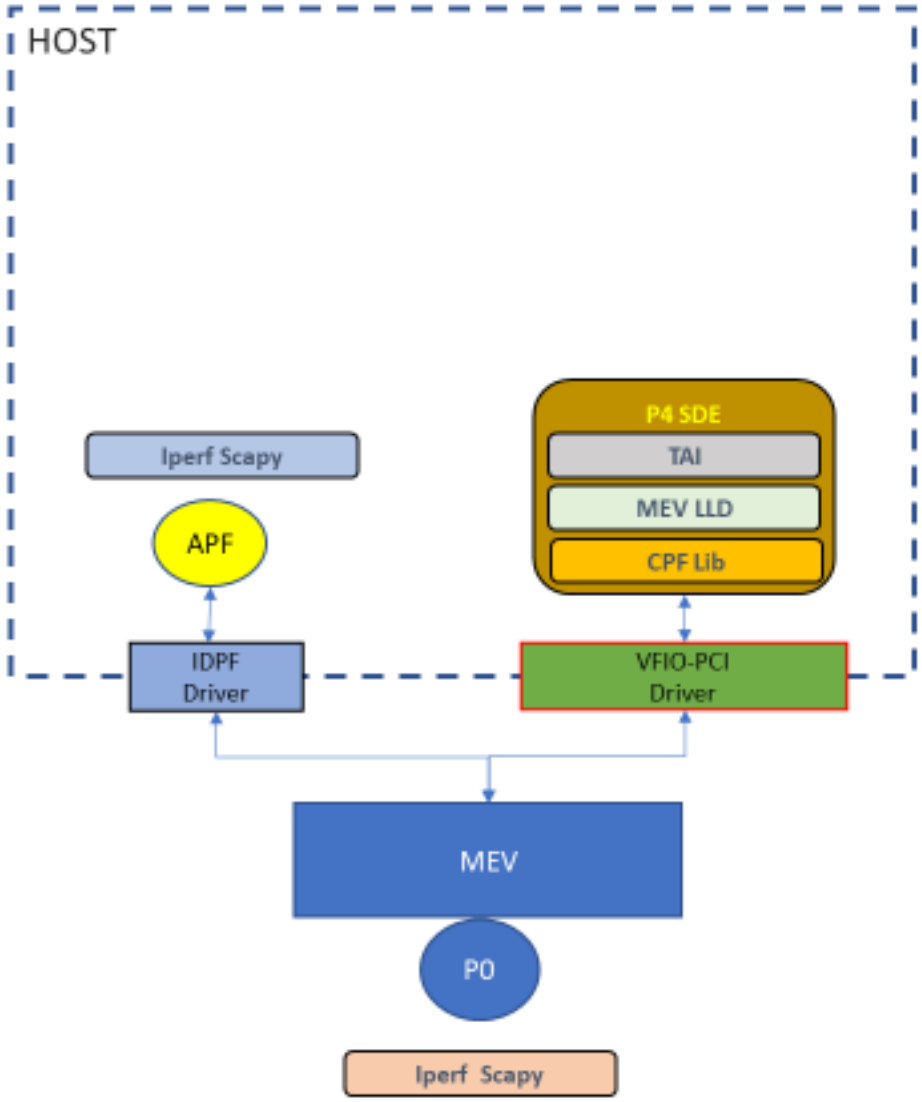
https://github.com/ipdk-io/ipdk/tree/main/build/networking

3. Link for TDI - https://github.com/p4lang/tdi/blob/main/README.md

Future:

Fast closure, immediate removal of flows to be added. New tables will be added to do sequence number match in the final removal state.

CT DEMO


HOST
Iperf Scapy
APF
IDPF
Driver
P4 SDE
TAI
MEV LLD
CPF Lib
VFIO-PCI
Driver
MEV
P0
Iperf Scapy

P4 program Overview

- P4 Program – Mount Evans IPU Connection Tracking. Important tables
- Tables and actions
  - Table - set_ct_options
    - *Match Fields –*
      common.direction, tcp.flags
    - *Actions –*
      tcp_syn_packet, tcp_fin_or_rst_packet, tcp_other_packets
  - Table - ct_tcp_table
    - *Match Fields –*
      Five Tuple : dstAddr, srcAddr, protocol, dstPort, srcPort
    - *Actions –*
      ct_tcp_table_miss, ct_tcp_table_hit

Scenario

- P4 Program – Mount Evans IPU Connection Tracking
- Using IDPF (LAN Generic Interface) and Physical Ports for traffic
- Standard utilities to verify the traffic
  - Scapy
  - Iperf
  - tcpdump

P4 Flow - Artifacts

- P4 Flow
  - Compiler (p4c)
    - Mount Evans IPU PNA Compiler
  - Target packages
    - Mount Evans IPU Package
  - Control plane
    - tdi.json (Table Driven Interface)
    - P4Info.txt
    - context.json

Rule Configuration

- Table : set_ct_options

bfrt.pna_tcp_connection_tracking_2.main.MainControlImpl.set_ct_options.add_with_tcp_syn_packet(direction=0, flags=0x02)

bfrt.pna_tcp_connection_tracking_2.main.MainControlImpl.set_ct_options.add_with_tcp_fin_or_rst_packet(direction=0, flags=0x01)

bfrt.pna_tcp_connection_tracking_2.main.MainControlImpl.set_ct_options.add_with_tcp_fin_or_rst_packet(direction=0, flags=0x04)

DEMO

Summary

- Intel IPU CT implementation is completely Hands Free based on feedback from Google.
- We are evaluating some more events for our next release this year.
  - During Add/Delete some more ability to do enhances auto-add of actions such a counter, meter, mod etc. for the flow cache in more evolved use cases such as Load Balancer Stateful flow cache.
  - Fast flow removal of tables and pipeline definition to be added.
  - SYN flood detection and mitigation
- Use the PNA model for CT as posted on the PNA github.

Notices and Disclaimers

- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure.
- Your costs and results may vary.
-

Thank You

intel