1. How to map the Ingress Stage of the P4 code to HW to match RMT restrictions

Ans: Here we will discuss only the P4 program for leaf switch. The P4 program for Spine switch is also similar and less complex.

Code Block	Note

`if(hdr.p2p_feedback.isValid()){`
`  standard_metadata.egress_spec = CPU_PORT;`
`  local_metadata.flag_hdr.do_l3_l2 = false;`
`  exit;`
`}else if (hdr.packet_out.isValid()) {`
`  standard_metadata.egress_spec = hdr.packet_out.egress_port;`
`  hdr.packet_out.setInvalid();`
`  exit;`
`}else if (hdr.packet_in.isValid() && IS_RECIRCULATED(standard_metadata)) {`
`  local_metadata.flag_hdr.do_l3_l2 = false;`
`  egress_queue_rate_value_map.write((bit<32>)hdr.packet_in.path_delay_event_port,`
`(bit<48>)local_metadata.egress_rate_event_hdr.egress_traffic_color );`
`  egress_queue_rate_last_update_time_map.write((bit<32>)hdr.packet_in.path_delay_event_port,`
`standard_metadata.ingress_global_timestamp);`
`  mark_to_drop(standard_metadata);`
`}else{`
`  init_pkt();`
`  ingress_delay_processor_control_block.apply(hdr, local_metadata, standard_metadata);`
`  ingress_rate_monitor_control_block.apply(hdr, local_metadata, standard_metadata);`
`}`
 	
`if ((hdr.icmpv6.type == ICMP6_TYPE_NS ) && (hdr.icmpv6.type == ICMP6_TYPE_NS)){`
`  ndp_processing_control_block.apply(hdr, local_metadata, standard_metadata); //This will set the`
`local_metaata.do_l3_l2 field to true if this is a NDP packet`
`  //log_msg("egress spec is {} and egress port is {}",{standard_metadata.egress_spec ,`
`standard_metadata.egress_port});`
`  //TODO we may need to remove the extra headers if other switches forward these packet`
`  exit;`
`}`
	1

`if (local_metadata.flag_hdr.do_l3_l2) {`
`  l2_ternary_processing_control_block.apply(hdr, local_metadata, standard_metadata);`
`  my_station_processing_control_block.apply(hdr, local_metadata, standard_metadata);`
	2

`if (hdr.ipv6.isValid() && local_metadata.flag_hdr.my_station_table_hit) {`
`  downstream_routing_control_clock.apply(hdr, local_metadata, standard_metadata);`
`  if(local_metadata.flag_hdr.downstream_routing_table_hit){`
`  local_metadata.flag_hdr.is_pkt_toward_host = true;`
`  if(hdr.ipv6.hop_limit == 0) { mark_to_drop(standard_metadata); }`
`  }`
`else{`
	3

`  local_metadata.flag_hdr.is_pkt_toward_host = false;`
`  local_metadata.flag_hdr.found_multi_criteria_paths = true;`
`  #ifdef DP_ALGO_ECMP`
`  upstream_ecmp_routing_control_block.apply(hdr, local_metadata, standard_metadata);`
`  #endif`
	4

`  #ifdef DP_ALGO_CP_ASSISTED_POLICY_ROUTING`
`  cp_assisted_multicriteria_upstream_routing_control_block.apply(hdr,`
`local_metadata,standard_metadata);`
`  cp_assisted_multicriteria_upstream_policy_routing_control_block.apply(hdr, local_metadata,`
`standard_metadata);`
`  #endif`
 	
`  }`
`  }`
`}else{`
	5

```
    //log_msg("Unhandled packet in ingress processing");
}
```

Note:

1) All the if-else expression of this code block needs some variable to check. Carefully look at. Except for the following block, variables used in all other if-else block expressions are previously known. So simply we can pass these fields to a TCAM based match action table. This table will be fixed and will only need one TCAM. This saves the use of multiple stages for mapping if-else block. Serves the same purpose using TCAM efficiently.

```
else{
    init_pkt();
    ingress_delay_processor_control_block.apply(hdr, local_metadata, standard_metadata);
    ingress_rate_monitor_control_block.apply(hdr, local_metadata, standard_metadata);
}
```

2) Both of these 2 blocks need 2 TCAM
3) 3 nested if-else 3 stages required. `Downstream_routing_control_clock` is a simple MAT and can be mapped in first stage
4) A simple MAT for ECMP routing is required if ECMP is used
5) If we use the P4TE
    a) cp_assisted_multicriteria_upstream_routing_control_block.apply(hdr, local_metadata,standard_metadata) : -- This is simple 3 MAT that can be matched paralley.
    b) cp_assisted_multicriteria_upstream_policy_routing_control_block.apply(hdr, local_metadata, standard_metadata): This looks like a lot of dangling if-else. But Carefully look all of the variable used in if-else are already available in the metadata before the bloc starts executing. And the number of variables and their bitwidth is pretty small. All of them can be used as exact-match field of TCAM and the corresponding action can be selected using TCAM matching action.

       Basically whenever you have some exact match variables to match in if-else expression you can use them in TCAM as exact match. As the number of if-else logic is always small, they can be easily mapped to hardware

```
lookup_flowlet_id_map();
   if (hdr.ipv6.traffic_class == TRAFFIC_CLASS_LOW_DELAY){
       if (local_metadata.flow_inter_packet_gap > FLOWLET_INTER_PACKET_GAP_THRESHOLD){
            bit<48> low_delay_path_rate_status = 0;
           egress_queue_rate_value_map.read(low_delay_path_rate_status,     (bit<32>)
local_metadata.delay_based_path);
            if(low_delay_path_rate_status == (bit<48>)GREEN ){
               use_low_delay_port();
            }else if(low_delay_path_rate_status == (bit<48>)YELLOW ){
                use_low_egress_queue_rate_port();
            }else if((low_delay_path_rate_status == (bit<48>)RED ) ){ // use safe rate port
                 use_low_egress_queue_depth_port();
            }
            update_flowlet_id_map();
       }else{
           use_old_port();
           update_flowlet_id_map();
       }
   }else if (hdr.ipv6.traffic_class == TRAFFIC_CLASS_HIGH_THROUGHPUT){
```

```
       if (local_metadata.flow_inter_packet_gap > FLOWLET_INTER_PACKET_GAP_THRESHOLD){
            bit<48> low_utilization_path_rate_status = 0;
             egress_queue_rate_value_map.read(low_utilization_path_rate_status,
(bit<32>)local_metadata.egr_queue_based_path);
           if(low_utilization_path_rate_status == (bit<48>)GREEN ){
              use_low_egress_queue_depth_port();
           }else if(low_utilization_path_rate_status == (bit<48>)YELLOW ){
             use_low_egress_queue_rate_port();
           }else if((low_utilization_path_rate_status == (bit<48>)RED ) ){
              use_low_delay_port();
           }
           update_flowlet_id_map();
       }else{
           use_old_port();
           update_flowlet_id_map();
       }
   }else{

            use_low_delay_port(); //for all other traffic try to reduce FCT
   }

}
```

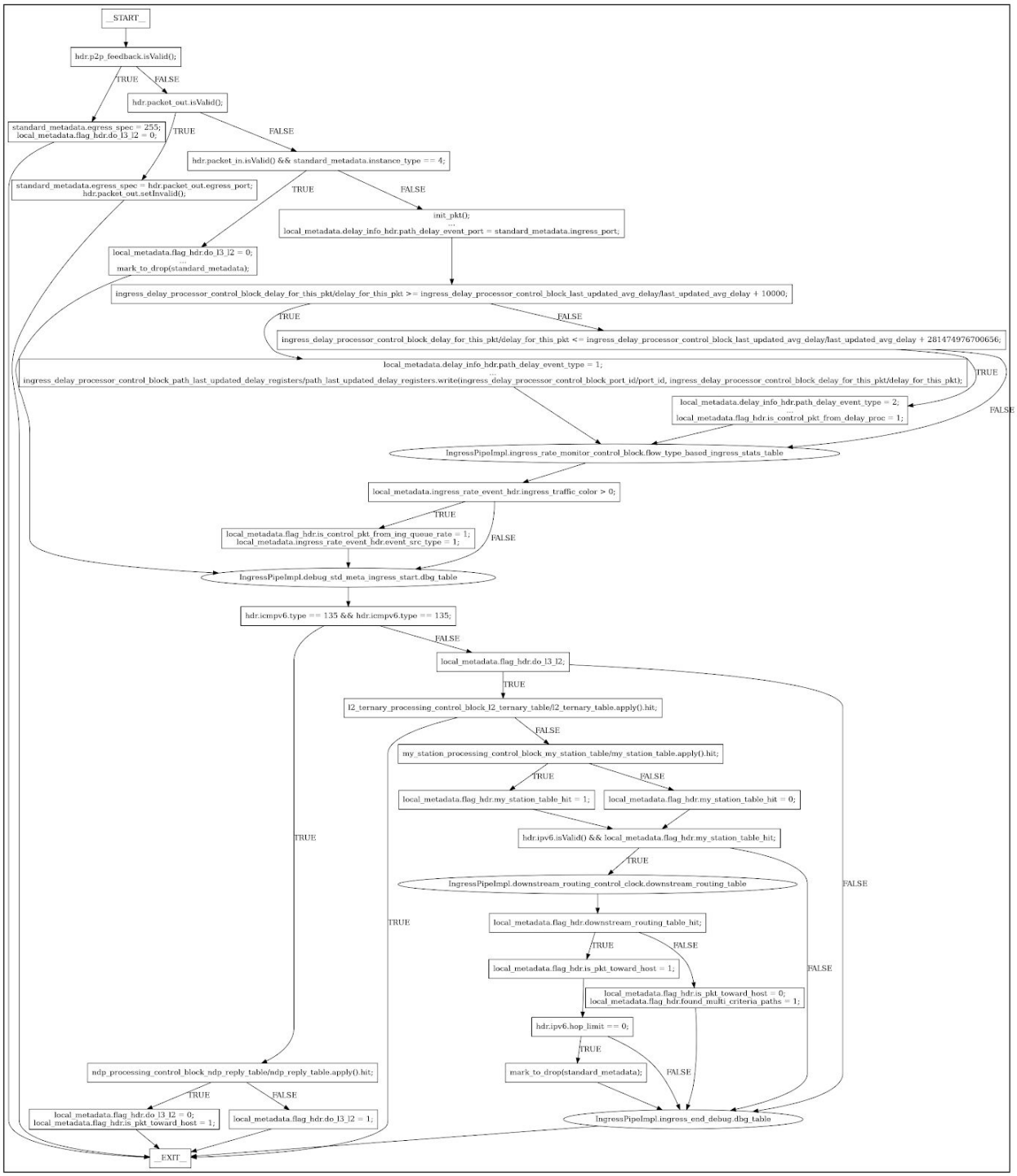


Figure 1: Leaf Switch Ingress Pipeline P4 Program Graph

2. How to map the Egress Stage of the P4 code to HW to match RMT restrictions

Ans: The code for egress stage is divided into 2 blocks

a) Block 1: this block is basically used for controlling when to clone or recirculate a packet to generate a new packet. All the if-else expression is based on some already available value in the pipeline. And here we are only using some equality checking. So simply we can replace this whole if-else logic using a exact match based TCAM

```
if(IS_NORMAL(standard_metadata)){

   egress_queue_depth_monitor_control_block.apply(hdr, local_metadata, standard_metadata);
   egress_rate_monitor_control_block.apply(hdr, local_metadata, standard_metadata);
   #ifdef DP_BASED_RATE_CONTROL_ENABLED
   leaf_rate_control_processor_control_block.apply(hdr, local_metadata, standard_metadata);
   #elif  DP_ALGO_ECMP
   if(standard_metadata.deq_qdepth > ECN_THRESHOLD) hdr.ipv6.ecn = 3; //setting ecm mark
   #endif

   if (local_metadata.is_multicast == true ) {
       exit;
   }
   #ifdef DP_ALGO_CP_ASSISTED_POLICY_ROUTING
   if(IS_RECIRC_NEEDED(local_metadata)) {
       is_recirculation_needed = true;
   }
   #endif
   if(is_recirculation_needed &&   IS_CONTROL_PKT_TO_NEIGHBOUR(local_metadata) &&
IS_CONTROL_PKT_TO_CP(local_metadata)){

       clone3(CloneType.E2E, (bit<32>)(standard_metadata.ingress_port)+ ((bit<32>)MAX_PORTS_IN_SWITCH *
2), {standard_metadata, local_metadata});
   }else if(  IS_CONTROL_PKT_TO_NEIGHBOUR(local_metadata) && IS_CONTROL_PKT_TO_CP(local_metadata)){
       clone3(CloneType.E2E, (bit<32>)(standard_metadata.ingress_port)+ (bit<32>)MAX_PORTS_IN_SWITCH,
{standard_metadata, local_metadata});
   }else if (IS_CONTROL_PKT_TO_CP(local_metadata)) {
       clone3(CloneType.E2E, CPU_CLONE_SESSION_ID, {standard_metadata, local_metadata});
   }else if ( IS_CONTROL_PKT_TO_NEIGHBOUR(local_metadata)) {
        clone3(CloneType.E2E, (bit<32>)(standard_metadata.ingress_port), {standard_metadata,
local_metadata});
   }else{
        //log_msg("Unhandled logic in cloning control block");
   }
}
```

b)

c) Block 2 : The code is shown in following table. The nested if-else can go up to 8 levels. So apparently seems like we need 8 stages. But All the expressions in ef-else just uses the equality operator. So, simply make MAT with 8 fields. And use the TCAM based MAT for executing the actions. The actions involve only copying or modifying some fields from the header vector to another. No memory modification is involved. There are some counters or meters, they are only used to generate some reports for a paper. They have no practical use. So we can just ignore them

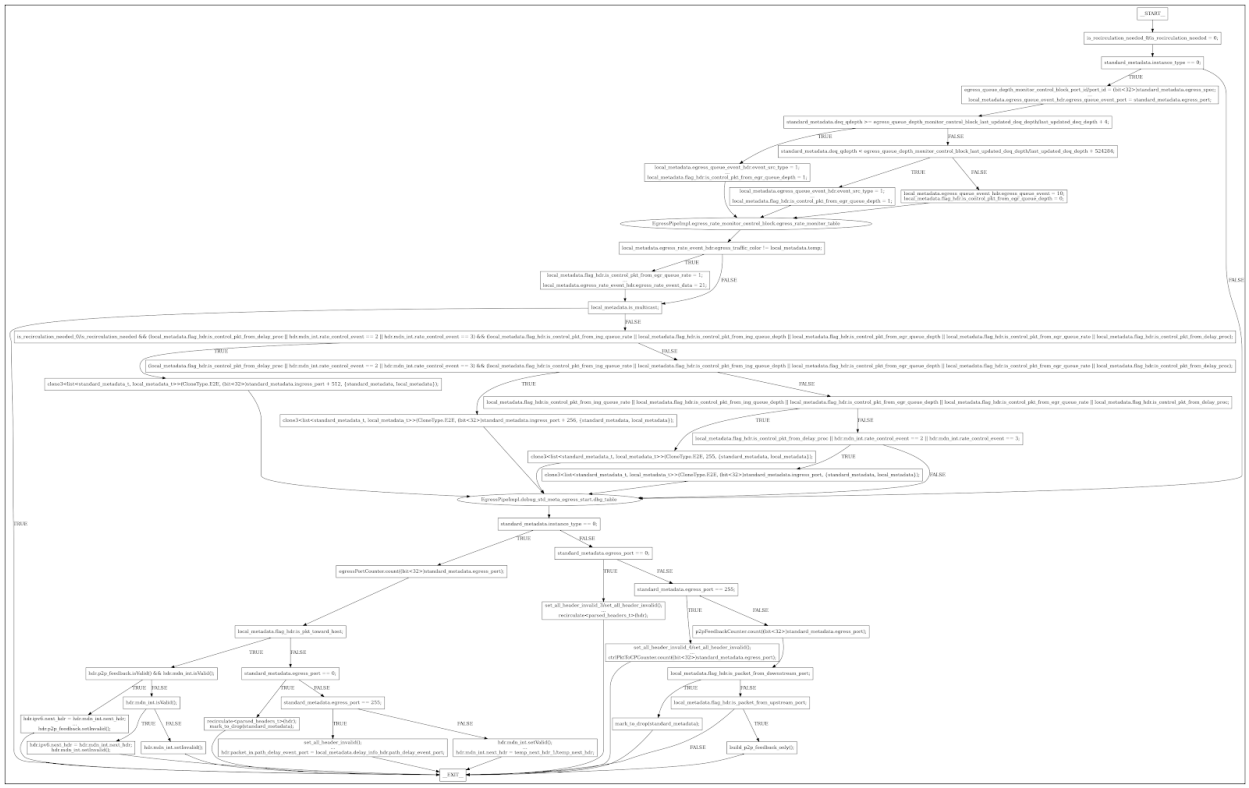
```
if(IS_NORMAL(standard_metadata)){
   egressPortCounter.count((bit<32>)standard_metadata.egress_port);
   if(local_metadata.flag_hdr.is_pkt_toward_host){
        if(hdr.p2p_feedback.isValid() && hdr.mdn_int.isValid()){
            //making some header fields valid/invalid and set value of some header field from metadata
        }else if(hdr.mdn_int.isValid()){
            //making some header fields valid/invalid and set value of some header field from metadata

        }else{
             //making some header fields valid/invalid and set value of some header field from metadata
         }
   }else if (standard_metadata.egress_port == PORT_ZERO) {
         recirculate<parsed_headers_t>(hdr);
         mark_to_drop(standard_metadata);
   }else if (standard_metadata.egress_port == CPU_PORT) {
        //making some header fields valid/invalid and set value of some header field from metadata
   }
   else{
        //making some header fields valid/invalid and set value of some header field from metadata
   }
}else{
   if (standard_metadata.egress_port == PORT_ZERO) {
        //making some header fields valid/invalid and set value of some header field from metadata
        recirculate<parsed_headers_t>(hdr);
   }else if (standard_metadata.egress_port == CPU_PORT) {
        //making some header fields valid/invalid and set value of some header field from metadata
        ctrlPktToCPCounter.count((bit<32>)standard_metadata.egress_port);
   }else{
        p2pFeedbackCounter.count((bit<32>)standard_metadata.egress_port);
        #ifdef DP_BASED_RATE_CONTROL_ENABLED
        if (hdr.mdn_int.isValid()   && (hdr.mdn_int.rate_control_event  ==
RATE_DECREASE_EVENT_NEED_TO_BE_APPLIED_IN_THIS_SWITCH)){
            if(local_metadata.flag_hdr.is_packet_from_downstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }else if (local_metadata.flag_hdr.is_packet_from_upstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }
        }else{
            if(local_metadata.flag_hdr.is_packet_from_downstream_port == true){
                mark_to_drop(standard_metadata);
            }else if (local_metadata.flag_hdr.is_packet_from_upstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }

        }
        if (hdr.mdn_int.isValid()   && (hdr.mdn_int.rate_control_event  ==
RATE_INCREASE_EVENT_NEED_TO_BE_APPLIED_IN_THIS_SWITCH)){
            if(local_metadata.flag_hdr.is_packet_from_downstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }else if (local_metadata.flag_hdr.is_packet_from_upstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }
        }else{
            if(local_metadata.flag_hdr.is_packet_from_downstream_port == true){
                mark_to_drop(standard_metadata);
            }else if (local_metadata.flag_hdr.is_packet_from_upstream_port == true){
                //making some header fields valid/invalid and set value of some header field from metadata
            }

        }
        //build_p2p_feedback_only();
        #else
        if(local_metadata.flag_hdr.is_packet_from_downstream_port == true){
            mark_to_drop(standard_metadata); //Because we do not want to send the feedback packets to hosts
        }else if (local_metadata.flag_hdr.is_packet_from_upstream_port == true){
            build_p2p_feedback_only();
        }
        #endif
   }
}
```

Following is the call graph for Egress stage



3. Example conversion from Dangling if-else to TCAM approach

Ans:
Consider the following example

```
if(IS_NORMAL(standard_metadata)){
   egressPortCounter.count((bit<32>)standard_metadata.egress_port);
   if(local_metadata.flag_hdr.is_pkt_toward_host){
       if(hdr.p2p_feedback.isValid() && hdr.mdn_int.isValid()){
```

Here all the variables used inside the if-else expressions are known beforehand. All those variables are already filled in before executing this block. Whenever some value is not filled in earlier, we can not convert the if-else to TCAM based MAT. Here ins this case and most of our cases uses in P4TE, dangling if-else blocks are convertible in TCAM.

Here, in this example, IS_NORMAL is a macro that depends on 4 boolean variables. Therefor

1) IS_NORMAL needs b bits

2) Local_metadata.flag_hdr.is_pkt_toward_host needs one bit
3) hdr.p2p_feedback.isValid() && hdr.mdn_int.isValid() → needs 2 bit .

Simply, use all those bits as a match field in a TCAM based table. And a number of entries is really small because there will be not too many if-else branching in a system. So we can simply convert them into TCAM based matches.