**P4 Code Implementation for the DPDM Algorithm**

|  |
| --- |
| **// P4 program for Data-plane-based Dynamic Monitoring (DPDM)**  #include <core.p4>  #include <v1model.p4>  // Define a maximum number of state tables  #define MAX\_STATE\_TABLES 14  // Define headers  header\_type ethernet\_t {  fields {  dstAddr : 48;  srcAddr : 48;  ethertype : 16;  }  }  header\_type ipv4\_t {  fields {  version : 4;  ihl : 4;  dscp : 6;  ecn : 2;  length : 16;  identification : 16;  flags : 3;  fragmentOffset : 13;  ttl : 8;  protocol : 8;  headerChecksum : 16;  srcAddr : 32;  dstAddr : 32;  }  }  struct headers {  ethernet\_t ethernet;  ipv4\_t ipv4;  }  // Define metadata for the DPDM program  struct metadata {  bit<32> packetCount;  bit<32> byteCount;  bit<32> flowStatistics;  bit<32> stateRegister[MAX\_STATE\_TABLES];  bit<32> thresholds[MAX\_STATE\_TABLES];  bit<32> adaptiveParameters[MAX\_STATE\_TABLES];  }  // Define state tables  table state\_tables {  key {  hdr.ipv4.srcAddr : exact;  hdr.ipv4.dstAddr : exact;  }  actions {  update\_state;  drop;  }  size : 1024;  }  // Define actions for state tables  action update\_state() {  // Action to update state  // Placeholder for state update logic  }  action drop() {  // Action to drop packets  drop();  }  // Define counters and registers  counter<bit<64>> packetCount;  counter<bit<64>> byteCount;  // Define the control flow  control ingress {  apply(state\_tables);  // Example of packet count update  packetCount.ingress(1);  byteCount.ingress(1);  // Example of state table updates based on packet  if (hdr.ipv4.srcAddr == 0x0A000001) { // Example condition  apply(state\_tables);  }  }  control egress {  // Apply any egress processing if needed  apply(state\_tables);  }  // Define the pipeline  pipeline {  ingress;  egress;  }  // Define the main P4 program  control dpdm\_program {  apply(ingress);  apply(egress);  }  // Define the main function  main {  apply(dpdm\_program);  }  // State Transitions Function  // Implemented in P4 program logic  action transition\_state() {  // Implement state transition logic here  }  // Configure Memory Allocation  // This is configured in the P4 switch’s control plane, not in the P4 program itself  // Holistic Monitoring Function  // Aggregation and monitoring typically done by the control plane or monitoring systems |