```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//  File name : coverage.sv                                           //
//  Author    : G. Andres Mancera                                     //
//  License   : GNU Lesser General Public License                     //
//  Course    : Advanced Verification with SystemVerilog OOP          //
//              Testbench - UCSC Silicon Valley Extension             //
//                                                                    //
////////////////////////////////////////////////////////////////////////

class coverage;

  packet    cov_packet;

  covergroup cov_packet_cg;
    option.name = "Covergroup for all the packet fields";
    mac_dst_addr : coverpoint cov_packet.mac_dst_addr
                   {
                       bins ucast_dst_addr = { [48'h0:48'hFFFFFFFFFF] };
                       bins mcast_dst_addr = { [48'h10000000000:48'hFFFFFFFFFFE] };
                       bins bcast_dst_addr = { 48'hFFFFFFFFFFF };
                   }
    mac_src_addr : coverpoint cov_packet.mac_src_addr
                   {
                       bins ucast_src_addr = { [48'h0:48'hFFFFFFFFFF] };
                       bins mcast_src_addr = { [48'h10000000000:48'hFFFFFFFFFFE] };
                       bins bcast_dst_addr = { 48'hFFFFFFFFFFF };
                   }
    ether_type   : coverpoint cov_packet.ether_type
                   {
                       bins ipv4   = { 16'h0800 };
                       bins arp    = { 16'h0806 };
                       bins ipv6   = { 16'h86DD };
                       bins fcoe   = { 16'h8906 };
                       bins others = default;
                   }
    payload      : coverpoint cov_packet.payload.size()
                   {
                       bins undersize_pkt  = { [0:45] };
                       bins small_pkt      = { [46:256] };
                       bins medium_pkt     = { [257:1000] };
                       bins large_pkt      = { [1001:1500] };
                       bins oversize_pkt   = { [1501:9000] };
                   }
    ipg          : coverpoint cov_packet.ipg
                   {
                       bins zero_ipg_delay     = { 0 };
                       bins short_ipg_delay    = { [1:10] };
                       bins medim_ipg_delay    = { [11:45] };
                       bins large_ipg_delay    = { [46:$] };
                   }
  endgroup


  // Constructor
  function new();
    cov_packet_cg = new();
  endfunction : new


```

```
61    // Class methods
62    task collect_coverage( input packet drv_pkt );
63      this.cov_packet = drv_pkt;
64      cov_packet_cg.sample();
65    endtask : collect_coverage
66
67  endclass
68
```