**Priority Forwarding Module Implementation in Floodlight Controller**

//get port list which have active connection

Set<OFPort> portList = topologyService.getPortsWithLinks(srcSw);

String inline = "";

Path selectedPath = null;

//An array of paths from each active ports

Path[] pathArray = new Path[portList.size()];

int currentRouteIndex = 0;

//An Array of bandwidth for each path

int[] bandwidths = new int[pathArray.length];

//Generating all path from each ports and storing it in path array

 for(OFPort port:portList) {

Path temp = routingEngineService.getPath(

           srcSw,port,dstAp.getNodeId(),

                dstAp.getPortId());

        if (temp != null)

            pathArray[currentRouteIndex++] = temp;

        else

       pathArray[currentRouteIndex++] = null;

}

int bw = 0;

//getting bandwidths of all path and storing it in bandwidth array

for(Path path:pathArray) {

if(path != null) {

List<NodePortTuple> direction =  path.getPath();

           int nodesBandwidth = 0;

          for(NodePortTuple tuple:direction) {

          String nodeId = tuple.getNodeId().toString();

          String portId = tuple.getPortId().toString();

          URL url = new URL("172.16.15.40:8080/wm/statistics/bandwidth/"+nodeId+"/"+portId+"/json");

          HttpURLConnection conn = (HttpURLConnection) url.openConnection();

          conn.setRequestMethod("GET");

          conn.setRequestProperty("Accept", "application/json");

          conn.connect();

          if(conn.getResponseCode() != 200) {

          throw new RuntimeException("Failed because of HTTP Error Code"+conn.getResponseCode());

}

else {

inline = "";

Scanner sc = new Scanner(url.openStream());

while(sc.hasNext()) {

inline+=sc.nextLine();

}

sc.close();

}

String[] temp = inline.split(":|,");

          for(int i=0;i<temp.length;i++) {

          if(temp[i].contains("\"link-speed-bits-per-second\"")) {

          nodesBandwidth += Integer.parseInt(temp[i+1]);

          break;

          }

}

}

bandwidths[bw] = nodesBandwidth;

bw++;

}else {

bandwidths[bw++] = 0;

}

int highestBandwidth = 0;

int secondHighest = 0;

for(int i=0;i<bandwidths.length;i++) {

if(bandwidths[highestBandwidth] < bandwidths[i])

highestBandwidth = i;

else if(bandwidths[secondHighest] < bandwidths[i])

secondHighest = i;

}

byte[] data = pi.getData();

log.info("\n\n\nHello data here: "+Base64.getEncoder().encodeToString(data)+"\n\n\n");

int priority1 = data[0];

int priority2 = data[1];

String priority = "" + priority1 + priority2;

if(priority == "11" || priority == "10") {

selectedPath = pathArray[highestBandwidth];

}

else {

selectedPath = pathArray[secondHighest];

}

}

Path path = routingEngineService.getPath(srcSw,

           srcPort,

            dstAp.getNodeId(),

            dstAp.getPortId());

path = selectedPath;

**Description**

The floodlight Controller consist of different modules like routing modules, forwarding modules, topology module etc. We try to modify the forwarding module of the floodlight controller so that it can follow our priority forwarding algorithm.

The algorithm starts with getting the list of all the connected ports of the switch where the packet arrives. Then it generates the path from each port to the destination and save it in a paths array. If the path from the port to destination doesn’t exist then we save null at that instance.

In the next phase we start to calculate the total bandwidth of each path that we generated earlier. To calculate the bandwidth of a given path from source switch to destination switch, we use the floodlight controller’s built in API that calculates the bandwidth between two points. This API works only with the master version of the floodlight controller. The GET request to this API returns a JSON object which can be parsed to get the bandwidth. The bandwidth for each path is stored in a separate array.

Next, we determine the highest and the second highest bandwidth from this list and assign the high priority packet with the highest bandwidth path and the medium packet with the second highest bandwidth path. The low priority packet gets the default route computed by the floodlight controller itself.