Algorithm For D-ARPSpoof

1 Data Structures Used

1. HashMap<DatapathId,HashMap<OFPort,<Vlan, IP, Mac>> : portIpMap
2. HashMap<MacAddress,Switch-Port> : macPortMap

2 Algorithm

2.1 Handling Packet-IN DHCP Messages

2.1.1 Updating Data Structures

- **DHCP REQUEST**

  1. Delete mapped mac-port pair from macPortMap with mac equals source MAC of incoming Packet.
  2. If portIpMap has entry for inPort then delete that entry.
  3. If macPortMap does not have any entry for incoming packet's source MAC Address, then add those entry in macPortMap.

- **DHCP ACKNOWLEDGEMENT**

  1. Get Switch-Port pair from macPortMap for destination mac address and name that as pair.
  2. Add following entry in portIpMap:
     <*pair.switch* , *pair.port* , vid , dhcpPayload.yourIPAddress , destination-MAC>

2.1.2 Updating Flow Rules

- **DHCP REQUEST**

  1. If portIpMap has entry for received switch and input port then remove flow entry from current switch with :
     - inPort : incomingPacket's input port.
     - etherType: ARP

- **DHCP ACKNOWLEDGEMENT**

  1. Get Switch-Port pair from macPortMap for destination mac address and name that as pair.
  2. Then add a flow rule to block all ARP packet from the current input port. Write this flow with priority 10.
  3. Add flow in *pair.switch* to direct the packet with
     - IP address : dhcp-payload's yourIPAddress
     - vlan-vid : incomingPacket's vlan id
     - inPort : *pair.port*

     to goto flow table 1 of *pair.switch* . Write this flow with priority 20.

2.2 Handling DHCP-ACK PacketOut Messages

Same as section 2.1.1 DHCP ACKNOWLEDGEMENT handling.

2.3 Handling ARP

1. Check for packet's target protocol address and vlan id in ipPortTable. If not present then drop it otherwise get <Switch-Port> pair associated with it. Name it destination.
2. Get <Switch-Port> pair associated with packet's sender protcol address and vlan id. Name it source.
3. Get the path from source to destination using routing service and name it 'path'.
4. Install the rules as:

```
Match m = (EtherType,ARP) + (Vlan-id , Packet vlan-id)+(ArpTargetProtocolAddress , Packet
    ArpTargetProtocolAddress);
List<SwitchPortPair> list = path;
for(index = list.size-1;index>0;index-=2){
   installFlowRule(table-id=2, match=m, switch=list[index].switch, priority=30,
       action=output(list[index].port);
}
pushOFPacketOut(switch=curSwitch, packet=packet-in, outPort=list[1].port,
    inPort=list[0].port);
```

2.4 Handling Switch Added event

1. Write flow rule to forward all ARP packets to controller in flow table 1 of added switch with priority 0. (Default flow for table 1)

2.5 Handling Switch Removed event

1. Remove all entry of removed swith from portIpMap
2. Also, Remove all entry of removed switch from macPortMap.

2.6 Handling link update

1. If any of source switch-port pair and destination switch-port pair exist in our portIPMap, and update type is direct link, then:
   - If source switch exist in switch port pair:
     (a) remove any flow matching inPort to be source port from the respective source switch.
     (b) remove the mac from macPortMap associated with that switch port pair.
     (c) remove the switch-port pair from portIpMap also.
   - If destination switch exist in switch port pair:
     (a) remove any flow matching inPort to be destination port from the respective destination switch.
     (b) remove the mac from macPortMap associated with that switch port pair.
     (c) remove the switch-port pair from portIpMap also.