Floodlight Tutorial

CPS514
September 23, 2015
Brendan Tschaen
Reminder - SDN Stack

- Separation of Control Plane
  - “Brains of the network”

- From Data Plane
  - Routers/Switches
Reminder - SDN Stack

- Control Plane => Floodlight
- Data Plane => Mininet
What is Mininet?

- Mininet\textsuperscript{1} is a network prototyping tool
- Simulate an entire network on your laptop!
- Design network topology
- Test your topology

Creating Mininet Topology

- Built in topologies: linear, tree

```
$ sudo mn --topo=linear,4
$ sudo mn --topo=tree,3
```
Creating Mininet Topology

- Mininet has an API to design your own network topology
- Create a switch
  - `s = self.addSwitch('s1')`
- Create a host
  - `h = self.addHost('h1')`
- Create a link
  - `self.addLink(h, s)`
  - First link added to the switch connects to port 1, then port 2...

```
$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --test pingall
http://mininet.org/walkthrough/#custom-topologies
```
Connecting to a Controller

- Mininet includes a controller, by default
- Often you want to test with your SDN controller
  - Specify connection to “remote controller”
  - `mininet> sudo mn --controller=remote,ip=127.0.0.1,port=6633`
Testing the topology

- After configuration, test that packets are routed correctly
- Pings are a good way to check connectivity
  - `mininet> h1 ping h2`
  - `mininet> pingall`
- Use iperf for TCP packets
  - Source h1, destination h2:
    - `mininet> h2 iperf -s &`
    - `mininet> h1 iperf -c h2`
- Analyze the rules inserted at each switch
  - `mininet> dpctl dump-flows`
Address Resolution Protocol (ARP)

- **Host A (10.0.0.1) wants to talk to Host B (10.0.0.2)**
  - A broadcast request:
    - “I need to talk to the MAC address of IP 10.0.0.2”
    - with a broadcast MAC address of ff:ff:ff:ff:ff:ff
  - B is the one (and only one) that responds with its MAC address
  - A caches the mapping and can now communicate directly with B

- **ARP requests are forwarded to the controller by default**
  - You may need to handle these by forwarding out the appropriate port

- **Mininet can pre-populate host arp tables**
  - `sudo mn --arp`
Reminder - SDN Stack

- Control Plane => Floodlight
- Data Plane => Mininet
What is Floodlight?

- Open-source SDN Controller Platform
- Java based controller
- Supports OpenFlow protocol
- Allows developers to create SDN applications
- v0.91
Basic Controller Functionality

- **Install/Remove forwarding rules on switches**
  - Need to route flows along the correct path
  - Flows are packets with same header

- **Topology Discovery**
  - Need to know what the network looks like
  - Link Layer Discovery Protocol

- **Statistics**
  - Need to know what is happening in the network
Basic Controller Functionality

- Install/Remove forwarding rules on switches
  - Need to route flows along the correct path
  - Flows are packets with same header

- Topology Discovery
  - Need to know what the network looks like
  - Link Layer Discovery Protocol

- Statistics
  - Need to know what is happening in the network
Flows Rules

- Flows consist of <match, action>
- Match - routing information
- Action - where to forward the packet

Match
src ip: 10.0.0.1
dst ip: 10.0.0.2

Action
output: port 2
Flow Rules: Match

- Flow - set of packets that have same value in certain fields
- Match - composition of all same fields

Examples:

- <dst ip: 8.8.8.8, port 80>
- <src mac address: 01:23:45:67:89:ab>
- <protocol: ipv4, in port: 5>
Creating a Match

OFMatch match = new OFMatch();
match.setWildcards(Wildcards.FULL.matchOn(Flag.DL_TYPE).matchOn(Flag.NW_DST).
withNwDstMask(24) );
match.setDataLayerType( Ethernet.TYPE_IPV4 );
match.setNetworkSource( IPv4.toIPv4Address("152.3.140.0") );

- Wildcards determine what information is ignored
- <ip packet, ip=152.3.140.0/24>
Flow Rules: Action

- In SDN switches are dumb
- Action tells them what to do with a matched packet
- Important actions:
  - Send packet out a port
  - Modify the packet’s header

Examples:

<output=3>
<mod_nw_src=123.45.67.89, output=1>
Creating an Action

ArrayList<OFAction> actions = new ArrayList<OFAction>();
OFActionOutput action = new OFActionOutput().setPort((short) 3);
OFActionNetworkLayerSource ofanls = new OFActionNetworkLayerSource();
ofanls.setNetworkAddress( IPv4.toIPv4Address("8.8.8.8") );

examples:
<output = port 3, modify IP address to = 8.8.8.8>
Putting it together

```java
OFFlowMod flowMod = new OFFlowMod();
flowMod.setMatch( match );
flowMod.setActions( actions );
flowMod.setLength( OFFlowMod.MINIMUM_LENGTH + OFActionOutput.MINIMUM_LENGTH +
                     OFActionNetworkLayerSource.MINIMUM_LENGTH ) ;

try {
    sw.write(flowMod, cntx);
    sw.flush();
} catch (IOException e) {
    log.error("Failure writing flowMod", e);
}
```
Getting messages from switches
Getting messages from switches

- Switches can let you know when they see a packet
- Apps need to register themselves as being interested in types of messages
- Apps need to have a handler for each type of message
- Every message from a switch matching this type of message will be forwarded to the switch
IOFMessageListener

First you must register your module:

```java
... implements IFloodlightModule, IOFMessageListener{
    protected IFloodlightProviderService floodlightProvider;
    ...
    public void init(FloodlightModuleContext context) throws FloodlightModuleException {
        floodlightProvider = context
            .getServiceImpl(IFloodlightProviderService.class);
        floodlightProvider.addOFMessageListener(OFType.PACKET_IN, this);
        ...
    }
    ...
}
```
Then you must handle the messages:

```java
... implements IFloodlightModule, IOFMessageListener{

    public Command receive(IOFSwitch sw, OFMessage msg, FloodlightContext cntx) {
        OFPacketIn pi = (OFPacketIn) msg;
        Ethernet eth = IFloodlightProviderService.bcStore.get(cntx,
                        IFloodlightProviderService.CONTEXT_PI_PAYLOAD);

        ...
    }

    ...

}  //end of the implementation
```
Basic Controller Functionality

- Install/Remove forwarding rules on switches
  - Need to route flows along the correct path
  - Flows are packets with same header

- Topology Discovery
  - Need to know what the network looks like
  - Link Layer Discovery Protocol

- Statistics
  - Need to know what is happening in the network
Topology Discovery

- Floodlight discovers and maintains the network topology for you
  - it uses Link Layer Discovery Protocol (LLDP)
- 2 options:
  - Query the controller for the topology
  - Listen to the changes in topology
Link Discovery

First you must register your module:

```java
... implements IFloodlightModule, IOFMessageListener{
    protected IFloodlightProviderService floodlightProvider;
    protected ILinkDiscoveryService linkDiscoverer;
    ...
    public void init(FloodlightModuleContext context) throws FloodlightModuleException {
        floodlightProvider = context.getServiceImpl(IFloodlightProviderService.class);
        linkDiscoverer = context.getServiceImpl(ILinkDiscoveryService.class);
        linkDiscoverer.addListener( this );
        ...
    }
    ...
```
Link Discovery

Then you can get the topology:

```java
... implements IFloodlightModule, IOFMessageListener{
    protected IFloodlightProviderService floodlightProvider;
    protected ILinkDiscoveryService linkDiscoverer;
    ...
    public void myFunction() {
        Map<Link, LinkInfo> linkMap = linkDiscoverer.getLinks();
        for( Map.Entry<Link, LinkInfo> linkEntry : linkMap.entrySet() ){
            Link link = linkEntry.getKey();
            //link.getSrc(), link.getDst(), link.getSrcPort, link.getDstPort
        }
    }
}
```
Running your application

- Need to tell Floodlight where the application is
  - add your application path to: `src/main/resources/META-INF/services/net.floodlightcontroller.core.module.IFloodlightModule`

- Tell floodlight to run your application
  - add your application to:
    - `src/main/resources/META-INF/floodlightproperties`
Your turn!

Modify the included hub (net.floodlightcontroller.hub) into a firewall that drops ICMP packets
Resources

Mininet:
http://mininet.org/walkthrough/

Floodlight:
https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Tutorials

Look at other apps included in Floodlight