Managing a Virtual Network Function using SDN and Control Theory

GENI Summer Camp @ TAMU
May 24th, 2017

Ibrahim Matta

Joint work with Nabeel Akhtar and Yuefeng Wang
GENI resources that we need ...
So before we get started ...

• Login to GENI at http://portal.geni.net and select the GRW-Summer-Camp-TAMU project
• Create two new slices for network and controller
  – Ryu version: http://tinyurl.com/geninfv
  – Follow Step 3 (Obtain resources: 3.1 & 3.2) under Design/Setup
• Bind your resources to an InstaGENI rack
• Reserve your resources
• Later we will login to these VMs
Control Theory
Control Theory

Goal: Design the input valve control to maintain a constant height regardless of the setting of the output valve.
“90% of the real world applications are based on 10% of the existing control methods and theory”

Dimitry Gorinevsky – Stanford University
Examples of Control Theory in CS

- TCP/IP

```plaintext
for every loss {
    w = w/2
}
for every ACK {
    w += 1/w
}
```

\[ \dot{x} = \frac{1-q}{\tau^2} - \frac{1}{2} q x^2 \]

- Analysis and systematic design was developed some 20 years later
- QoS in Caching
- Apache QoS differentiation
- ...
Managing NFV using SDN & Control Theory

Use-case: VNF-IDS load balancing
Network Traffic

Network Sliver

S2

S1

OVS

destination

VNF1

VNF2

Controller Sliver

controller

GENI tesbed
Network Traffic

Network Sliver

S2

S1

OVS

destination

VNF1

VNF2

Controller Sliver

controller

GENI tesbed
Snort as IDS

- Open source IDS system widely deployed
- InfoWorld's Open Source Hall of Fame as one of the "greatest open source software of all time"
- Protocol analysis, content searching and content matching
Recursive InterNetwork Architecture (RINA)

- Clean slate Future Internet Architecture
- Networking is Inter-process communication (IPC)
  - Old principle applied (e.g., TCP RFC 793, 1981)
- DIF (Distributed IPC Facility)
  - processes cooperating to provide IPC
- DAF – processes cooperating to perform a certain function

See GEC19 Tutorial: [www.youtube.com/watch?v=qUDvduy-JEs](http://www.youtube.com/watch?v=qUDvduy-JEs)
Controller
Proportional Integral (PI) Controller
Proportional Integral (PI) Controller

\[
x(t) = \max[0, \min[1, x(t-1) + K(\frac{L(t)}{T} - 1)]]
\]

- **x(t)**: ratio of traffic diverted to VNF2 at time \( t \)
- **L(t)**: load on VNF1
- **T**: target load on VNF1

**Algorithm 1 PI controller**

**Input:** \( IDS_{load}.txt \)

**Output:** \( x(t) \)

1. \( T = 0.5 \)
2. \( x(t - 1) = 0.0 \)
3. \( x(t) = 0.0 \)
4. \( K = 0.2 \)
5. **while True do**
6. \( L(t) = \text{getLoad}(IDS_{load}.txt); \)
7. \( x(t) = \max[0, \min[1, x(t - 1) + K(\frac{L(t)}{T} - 1)]]; \)
8. \( \text{write}(t, x(t)); \)
9. **end while**
PI-based OVS Controller
Algorithm 2 PI-based OVS controller

Input: Flows, \( x(t) \)

1: for all \( f \) in Flows do
2: \hspace{1cm} random = generateRandom();
3: \hspace{1cm} if random > \( x(t) \) then
4: \hspace{2cm} vnfSelected = IDS1;
5: \hspace{1cm} else
6: \hspace{2cm} vnfSelected = IDS2;
7: \hspace{1cm} end if
8: \hspace{1cm} sendFlow(\( f \), vnfSelected);
9: end for
OVS Ryu controller (1)

- In the `_packet_in_handler` method, the code below decides on the type of controller, and which VNF instance the flow should be forwarded:

```python
if out_port != ofproto.OFPP_FLOOD:
    # forward a duplicate to VNF (either vnf1 or vfn2)
    if pkt_ipv4:
        if pkt_ipv4.dst == ip_dst and (pkt_ipv4.src == ip_s1 or pkt_ipv4.src == ip_s2):
            if controller_type == 'RR':
                if vnf_port == vnf1_interface:
                    vnf_port = vnf2_interface
                else:
                    vnf_port = vnf1_interface
            elif controller_type == 'PI':
                vnf_port = self.PISelection()
            else:
                vnf_port = vnf1_interface
    self.logger.info("output port for selected VNF instance: %s", str(vnf_port))
    actions.append(datapath.ofproto_parser.OFPActionOutput(vnf_port))
```

- However, if the source of the IP packet is in the blacklist, then the `_packet_in_handler` method drops the packet by exiting the method call:

```python
if pkt_ipv4:
    checkRe = self.checkAttackerList(str(pkt_ipv4.src))
    if checkRe:
        self.logger.info("%s in attacker list, packet being dropped!", pkt_ipv4.src)
        return
```
OVS Ryu controller (2)

- If ‘PI’ controller is selected, then the `PISelection` method decides to which VNF instance the flow should be forwarded:

```python
def PISelection(self):
    # get the value for load balancer from PI controller output file
    f = open(file_path_pi, 'r')
    txt = f.read()
    txt = txt.strip(' \
')
    # value is saved in 'X' variable
    X = float(txt.split('=')[1])

    # generate a uniform random number between 0 and 1
    ran = R.random()

    # if generated number is > X, then send to VNF1, else send to VNF2
    if ran > X:
        return vnf1_interface
    elif ran <= X:
        return vnf2_interface

    return -1
```
Algorithm 3 Round Robin based OVS controller

Input: Flows

1: vnfSelected = IDS1
2: for all f in Flows do
3:   if vnfSelected == IDS1 then
4:     vnfSelected = IDS2;
5:   else
6:     vnfSelected = IDS1;
7:   end if
8:   sendFlow(f, vnfSelected);
9: end for
Round Robin vs PI Control based load balancer

Simple Round Robin load balancing vs.
Load balancing based on PI control ($T = 50\%$)
Scaling

VNF-1  VNF-2  VNF-3  VNF-4  ...

...
DEMO
Conclusion

• First work that combines Control Theory with SDN/NFV management
• Control Theory can play crucial role in SDN/NFV management
• Use case: Load balancer for IDS (VNF)
  – GENI test-bed is used for realistic experimentation
  – RINA based distributed application is used for monitoring
  – PI-Controller
  – Scaling
Tutorial to reproduce results:

POX version: http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/NFV

Ryu version: http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/NFV/Ryu

Our paper: