Managing a Virtual Network Function using SDN and Control Theory

GENI Summer Camp @ U. Kentucky
May 16\textsuperscript{th}, 2018

Ibrahim Matta

Joint work with Nabeel Akhtar
GENI resources that we need ...

![GENI tesbed diagram]

- **Network Sliver**: S2, S1, OVS, VNF1, VNF2, destination
- **Controller Sliver**: controller
So before we get started ...

- Login to GENI at [http://portal.geni.net](http://portal.geni.net) and select the GRW-Summer-Camp-UKentucky project
- Create two new slices for network and controller
  - Ryu version: [http://tinyurl.com/geninfo](http://tinyurl.com/geninfo)
  - 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

\[
\begin{align*}
\text{for every loss} & \{ \\
& \quad W = W/2 \\
\} \\
\text{for every ACK} & \{ \\
& \quad W += 1/W \\
\}
\end{align*}
\]

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

• \(x\) - transmission rate
• \(\tau\) - round trip time
• \(q\) - loss probability

• Analysis and systematic design was developed some 20 years later
• QoS in Caching
• Apache QoS differentiation
• ...
• See: Optimizing and Modeling Dynamics in Networks. In Hamed Haddadi and Olivier Bonaventure, editors, eBook on Recent Advances in Networking, volume 1. ACM SIGCOMM, August 2013. Licensed under a CC-BY-SA Creative Commons license.
Managing NFV using SDN & Control Theory

Use-case:
VNF-IDS load balancing
System Overview

Network Sliver
- S2
- S1
- OVS
- VNF1
- VNF2
- destination

Controller Sliver
- controller

GENI testbed

Controller Sliver
- Snort Alerts
- Attacker list
- OVS rules
- OVS controller

RINA Network
- RINA App
- RIB
- Load Balancer
- IDS state info
- load balancing info

VNF1
- SNORT IDS
- RINA App

VNF2
- SNORT IDS
- RINA App

Controller
- Attack Analyzer
- Load Balancer
Network Traffic

Network Sliver

S2

S1

OVS

destination

VNF1

VNF2

Controller Sliver

controller

GENI testbed
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

http://csr.bu.edu/rina/
See GEC19 Tutorial: www.youtube.com/watch?v=qUDvduy-JEs
Proportional Integral (PI) Controller
Proportional Integral (PI) Controller

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

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

Algorithm 1 PI controller

<table>
<thead>
<tr>
<th>Input: ( IDS_{load}.txt )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: ( x(t) )</td>
</tr>
</tbody>
</table>

1: \( T = 0.5 \)
2: \( x(t - 1) = 0.0 \)
3: \( x(t) = 0.0 \)
4: \( K = 0.2 \)
5: while True do
6: \( L(t) = getLoad(IDS_{load}.txt); \)
7: \( x(t) = \max[0, \min[1, x(t - 1) + K\left(\frac{L(t)}{T} - 1\right)]] \)
8: 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:     random = generateRandom();
3:     if random > $x(t)$ then
4:         vnfSelected = IDS1;
5:     else
6:         vnfSelected = IDS2;
7:     end if
8:     sendFlow($f$, vnfSelected);
9: end for
Attack Analyzer

Analyze Snort Alerts!
OVS controller with Attack Analyzer

Block Attack Traffic!
Load Balancer:
Round Robin based OVS Controller

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
Load Balancer: Round Robin vs PI Control

Simple Round Robin load balancing

Load balancing based on PI control ($T = 50\%$)
Load Balancer:
Load balancer vs No load balancer

• Attacks are detected significantly faster with load balancer

Port Scanning Attack (a) 2.6 seconds with load-balancer (b) 5.4 seconds without load-balancer
Scaling

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

...
DEMO

Experiment 3: Handling Intrusion with Ryu Controller: Ping Attack
To Do:

Go to tutorial http://tinyurl.com/geninfv and do following steps:

1. Design/Setup
2. Execute
3. Experiments:
   - **Experiment 1**: Load Balancing using Round Robin Control with Ryu Controller
   - **Experiment 2**: Load Balancing using Proportional Integral (PI) Control with Ryu Controller
   - **Experiment 3**: Handling Intrusion with Ryu Controller: Ping Attack
   - **Experiment 4**: Handling Intrusion with Ryu Controller: Port Scanning Attack
To Do:

Go to tutorial http://tinyurl.com/geninfv and do following steps:

1. Design/Setup
2. Execute
3. Experiments:
   - **Experiment 1**: Load Balancing using Round Robin Control with Ryu Controller
   - **Experiment 2**: Load Balancing using Proportional Integral (PI) Control with Ryu Controller
   - **Experiment 3**: Handling Intrusion with Ryu Controller: Ping Attack
   - **Experiment 4**: Handling Intrusion with Ryu Controller: Port Scanning Attack
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
• Attacks detected faster with load balancer!
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 papers:


[Demos at CNERT and IEEE INFOCOM 2018]