NFV Tutorial

Managing a Virtual Network Function using SDN and Control Theory

Nabeel Akhtar       Ibrahim Matta
Control Theory
Control Theory

Goal: Design the input valve control to maintain a constant height regardless of the setting of the output valve.
Control Theory

“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

  for every loss {
    \[ W = W/2 \]
  }

  for every ACK {
    \[ W += 1/W \]
  }

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

  • \(x\) - 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

• ...
Managing NFV using SDN & Control Theory

Use-case:
VNF-IDS load balancing
System Overview

Network Sliver:
- S2
- S1
- OVS
- destination

Controller Sliver:
- VNF1
- VNF2
- controller

GENI tesbed:
- VNF1 SNORT IDS
- RINA Network
- DAF
- CDAP
- Load Balancer
- OVS controller
- OF rules
- IDS load info
- load balancing info
GENI Test-bed
Network Traffic

Network Sliver
- S2
- S1
- OVS
- destination

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
RINA
Recursive InterNetwork Architecture (RINA)

- Clean slate Future Internet Architecture
- Networking is Inter-process communication (IPC)
- DIF (Distributed IPC Facility) – processes cooperating to provide IPC
- DAF – processes cooperating to perform a certain function

http://csr.bu.edu/rina/
Controller
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

**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: \( L(t) = getLoad(IDS_{load}.txt); \)
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
Algorithm 3 Round Robin based OVS controller

Input: Flows

1: \textit{vnfsSelected} = \textit{IDS}1
2: \texttt{for all } f \texttt{ in Flows do}
3: \hspace{1em} \texttt{if } \textit{vnfsSelected} == \textit{IDS}1 \texttt{ then}
4: \hspace{2em} \textit{vnfsSelected} = \textit{IDS}2;
5: \hspace{1em} \texttt{else}
6: \hspace{2em} \textit{vnfsSelected} = \textit{IDS}1;
7: \hspace{1em} \texttt{end if}
8: \texttt{sendFlow}(f, \textit{vnfsSelected});
9: \texttt{end for}
Round Robin vs PI Control based load balancer

Simple Round Robin load balancing

Load balancing based on PI control ($T = 50\%$)
Scaling

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

...
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: