PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks

Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba
David R. Cheriton School of Computer Science, University of Waterloo
[mfbari|sr2chowdhury|r5ahmed|rboutaba]@uwaterloo.ca
Roadmap

• Motivation
• Our Contribution
• Our Approach
• Simulation Results
• Conclusion & Future Work
Motivation

• Network management systems are being continuously challenged to satisfy application QoS requirements
• Policy based management can tackle these challenges
• Recently emerging field of Software Define Networking (SDN) can provide features like:
  • Per flow control
  • Dynamic flow aggregation
  • Dynamic traffic classes
  • Avoid protocol clutter problem
  • Ease of deployment
• Policy based management can be coupled together with SDN to provide autonomic policy based management
Roadmap

• Motivation
• Our Contribution
• Our Approach
• Simulation Results
• Conclusion & Future Work
Our Contribution

• We have designed and implemented a prototype of an autonomic QoS policy enforcement framework, PolicyCop that:
  • Leverages the programmability offered by SDN for
    • Dynamic traffic steering
    • Flexible Flow level control
    • Dynamic traffic classes
    • Custom flow aggregation levels
  • Monitors the network to detect policy violations
  • Reconfigures the network to reinforce the violated policy
Roadmap

• Motivation
• Our Contribution
• Our Approach
• Simulation Results
• Conclusion & Future Work
Our Approach
PolicyCop: Control Plane

PolicyCop

- Admission Control
- Routing
- Device Tracker
- Statistics Collector
- Rule DB

SDN Controller

Control Plane

NB API

OpenFlow
PolicyCop: Management Plane

PolicyCop

Management Plane

Policy Enforcer
- Topology Manager
- Resource Manager
- Policy Adaptation
- Resource Provisioning

Policy DB

Policy Validator
- Event Handler
- Policy Checker
- Traffic Monitor

Event Manager

Autonomic Action

Manual Action

NB API

Event Types
Our Approach (Workflow)
Roadmap

- Motivation
- Our Contribution
- Our Approach
- Simulation Results
- Conclusion & Future Work
Experimental Setup

- 5 Open vSwitches (OVSs) & 4 hosts
- OVSs’ interconnected with GRE tunnels to simulate bandwidth and latency
- One floodlight controller
- Used `iperf` to generate traffic
Use Case 1: Link Failure
Use Case 2: Throughput Violation
Roadmap

• Motivation
• Our Contribution
• Our Approach
• Simulation Results
• Conclusion & Future Work
Conclusion & Future Work

• We have
  • Presented the design of PolicyCop, an autonomic QoS policy enforcement framework for SDN
  • Demonstrated the effectiveness of PolicyCop through a working prototype

• Our next step
  • Implement all component of PolicyCop
  • Interface with existing policy specification languages (e.g., Ponder)
  • Provide a collection of controller applications for other network management function
Questions?