NGWN Final Review

DATE: 2017-06-06
Overview

Flow Scheduling
- Hedera
- Mahout
- DARD

Security
- Topoguard
- SE-Floodlight

Scalability
- Onix
- Kandoo

Update
- Dionysus
Hedera: Dynamic Flow Scheduling for Data Center Networks

- Current network in datacenter uses ECMP to route the traffic.
- ECMP is really good when handling mice flows but not as good when handling the elephant flows.
Hedera: Dynamic Flow Scheduling for Data Center Networks

- Since SDN has the advantage over traditional network because of global view, controller can schedule the flows to avoid any collision between each other.
- The challenge of this question is that scheduling these flows is NP-Hard.
- Hedera uses simulated annealing to find a near optimal solution, they also optimize the algorithm by reducing the search space of the algorithm (per flow per core switch -> per destination host per core switch)
Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection

- Although Hedera successfully uses SDN to enhance the performance of traffic management services, it suffers from scalability.
- Mahout uses push-based method to detect the elephant flows.
- It uses a simple algorithm to tag elephant flow.
- Controller just install certain rules into the switches to detect the elephant flow.
DARD: Distributed Adaptive Routing for Datacenter Networks

- Although Mahout uses host to do the elephant detection and successfully reduce the load of SDN, the invent of DARD push this method to the limit.
- DARD let the hosts to decide which path to take by itself.
- The controller only needs to install proactive rules first.
Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures

- The work points out two attacks that exploit vulnerabilities inside SDN topology discovery services:
  - **Host Location Hijacking Attack**
    - Adversary sends forge packets to cheat the controller believing the host moves to another location.
    - This attack can be used to launch other attacks like phishing.
  - **Link Fabrication Attack**
    - Adversary sends forge packets to cheat the controller believing there’s a fake link between two ports.
    - The attack can let adversary launch other attacks like DoS attack, MITM attack, etc.
Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures

- The work uses verification methods to defense the attack
- For link fabrication attack, the work uses port flag to block the attack
- For host location hijacking attack, the work observe two events to determine if there's an attack.
  - Port down message
  - Use packet to probe the original location to make sure the host really leave the place.
Securing the Software Defined Network Control Layer

- The author point out several problems inside current SDN control plane.
  - Controller does not know if the rules have confliction, or the rules combine together will conflict another rules
  - Controller does not know who install the rules
  - Every apps running on control plane can modify any configuration inside the switches.
  - Apps can run malicious code and bring down the whole controller
Securing the Software Defined Network Control Layer

- The work uses several methods to solve these problems.
  - Applications have different priorities
  - Some OpenFlow messages are limited to certain applications, applications need authentication before sending messages.
  - A chain rule construct module and chained conflict analysis algorithm. (the work improved the conflict detection to let the controller know if there is chained conflict)
  - Separate the controller process from the application processes.
  - An audit server to log the behaviors of the applications.
The work discovered that previous work only calculate a feasible solution for the network update, but does not care about the order.

The time a switch update the rule varies due to workload, number of rules to be updated, the type of update operation and the priority of the rules that need to update, some orders may take less time then others.

Dionysus will consider the above conditions and schedule the update order for the network.
Dynamic scheduling of network updates

- Dionysus will first construct a dependency graph every time it wants to schedule the network update.
- If there’s a loop inside the dependency graph, Dionysus will first view the cycle as a node and run the algorithm, then it’ll try to handle all the nodes that are really a cycle.
- If there does not exist any valid operation for the cycle, Dionysus will limit certain flow rate to free some space for updating.
Onix: A Distributed Control Platform for Large-scale Production Networks

- Onix is a distributed controller.
- It consists of four components
  - Network Control Logic
  - Onix Instance
  - Connectivity Infrastructure
  - Physical Infrastructure
Onix: A Distributed Control Platform for Large-scale Production Networks

- The NIB is Onix’s database that stores all the information an application needs.
  - There are two kinds of data store inside NIB.
    - Volatile Data
    - Consistent Data
- For scalability issues, Onix supports two strategies to reduce the workload of every Onix instance.
  - Partition
  - Aggregation
Kandoo

- The author proposed Kandoo, a hierarchical SDN control plane that solve the scalability issue existed inside the SDN architecture.
- Kandoo define two kinds of controller, Root controller and Local controller.
- By defining what applications to run in root controllers and what applications to run in local controllers, Kandoo can manage large network without scalability issue.