

Mawlana Bhashani Science and Technology University



Department of Information and Communication Technology

Course Code : *ICT-4101*
Course Name : *Telecommunication Engineering*
Name of the Assignment : *Controller Rest API*
Assignment No. : *03*

Submitted to:
Mr. Nazrul Islam
Assistant Professor,
Dept. of ICT, MBSTU

Submitted by:
Md. Nazmul Hasan
ID: IT-17005
Session: 2016-17
4th Year 1st Semester

Date of Submission: 20-October-2020

Controller Rest API

Objective:

- Understand the working principles of Controller Rest API.
- Understand the difference between proactive and reactive installation flows.

Q: 01. Explain the OpenFlow mode of operations.

Answer: When using OpenFlow to populate tables in switches there are essentially three modes of operation:

- **Reactive Flow Instantiation:** When a new flow comes into the switch, the OpenFlow agent software on the switch does a lookup in the flow tables. If no match for the flow is found, the switch creates an OFP packet-in packet and sends it off to the controller for instructions. Reactive mode reacts to traffic, consults the OpenFlow controller and creates a rule in the flow table based on the instruction. This behavior was tested on previous lab.
- **Proactive Flow Instantiation:** Rather than reacting to a packet, an OpenFlow controller could populate the flow tables ahead of time for all traffic matches that could come into the switch. By pre-defining all of the flows and actions ahead of time in the switches flow tables, the packet-in event never occurs. The result is all packets are forwarded at line rate. Proactive OpenFlow flow tables eliminate any latency induced by consulting a controller on every flow. This behavior will be tested on this lab.
- **Hybrid flow instantiation:** A combination of both would allow for flexibility of reactive for particular sets a granular traffic control that while still preserving low-latency forwarding for the rest of the traffic.

Q: 02. What is API? How does it use in the context of SDN?

Answer: Application program interface (API) is an interface presented by software (such as a network operating system) that provides the capability to collect information from or make a change to an underlying set of resources.

APIs in the context of SDN: In an open SDN model, a common interface discussed is the northbound interface (NBI). The NBI is the interface between software applications, such as operational support systems, and a centralized SDN controller. One of the common API technologies used at the northbound interface is the Representational State Transfer (REST) API. REST APIs use the HTTP/HTTPS protocol to execute common operations on resources represented by Uniform Resource Identifier (URI) strings. An application may use REST APIs to send an HTTP/HTTPS GET message via an SDN controller's IP address. That message would contain a URI string referencing the relevant network device

and comprising an HTTP payload with a JSON header that has the proper parameters for a particular interface and statistic.

Q: 03. What northbound API?

Answer: Software-defined northbound application program interfaces (SDN northbound APIs) are usually SDN RESTful APIs used to communicate between the SDN Controller and the services and applications running over the network. These APIs can be used to facilitate efficient orchestration and automation of the network to align with the needs of different applications via SDN network programmability.

Q: 04. How do northbound APIs work?

Answer: Northbound APIs are the link between the applications and the SDN controller. The applications can tell the network what they need (data, storage, bandwidth, and so on) and the network can deliver those resources, or communicate what it has.

These APIs support a wide variety of applications. This is possibly why SDN northbound APIs are one of the most moldable components in an SDN environment — a variety of possible interfaces exist in different places up the stack to control different types of applications via an SDN Controller.

Examples of the types of network applications that can be optimized via the northbound interface include load balancers, firewalls, or other software-defined security services, or orchestration applications across cloud resources.

Northbound APIs are also used to integrate the SDN Controller with automation stacks, such as Puppet, Chef, SaltStack, Ansible, and CFEngine, as well as orchestration platforms, such as OpenStack, VMware's vCloudDirector or Apache's open source CloudStack. The goal is to abstract the inner-workings of the network so that application developers can "hook" into the network and make changes to accommodate the needs of the application. project floodlight northbound REST APIs Placeholder Image

Conclusion: Before doing this assignment, I've not known about Restful API. After finishing this assignment, I come to learn about restful APIs in the context of SDN. Also understand the difference between proactive and reactive installation flows. I can't apply my knowledge practically because of the absenting of the Zodiac FX Open Flow switch.