IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

Enhancing Network Security throughSoftware Defined Networking (SDN)

N Bhagyasree[1], Nischal M R[2], Pavan Kumar[3], K Surendra[4]
Student, Department of Computer Science and Engineering[1,2,3]
Guide, Department of Computer Science and Engineering[4]
Alva's Institute of Engineering and Technology, Mangalore, India

Abstract: *The IoT connects many of the home appliances in the world, from intelligent thermostats to intelligent vehicles. By the end of 2015, we had 9 billion connected stuff. The Gartner forecast shows that the number of devices in the network will exceed 50 billion by 2020. Most of the connected devices in the existing network are based on obsolete security infrastructure and encryption, while many do not have provisions for remote updating of these devices. Taking into account the number of IoT devices, from off-shelf motion monitoring to machine tools, it is not possible to check which functions end up with any specific service or product. In short, this is the problem: you can't trust the integrity of the security of the device and determine exactly where the data is processed and stored. The SDN architecture is designed to increase the routing and networking performance of the existing networks by isolating the control plane from the data plane. The basic concept of Software Defined Networking can be applied to IoT Multi networks; however, the standard SDN implementations for wired networks are not directly suitable for distributed and ad-hoc mesh networks, which are common in IoT systems. However, the SDN architecture changes the pattern of communication of the IoT network, leading to a new approach to the manifestation of the Securities IoT network. Centralized data layer control and control layer not only simplifies data package management, but also increases safety. As mentioned above, the amount of data that flows into these systems is significant. Proper management of traffic and load balancing will help to simplify the data flow in some ways. The total system's power consumption may be reduced by the central station, which requires less power than the distributed control. To sum up, introducing SDN into IoT will help IoT and stimulate IoT in people's regular lives.*

Keywords: Software-Defined Networking (SDN), controller, OpenFlow, OpenDayLight

I. INTRODUCTION

The Internet of Things (IoT) is a collection of embedded devices which are interconnected together with some network and they collect, share and process the aggregated information to execute some job over it. The most common devices in IoT are sensors which sense the data and give the data to some processing device or cloud. In today‟s world everything needs to be smart. Smart houses, smart infrastructure and smart transportation these applications are realizable due to the advancement of IoT. The concept of IOT was put forward as early as 1999 by Professor Ashton at the American Auto-ID center of MIT. In 2005, ITU gave it the corresponding definition. IOT has become the hot spot question for scientific research technology personnel, because people realized that it has immeasurable potential, such as intelligent power network, intelligent traffic, intelligent logistics, intelligent building, GPS navigation, industrial monitoring, modern agriculture, public security, environmental management, remote medical treatment and digital urban management, digital home, digital battlefield and so on. People expect IOT to bring a lot of convenience in the vision. The expansion application of IOT and controllability of information has become a pair of contradiction. If no the in-depth research, these problems will cause the user to panic.

The Software-Defined Networking (SDN) [1], an emerging network architecture allows network supervisors to monitor network. The key characteristics of SDN are manageable, dynamic, cost-effective and malleable. Some of the major features of SDN are:

- **Programmable Networks**: The disengagement between control and forwarding planes allow a direct

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

programming of the network without caring about forwarding functions.

- **Flexible Development**: This disengagement also allows a fast and adaptable modification of flows and network functions.
- **Centralized Management**: SDN architecture is based on a centralized controller that keeps a global vision of the network, and also interacts with by altering flows.
- **Allows to Automate the Device Programming**: SDN allows administrators to configure self-written programs (as SDN does not belong to any software) that speed up the procedures of configuration, optimization and administration of network resources.
- **Based on Open Standards**: It allows the simplification of the network management as it does not depend on any specific providers, adaptation devices or protocols.

The objective behind the design of SDN architecture is to solve the concerns in stagnant traditional networks architecture. This objective is accomplished by splitting the network plane into the control and data planes. In legacy network, packets are moved to the same destination along the same path according to the pre-loaded primitives. On the other side considering SDN environment, the forwarding rules are managed by the centralized controller. Every SDN model is featured by an SDN Controller, along with southbound APIs and northbound APIs.

The controller is an application running on a server somewhere within the network. Controllers and network devices requires some interface to communicate; OpenFlow [2] protocol is one of the interface used by the most of the SDN controller. OpenFlow facilitates the network administrators with a set of entities that empowers them to define network flows and to define the path that they will follow without affecting the existing traffic. It also provides methods to define policies which assists in achieving certain characteristics, like having higher band width, suffering less latency, reducing the number of hops and reducing the required energy that needs traffic to reach its destination. The OpenFlow protocol indirectly helps administrators to move network control out of proprietary network switches. Applications interact with the controller with the help of northbound APIs. In the same way, all the instructions from controller to network devices are routed through the southbound APIs. Furthermore, southbound APIs provide a layer of abstraction, making network device type's indifferent for the controller and applications.

- **Southbound APIs:** They make possible the communication between the controller and network elements. They allow making dynamic changes to the network elements in order to adapt them to the requirements at real time. Moreover, they also allow the independence between the underlying elements of the controller. This is because the controller only has to be aware of sending instructions, that adaptation APIs will translate and route to the appropriate network element.
- **Northbound APIs**: They are used to communicate the controller with applications that are running over the network. These Northbound APIs boosts research and enable effective automation and orchestration of the network. Northbound APIs are the vital APIs in the SDN technology, since they play key role in providing interaction between network and the business applications.

II. Related Work

Today, business and technical network requirements include enhancing performance and realizing broader connectivity. Companies have to meet more and more industry specific security regulations and there is a growing demand for mobility. In order to comply with all of these criteria, networking protocols have evolved significantly over the last few decades. However, the way traditional networks are set up, deploying one protocol to realize these needs organization wide, is quite the challenge. The traditional approach to networking is characterized by two main factors:

- Network functionality is mainly implemented in a dedicated appliance. In this case, „dedicated appliance" refers to one or multiple switches, routers and/or application delivery controllers.
- Most functionality within this appliance is implemented in dedicated hardware.

The business organizations are defied with the limitations that get together with this hardware centric approach, such as:

- Traditional configuration is time-consuming andinefficient.
- Multi-vendor environments require a high level ofexpertise.
- Traditional architectures complicate networksegmentation Networking Software Defined Networking (SDN) is

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

rapidly becoming the new buzzword in the networking business. Expectations are that this emerging technology will play an important role in overcoming the limitations associated with traditional networking. This concerns the two network device planes [3],

- The plane that determines where to send traffic (controlplane).
- The plane that implements these decisions and forwards traffic (data plane)

Software Defined Networking is projected to have several business paybacks, viz., More configuration accuracy, consistency, flexibility and Data flow optimization, Integration, Visibility and reporting (Device visibility, Events and alerts and Traffic reporting), Security (Network Zoning, Live Threat Protection and Port Security), Reliable connectivity (Offline measures, Uplink Balancing, Traffic Prioritization).

The current scenario states that growing number of physical objects are being connected to the Internet at an alarming rate realizing the idea of the Internet of Things (IoT) [4]. IoT applications require communication compatibility between heterogeneous things. In addition, the traditional Internet architecture needs to be revised to match the IoT challenges. The IoT should be matured enough to interconnect large number of heterogeneous objects through the Internet, so there is an acute need for a flexible layered architecture.

Middleware is indispensable to ease the development of the diverse applications and services in IoT [2]. Although the existing middleware solutions address many requirements associated with middleware in IoTs, some requirements and related research issues remain relatively unexplored, such as scalable and dynamic resource discovery and composition, system-wide scalability, reliability, security and privacy, interoperability, integration of intelligence and context awareness.

UbiFlow [5] is a software-defined IoT system for efficientflow control and mobility management in urban Multi networks. Besides flow scheduling, it shifts handoveroptimization, mobility management and access pointselection functions from the relatively resource constrained IoT devices to more capable distributed controllers. The distributed controllers are organized in a scalable and fault tolerant manner. The system was evaluated through simulation and testbed.

The paper [3] presents an original SDN controller design in IoT Multinetworks whose central, novel feature is the layered architecture that enable flexible, effective, and efficient management on task, flow, network, and resources. A novel vision on tasks and resources in IoT environments, and how to bridge the gap between abstract high level tasks and specific low level network/device resources, is illustrated. The Network Calculus model is modified to accurately evaluate the end-to-end flow performance in IoT Multinetworks, which is further serving as essentials of a novel multi-constraints flow scheduling algorithm under heterogeneous traffic pattern and network links. The semantic modeling approach performs resource matching and the GA-based algorithm schedules flows. Those techniques can be viewed as plug- ins and can be adjusted or replaced in different IoT scenarios.

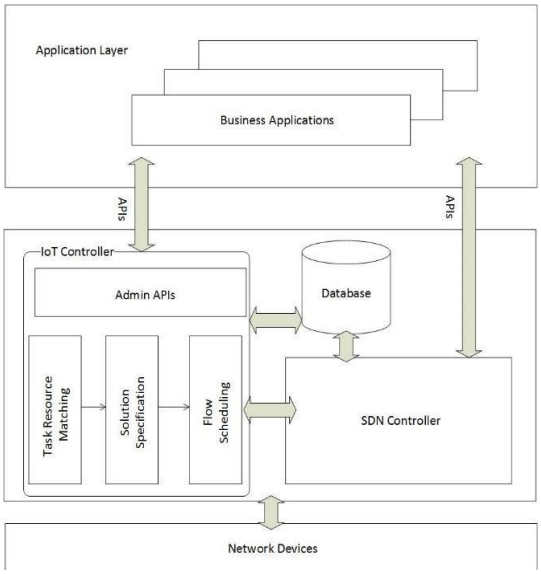

Figure 1: System Architecture

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

In Fig. 1 the task-resource matching component of the controller maps the task request onto the existing resources in the network. Once resource set solution is selected, the service solution specification component of the controller maps the characteristics of the devices and services involved in that solution to specific requirements for devices, networks, and application constraints. The Flow Scheduling component takes these requirements and schedules flows that satisfy them. Finally the controller triggers the necessary communications in the IoT network. There are many SDN controllers designed by network vendors, which provides SDN infrastructure for development. OpenDayLight (ODL) [6] is one of the SDN controller which is much developed than its competitor.

ODL builds infrastructure for SDN deployments. It provides a model-driven service abstraction (service abstraction layer) that allows users to develop applications that easily work across a wide variety of hardware and southbound protocols. It relies on the following technologies:

- **Maven**: Project management tool that simplifies and automates dependencies within a project or betweendifferent projects. This tooling will help developers to manage all the required plugins and dependencies, as well as to provide a project start-up using its defined archetypes.
- **Java**: It is the programming language that is used to develop applications and features in the OpenDaylight‟s controller. Developing in Java provides a valuable compile-time safety, as well as an easy way to implement defined services.
- **Open Service Gateway Interface (OSGi)**: It is the back-end of OpenDayLight controller as it allows to dynamically load bundles and JAR packages, and bind modules for exchanging information.
- **Karaf**: It is an application container built on top of OSGi, which simplifies aspects of packaging and installing applications.
- **Yet Another Next Generation (YANG):** it is the key- point of the model-driven behavior in the controller. Developers willuse YANG to model an application functionality, and to generate APIs from the defined models, which will be later used to provide its implementations. YANG supports modeling operational and configuration data, as well as RPC and notifications.

Genetic Algorithm (GA) is stimulated by the theory of natural evolution and its principles. It is one of the directed random search techniques, employed to find a near- optimal solution for many larger problems in complex multi-dimensional search spaces. Paper [1], proposes a multipurpose optimization method for QoS routing based on GA. The proposed method is a source based routing method and has a flexible and adaptive behavior. The proposed method can support QoS routing for multiplemetrics. The adaptive routing mechanism has a load balancing system among alternative paths. The individual genes are used to express the connected nodes of a route.

III. Conclusion

The SDN-based infrastructure and technologies will encounter the IoT at the crossroads of VPN enervation, uptime challenges and limited network resources. The projected result is that Software-Defined Networking will assist in boosting the expansion of IoT-enabled devices, enable more effective network resource sharing and enhance IoT service-level agreements (SLAs). In return, many vendors expect IoT will support SDN decisions and nourish greedy policy engines.

References

[1] "ONF SDN Evolution," 2016.

[2] K. Sood, S. Yu, and Y. Xiang, "Software defined wireless networking opportunities and challenges for internet of things: A review," 2015.

[3] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in 2014 IEEE network operations and management symposium (NOMS), pp. 1–9, IEEE, 2014.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

[5] D.Wu, D. I. Arkhipov, E. Asmare, Z. Qin, and J. A. McCann, "Ubiflow: Mobility management in urban-scale software-defined iot," in 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 208–216, IEEE, 2015.

[6] "Opendaylight Project" https://wiki.opendaylight.org/