Conceptual Framework for Secure Cluster Management In Software Networks

Chakali Maddilety [A] And Dr Pankaj Kawad Kar[b]

[a]*Research Scholar, Dept. of Computer Science & Engineering,Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, MadhyaPradesh, India*
[b]*Research Guide, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, MadhyaPradesh, India*

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: Software Defined Networks (SDN) is a new concept of network architecture that provides the separation of control plane (controller) and data plane (switches) in network devices. Hidden Markov Model (HMM) scheme is proposed for cluster management. The utilization of the bandwidth is an issue because of the extensibility of the control plane. The Control plane in SDN focuses on a centralized solution, where a single control entity has an overall view of the network. To meet the administration, the board necessities of large-scale network situations, the controls plane is regularly completed in the sort of distributed controller clusters, Cluster management technology monitors all sort of events and should keep up a reliable global network status, which usually leads to big data in SDNs. Cluster management technology monitors all types of events and must maintain a consistent global network status, which usually leads to big data in SDNs. Simultaneously, the cluster security is an open issue because of the programmable and dynamic features of SDNs.

Keywords: Clustering, SDN, Hidden Markov Model, cluster management.

Introduction

A new concept of network architecture such as Software Defined Networking (SDN), decouples the network control (control plane) from the underlying network resources (data plane) being controlled through a centralized controller using OpenFlow as standard protocol in their communications. The appropriate SDN-based network behaviour depends on the capacity of the controller to make good decisions. The controller can not only correct failures inside the network, but it can also prevent future issues based on the available monitoring information. For instance, the controller can prevent DDoS or a decreasing of QoS/QoE analyzing metrics as data rate, packet loss or delay in the links within the network. Therefore, introducing SDN enables the exchange of network information with flexibility and adaptability.

Software Defined Networking and OpenFlow Protocol

Software Defined Networking (SDN) has emerged as the industry’s response to meeting these challenges. SDN allows networks to react dynamically to changes in usage patterns and availability of network resources. Network architectures can be instantly adjusted, respond to application and user requests, and services can be introduced far more quickly, easily and at a lower cost.

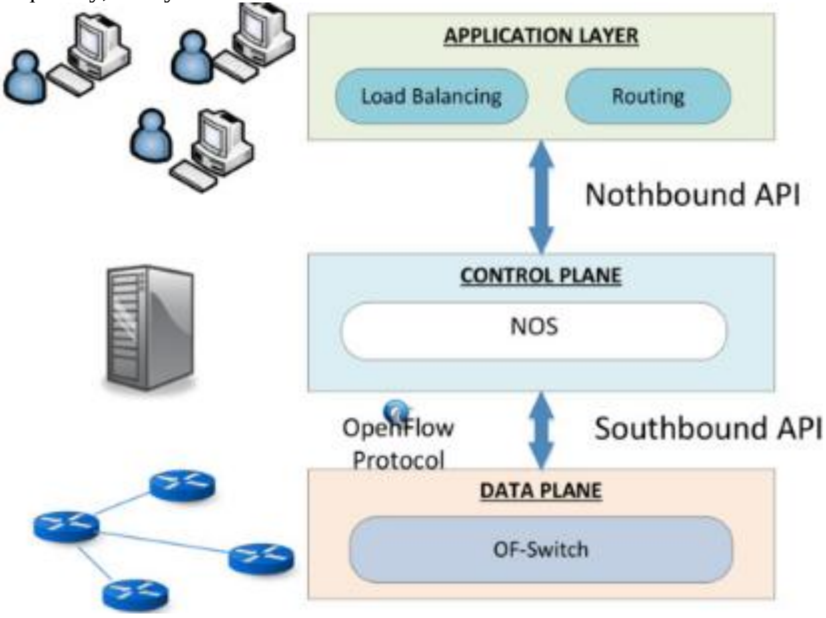

FIGURE 1.1: SDN Architecture

SDN provides separation between the control plane (controller) and data plane (switch) functions of networks using a protocol that modifies forwarding tables in network switches. This makes it possible to optimize networks on the fly and quickly respond to changes in network usage without the need for manually reconfiguring existing infrastructure or purchasing new hardware. SDN separates the control of network devices from the data they transport, and the switching software from the actual network hardware.SDN technology is currently available for industrial control applications that require extremely fast failover, called Operational Technology (OT) Software Defined Networking (SDN). OT SDN technology is an approach to manage network access control and Ethernet packet delivery on environmentally hardened hardware for critical infrastructure networks.

OpenFlow: It is a multivendor standard defined by the Open Networking Foundation (ONF) for implementing SDN in networking equipment. The OpenFlow protocol defines the interface between an OpenFlow Controller and an OpenFlow switch, see Figure 1 below. The OpenFlow protocol allows the OpenFlow Controller to instruct the OpenFlow switch on how to handle incoming data packets.
The OpenFlow switch may be programmed to:
(1) Identify and categorize packets from an ingress port based on a various packet header fields;
(2) Process the packets in various ways, including modifying the header; and,
(3) Drop or push the packets to a particular egress port or to the OpenFlow Controller.
The OpenFlow instructions transmitted from an OpenFlow Controller to an OpenFlow switch are structured as "flows". Each individual flow contains packet match fields, flow priority, various counters, packet processing instructions, flow timeouts and a cookie. The flows are organized in tables. An incoming packet may be processed by flows in multiple "pipelined" tables before exiting on an egress port. The OpenFlow protocol standard is evolving quickly with release 1.3.2 as the current revision at the time of this blog being published.
The OpenFlow Network Architecture consists of three layers:
(1) One or more OpenFlow virtual and/or physical switches;
(2) One or two OpenFlow controller(s); and,
(3) One or more OpenFlow application(s)
OpenFlow, identified by the ONF, is a protocol between SDN architecture's control and forwarding layers and is by far the most widespread SDN implementation. A basic architecture for OpenFlow consists of end hosts, a controller and switches enabled for OpenFlow. Remember that an OpenFlow switch is not limited to being a layer-2 system, contrary to the conventional nomenclature of the network. Using an Open-Flow API, the controller communicates with the switches. When a packet enters an OpenFlow switch, it handles the packets as follows:
1. A flow table search is done to match the packet's header fields to the local flow table. If there is no compatible entry, the packet will be sent for processing to the controller. When there are several entries in the flow table that suit the incoming packet, the packet with the highest priority will be picked.
2. Byte counters and packets are updated.
3. The action set is accompanied by the action(s) corresponding to the matching flow rule. If the execution chain is part of a different flow table, processing will continue.
4. The action set will be executed once all flow tables have been processed

SECURITY CLUSTER MANAGEMENT

To secure the cluster control for control plane in SDN, secure authentication is imperative when a controller joins the cluster in SDNs. As an existing security token, username/password is feasible to propose related authentication protocols. However, this security token alone does not enough security protection for the authentication. In other words, additional protections such as confidentiality, integrity, and nonreputation cannot be provided. To adapt the username/password token for this authentication, a hash function and a message authentication code (MAC) are introduced to improve security during clustering. Cluster security is an emergent property because it arises from the independent security aspects of the individual cluster nodes and is at the same time irreducible with regard to the overall cluster system. Within this work, we leverage this perspective on cluster security in order to identify various necessarily-distributed security services and their related characteristics. Our goal is to develop techniques that can be used to enhance the security of clusters that exist in domains ranging from a carrier-class telecommunications environment to a High-Performance Computing (HPC) environment

LITERATURE REVIEW

JEHAD ALI (2020) Deployment of new optimized routing rules on routers are challenging, owing to the tight coupling of the data and control planes and a lack of global topological information. Due to the distributed nature of the traditional classical internet protocol networks, the routing rules and policies are disseminated in a decentralized manner, which causes looping issues during link failure. Software-defined networking (SDN) provides programmability to the network from a central point. Consequently, the nodes or data plane devices in SDN only forward packets and the complexity of the control plane is handed over to the controller.

RAVINDRA, S.SHANKARAIAH (2020) Software-Defined Network (SDN) is regarded as one of the most significant areas for future networking. SDN architecture is a revolutionary new concept that offers more mobility, a high degree of automation and shorter delivery time by pushing the conventional network to be software-based. SDN architecture dynamically separates the control plane from the network data (forwarding) plane, providing a centralized view of the network as a whole and making it easier to manage and monitor the resources of the network. Furthermore, the SDN's initial design, with its centralized control point, does not accurately perceive the security requirements, which poses additional challenges to security issues.

DONG, S., MUDAR, S. (2019) The Distributed Denial of Service (DDoS) attack has seriously impaired network availability for decades and still there is no effective defense mechanism against it. However, the emerging Software Defined Networking (SDN) provides a new way to reconsider the defense against DDoS attacks. In this work, we propose two methods to detect the DDoS attack in SDN. One method adopts the degree of DDoS attack to identify the DDoS attack. The other method uses the improved K-Nearest Neighbors (KNN) algorithm based on Machine Learning (ML) to discover the DDoS attack. The results of the theoretical analysis and the experimental results on datasets show that our proposed methods can better detect the DDoS attack compared with other methods.

VASILEIOS GKIOULOS (2018) Software Defined Networking (SDN) is an evolving network architecture paradigm that focuses on the separation of control and data planes. SDN receives increasing attention both from academia and industry, across a multitude of application domains. In this work, we examine the current state of obtained knowledge on military SDN by conducting a systematic literature review (SLR). Through this work, we seek to evaluate the current state of the art in terms of research tracks, publications, methods, trends, and most active research areas. Accordingly, we utilize these findings for consolidating the areas of past and current research on the examined application domain, and propose directions for future research.

JESÚS ANTONIO PUENTE FERNÁNDEZ (2018) Prediction systems present some challenges on two fronts: the relation between video quality and observed session features and on the other hand, dynamics changes on the video quality. Software Defined Networks (SDN) is a new concept of network architecture that provides the separation of control plane (controller) and data plane (switches) in network devices. Due to the existence of the southbound interface, it is possible to deploy monitoring tools to obtain the network status and retrieve a statistics collection

SDN SECURITY CLASSIFICATION

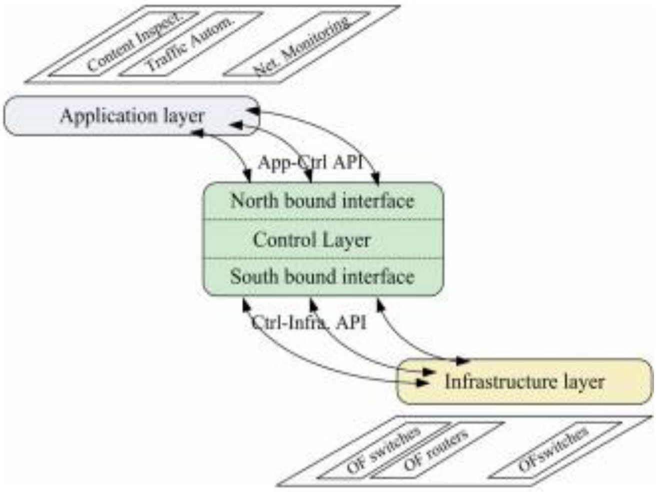

Figure 1.2: The three planes/layers in the SDN architecture

Network security systems leveraging from SDN can respond to network anomalies and spurious traffic conditions at runtime. To elaborate the functionality of the SDN architecture, the three main functional layers or SDN planes are presented in Fig. 1 and are constituted of:

- **Application Plane:** It contains SDN applications for various functionalities, such as network management, policy implementation, and security services.
- **Control Plane:** It is a logically centralized control framework that runs the NOS, maintains global view of the network, and provides hardware abstractions to SDN applications.
- **Data Plane:** It is the combination of forwarding elements used to forward traffic flows based on instructions from the control plane

Network security techniques can be implemented as applications in the SDN application plane. These applications would acquire the network state or resource information from the network control plane through the north-boundinterface (App-Ctrl API). Similarly, security applications can collect samples of packets through the control plane

HIDDEN MARKOV MODEL BASED SECURE CLUSTER MANAGEMENT

HMM is a powerful arithmetic tool for modeling instant sequence data. It is utilized for examining a generative discernible grouping that is described by some fundamental imperceptible arrangements. Markov chain is a straightforward concept that can clarify the mainly complex real-time processes.ve section says how to prepare a subsection. Just copy and paste the subsection, whenever you need it. The numbers will be automatically changes when you add new subsection. Once you paste it, change the subsection heading as per your requirement. Speech recognition, Text identifiers, Path recognition, and numerous other Artificial intelligence tools utilize this straightforward standard called the Markov chain. Markov chain depends on the rule of 'memorylessness. As such, the following phase of the procedure just relies upon the past state and not the arrangement of states. This basic suspicion makes the computation of contingent likelihood simple and empowers this calculation to be applied in various situations. All things considered, issues we, by and large, utilize the Latent Markov model, which is a much-developed variant of the Markov chain.

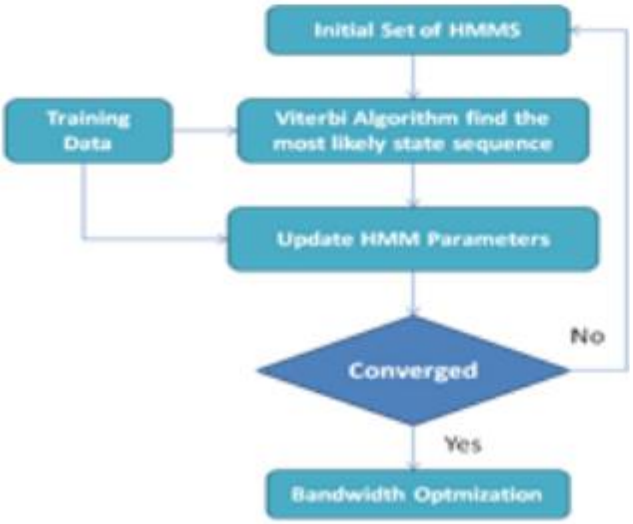

Figure 1.3: Flow of HMM algorithm

SECURITY CHALLENGES IN SDN

These are the main issues related to SDN

Forwarding Device Attack: The network traffic can be disturbed by access points or switches, which results in malicious users launching denial of service (DoS) attack that can result in network failure or disruption.

Threats in Control Plane: Due to the use of central controller, any problem arising in the network results in the failure of the central controller. The approach that is being used to solve this problem is to use either horizontal or hierarchical controller distributions.

Vulnerability of Communication Channel: SDN southbound API's such as Open Flow protocol uses TLS for data-control channel communication security but it is often disabled administratively and is prone to man-in-the middle attacks thus not suitable for implementation of channel security.

Fake Traffic Flows: A non-malicious faulty device or an attacker can launch this or DoS attack to dissipate the resources in forwarding devices or controllers.

Authenticity: It refers to the property that entities in networks are actually the ones they claim to be. The issue of authenticity for forwarding devices in SDWN networks is similar to that in traditional networks; it can result as hindrance in network performance.

Confidentiality: it prevents from the expose of information to unauthorized users, if not ensured can lead unauthorized users to access network information or data.

Availability: It means that authorized users can access data, devices, and services whenever they need.

Open Programmable API: The open nature of API makes the vulnerabilities more transparent to attackers.

Man-in-the-Middle-Monitors: The switches and the controllers are not directly connected for the transmission of information, which "man-in-the-middle" monitors can steal or misuse the information without being caught thus leading to black hole attack

CONCLUSION

SDN has changed the architecture of traditional networks separating the control and data planes removing its rigidity. In this way, SDN provides a unified view of the network providing scalability, flexibility and a centralized control. The above challenges are achieved by proposed system a secure cluster management for big data analysis for optimized control plane. A secure authentication scheme was proposed to ensure the legality of the data sources. Then Hidden markov model used to enable big data scheme and implemented system was proposed to optimized control plane. This proposed system is significant in improving the performance and efficiency of application running in SDN. Moreover, in spite of risks or issues, the security benefits in a centralized SDN framework are being exploited by research efforts, which are real-time programmability and global traffic monitoring capability.

REFERENCES

1. Alsmadi, I.: The integration of access control levels based on SDN. Int. J. High Perform. 1470 AU10 Comput. Networking (2016, accepted for publication)
2. Aleroud, A., Alsmadi, I.: Identifying DoS attacks on Software Defined Networks: a relation 1472 context approach. In: NOMS (2016)
3. Alsmadi, I., Munakami, M., Xu, D.: Model-based testing of SDN firewalls: a case study. In: 1474 Proceedings of the Second International Conference on Trustworthy Systems and Their 1475 Applications (TSA'15), Taiwan, July 2015
4. Akyildiz, I.F.; Wang, P.; Lin, S.C. SoftWater: Software-defined networking for next-generation underwater communication systems. Ad Hoc Netw. 2016, 46, 1–11
5. Bader, A.; Alouini, M.S. Mobile Ad Hoc Networks in Bandwidth-Demanding Mission-Critical Applications: Practical Implementation Insights. IEEE Access 2017, 5, 891–910
6. Dong, S., Mudar, S.: "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software defined networks." IEEE Access (2019)
7. Dang, M., Jiao, Z., Ding, H., Tian, R., Zhang, B.: "Predictive big data collection in vehicular networks: A software defined networking based approach". IEEE Global Communication and Conference, vol. 7, pp.1–6 (2016).
8. Lyu, X., Tian, H., Ni, W., Liu, R., Zhang, P.: "Adaptive centralized clustering framework for software-defined ultra-dense wireless networks". IEEE Transactions on Vehicular Technology, vol. 66, pp. 8553–8557 (2017).
9. Yan, Q., Gong, Q. and Yu, F.R.: Effective software-defined networking controller scheduling method to mitigate DDoS attacks. Electronics Letters, 53(7), pp.469-471 (2017).
10. Wang, J., Cao, J., Li, B., Lee, S., Sherratt, R.: "Bio-inspired ant colony optimization-based clustering algorithm with mobile sinks for applications in consumer home automation networks". IEEE Transactions on Consumer Electronics, vol. 61, pp. 438-444 (2015).