Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7716481

A NEW MODEL PROPOSED FOR SECURE CLUSTER MANAGEMENT IN SOFTWARE DEFINED NETWORKS

Dr. Harsh Lohiya
Research Guide, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India

Chakali Maddilety
Research Scholar, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, Madhya Pradesh, India

Abstract
Software Defined Networks (SDN) is a new way to build network architecture. It divides networking hardware into a data plane (switches) and a control plane (controllers). HMM, which stands for "Hidden Markov Model," is a new way to handle cluster administration. Because the control plane can be made more scalable, the amount of bandwidth being used is a problem. In software-defined networking (SDN), the Control plane favours a centralised solution in which a single control unit can see the whole network. In large-scale networks, the controls plane is often set up as a cluster of distributed controllers to meet the needs of management and the board. Cluster management technology in SDNs creates a lot of information because it logs and analyses a wide range of events and is needed to give a consistent global view of the health of the network. Cluster management technology is an important part of SDNs because it keeps track of all kinds of events and makes sure the network is always in the same global state. This is why a lot of information is made all the time. Because SDNs can be programmed and change over time, there is still a question about security in the cluster.

Keywords: Software Defined Networks , secure cluster management , HMM

Introduction

Software-Defined Networking (SDN) is a new and forward-thinking way to design network architecture that separates the control plane and the data plane. The separation makes SDN architecture's many layers easier to manage and handles traffic more efficiently (SDNs). In SDN, the control plane tells the data plane what to do, and the data plane then sends data across the network. The SDN controller's observations of data plane forwarding entities and other SDN agents help the network learn more about itself. Centralized control always slows down the transfer of a lot of data, even though it is clear that it makes network management better. Also, because the controller is centralised, the controller's overhead costs go up as the number of users goes up. So, the controller becomes a bottleneck in the smooth flow of service, and if the controller fails, there is no way to manage the switch it was in charge of. Since the SDN controller is in charge of all forwarding decisions, it is possible that it could act as a single point of failure. If the SDN controller goes down or if the switches stop talking to the controller, the whole network could stop working.

Centralized network designs have problems with scalability, reliability, interoperability, and being able to handle problems. SDN, on the other hand, has the benefit of being centralised while also being very flexible and easy to programme. This is a good thing about the SDN. SDNs stand out because they can be changed and changed again in the network. Also, SDNs allow a large number of SDN controllers in different parts of the world to be connected to a network and work as backup controllers in case a primary controller fails. Having many controllers also lets you spread out the load in case a single controller can't handle all the flow requests. More controllers may also help with latency, scalability, fault tolerance, and availability in an SDN deployment. But the biggest problem with this approach is that it makes it hard to keep things the same across a wide range of controllers that are spread out.

As far as we know, this is the first time that an SDN has tried to cluster the distributed controller. This study looks at the placement of the controller as well as issues like scalability, fault tolerance, and dependability. Our goal is to improve the scalability, reliability, and performance of software-defined networks (SDNs) by putting together a cluster of distributed controllers. The proposed mechanism uses a mix of commercial and well-known SDN controllers in both proactive and reactive modes to make sure that all controllers in the clusters handle the same amount of work. We did two very long experiments to find out how long our connections take and how many packets get lost.

With the proposed method, packet loss can be cut down by a lot, and the controller's central processing unit doesn't have to do much extra work. The results of the simulation make SDNs work better as a whole, making it possible to handle changes in network demand that were not expected without stopping service. Even though one controller has failed, this is still done. Some of the problems that this text starts to address are fault tolerance, scalability, interoperability, and dependability.

Related work

Sumit Badotra et al.(2021) The OpenDayLight (ODL) clusters made the centralised controller environment in a Software-Defined Network (SDN) stronger. This is because if one controller (the leader) fails or stops working, the other controllers (the followers) will take over and keep the functionality going. The ability to work in a setting with more than one controller gives the system more flexibility and freedom from outside factors. Distributed denial of service attacks, on the other hand, are smarter and more dangerous than ever before. This article shows you how to set up a three-node ODL cluster and how vulnerable it is to distributed denial-of-service (DDoS) attacks. A wide range of distributed denial of service (DDoS) traffic penetration technologies are used to send out huge amounts of traffic. Because there is so much traffic on the network, the controller eventually crashes, which stops the whole network.

Junfeng, et al. (2019) Because the Internet and mobile communication technologies are becoming more and more popular, the infrastructure, devices, and resources used in networking systems are becoming more complex and varied. It is important to use more brainpower when setting up, running, maintaining, and optimising networking systems. Because traditional networks are spread out, it is hard to use and implement machine learning techniques to control and run them. Software-defined networking makes it possible to use network-based intelligence in new ways (SDN). Software-defined networking (SDN) features like logically centralised control, a global view of the network, software-based traffic analysis, and dynamic updating of forwarding rules make it easier to use machine learning strategies. In this paper,

we look at all the work that has been done on bringing machine learning techniques to SDN so far. First, we set the scene and give the canonical texts that are needed for this discussion. Then, we'll talk about what machine learning algorithms are and how they work. We also look at how machine learning algorithms are used in software-defined networking (SDN). We focus on how they are used to classify traffic, optimise routing, predict quality of service and user experience, manage resources, and keep networks safe. In the end, we talk about the challenges and the bigger picture.

Danda, et al. (2017) Software-defined networking, also known as SDN, is a new way of networking that, unlike traditional networking, breaks down the vertical integration of the network so that it can be programmed with (logically) centralised network control. SDN can instantly change the network's parameters based on how the network is working right now. The decoupled architecture of SDN was made to make network management more flexible and less time-consuming. Network visibility, which is made possible by SDN's centralised, cost-effective architecture, is a key part of how resources are used most efficiently and how well they perform. SDN's security, energy savings, and network virtualization can be used to improve the performance of a network. This is made possible by the fact that there are more smart, programmable devices in the network. We talk about both the security problems that SDN can help solve and the new security problems that arise as a direct result of using SDN. Also, tables have been used to summarise recent attacks on SDN security and how they were stopped. We also give a complete overview of the different ways that using SDN can help reduce energy use and improve network security. We try to predict how this new paradigm will change in the future based on how research is going right now. We talk about the most important ongoing research projects, problems, and directions for research in this field. This article will help people learn more about SDN's design, different security attacks and how to defend against them, and how to make SDN use the least amount of energy possible.

Fetia, et al. (2018) The Internet was originally thought of as a complex set of box-centric protocols and vertically integrated solutions. The SDN paradigm, on the other hand, advocates for the opposite: separating the control logic from the hardware and putting it in software-based controllers. On the other hand, the Internet was made to be a complicated collection of box-centric protocols and vertically integrated systems, with decentralised control logic at its core. These basic rules make it possible for unique applications to be made and for automatic and adaptive control components to be put in place. This makes network management easier and makes the user experience better. Even though SDN is exciting, it still has some problems that need to be fixed before it can be widely used. One of these is fixing the problems with scalability and stability that plague centralised systems. The way to solve these problems is to physically spread out the control plane. But this type of organisation, which is logically centralised but physically spread out, also faces a different set of problems. This study gives a broad overview of software-defined networking (SDN), with a focus on decentralised SDN management. The most unique thing about this survey is that it looks closely at cutting-edge distributed SDN controller environments. This analysis weighs the pros and cons of different platforms and sorts them in creative ways to come up with useful ideas for SDN research and deployment (physical and logical classifications). This survey does more than just review the SDN concept and compare it to the traditional design. It also gives an overview of the SDN

architecture. There are also some details about current and possible future developments in this field, as well as a thorough analysis of the biggest problems with distributed SDN control.

Jun, et al. (2018) The software-based controller is the most important part of an SDN's abstracted control plane, which is the most important symbolic feature of the network architecture. Even though the control plane and its controllers are physically spread out across multiple nodes, they all work the same way. The control plane is often made up of clusters of distributed controllers so that it can meet the needs of service management in large-scale networks. Cluster management technology is important for SDNs because it keeps track of all kinds of events and makes sure the network is always in the same global state. This is why a lot of information is made all the time. Because SDNs can be programmed and change over time, there is still a question about security in the cluster. In this study, we try to solve these problems by giving a safe cluster management architecture for the optimal control plane that is based on big data analysis. A security and authentication system for managing clusters has been shown. We also suggest a way to make control planes work better and a way to handle large data sets using a method based on ant colonies. The simulations and comparisons showed that the proposed plan is possible and will work. Using the proposed method, security and performance of the SDN control plane can be improved in a big way.

Methodology

In this paper, we define the network as an undirected graph, G = (V, E), where V is the set of n = |V | nodes and E is the set of m = |E| edges. The data in the network is sent from one node to the next using a method called "multi-hop communication." When one node is connected to another node by an edge, the two nodes are called neighbours. The following system assumptions will be made about our cluster-based flow control method, assuming that the set of nodes V is divided into clusters C = c1,... ck,... cu with u = |C|.

• Our work is based on the idea that an SDN controller in the middle of the network divides it up (in practise this could be multiple logically centralised controllers). Each WSN node tells the SDN controller how connected its neighbours are. The SDN controller builds the topology of the WSN and divides up the network's resources.

• Our method works best with networks that are set up in a fixed way. After the WSN's topology is changed, the nodes will tell the controller about the new connections, and the controller will re-split the network into clusters based on the new topology.

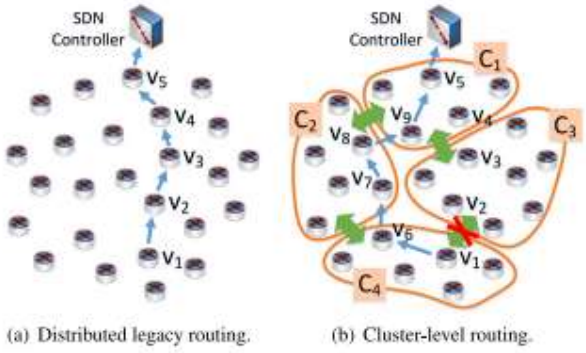
Cluster Head Nodes:

By giving their "cluster head nodes" names like h1 through hk and hu, where u is the number of clusters, we can change both the number of clusters as a whole and how they are arranged. The head node for each cluster must be located within that cluster. We require that there are no edges between any of the cluster-head nodes. This means that nodes labelled "h1," "...hk," and "...hu" are not really connected to each other.

Cluster Border Nodes:

When one of a node's neighbours also belongs to another cluster, that node is called a "border node" of the cluster it is in. Let's say that node vi is in cluster ck and that one of its neighbours is in cluster cc. The set of border nodes for cluster ck, which we'll call node set b, tells us what the cluster is.

This section is about the design of CluFlow. It gives an overview of the solution, a way to reduce the number of border nodes, and a protocol for cluster-based SDN management.



Software-defined hybrid wireless sensor network with flow control based on clusters

Conclusion

This study shows how clusters of distributed controllers can be made in SDNs (SDNs). On a set of three nodes, many controllers that are far apart have been set up for both active and reactive mode. The capacitated controller placement algorithm figures out where the group of controllers should be put. Simulations of the clustering method suggested show that it could work. When compared to a distributed controller that didn't use clustering and ran on HP Virtual Application Network (VAN) SDN and Open Network Operating System (ONOS) controllers, the results show that the proposed distributed controller clustering mechanism can significantly reduce both average latency and packet loss. The results show that the new proposal for a distributed controller that uses clustering works better than the current distributed controller that does not use clustering. It also uses the right amount of CPU resources. In the future, we plan to do more thorough studies using a wide range of commercial SDN controllers and a number of different metrics, such as the flow setup rate (throughput), the number of nodes in the cluster, and a number of other metrics. The results of this research can also be used in commercial SDN-based cloud infrastructures in a number of different data centres.

Reference

[1]. Sumit Badotra, Sarvesh Tanwar, and Ajay Rana. DDoS Penetration Testing on OpenDayLight 3-Node in Software Defined Networking [J]. Int J Performability Eng, 2021, 17(10): 866-872.

[2]. Junfeng, Xie., F., Richard, Yu., Tao, Huang., Renchao, Xie., Jiang, Liu., Chenmeng, Wang., Yunjie, Liu. (2019). A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. IEEE Communications Surveys and Tutorials, 21(1):393-430. doi: 10.1109/COMST.2018.2866942

[3]. Danda, B., Rawat., Swetha, Reddy. (2017). Software Defined Networking Architecture, Security and Energy Efficiency: A Survey. IEEE Communications Surveys and Tutorials, 19(1):325-346. doi: 10.1109/COMST.2016.2618874

[4]. Fetia, Bannour., Sami, Souihi., Abdelhamid, Mellouk. (2018). Distributed SDN Control: Survey, Taxonomy, and Challenges. IEEE Communications Surveys and Tutorials, 20(1):333-354. doi: 10.1109/COMST.2017.2782482

[5]. Jun, Wu., Mianxiong, Dong., Kaoru, Ota., Jianhua, Li., Zhitao, Guan. (2018). Big Data Analysis-Based Secure Cluster Management for Optimized Control Plane in Software-Defined Networks. IEEE Transactions on Network and Service Management, 15(1):27-38. doi: 10.1109/TNSM.2018.2799000

[6]. Rabia, Khan., Pardeep, Kumar., Dushantha, Nalin, K., Jayakody., Madhusanka, Liyanage. (2020). A Survey on Security and Privacy of 5G Technologies: Potential Solutions, Recent Advancements, and Future Directions. IEEE Communications Surveys and Tutorials, 22(1):196-248. doi: 10.1109/COMST.2019.2933899

[7]. Diego, Kreutz., Fernando, M., V., Ramos., Paulo, Veríssimo., Christian, Esteve, Rothenberg., Siamak, Azodolmolky., Steve, Uhlig. (2015). Software-Defined Networking: A Comprehensive Survey. 103(1):14-76. doi: 10.1109/JPROC.2014.2371999

[8]. Diego, Kreutz., Fernando, M., V., Ramos., Paulo, Veríssimo., Christian, Esteve, Rothenberg., Siamak, Azodolmolky., Steve, Uhlig. (2014). Software-Defined Networking: A Comprehensive Survey. arXiv: Networking and Internet Architecture,

[9]. Bo, Han., Vijay, Gopalakrishnan., Lusheng, Ji., Seungjoon, Lee. (2015). Network function virtualization: Challenges and opportunities for innovations. IEEE Communications Magazine, 53(2):90-97. doi: 10.1109/MCOM.2015.7045396

[10]. Rashid, Amin., Martin, Reisslein., Nadir, Shah. (2018). Hybrid SDN Networks: A Survey of Existing Approaches. IEEE Communications Surveys and Tutorials, 20(4):3259-3306. doi: 10.1109/COMST.2018.2837161

[11]. Alaitz, Mendiola., Jasone, Astorga., Eduardo, Jacob., Marivi, Higuero. (2017). A Survey on the Contributions of Software-Defined Networking to Traffic Engineering. IEEE Communications Surveys and Tutorials, 19(2):918-953. doi: 10.1109/COMST.2016.2633579

[12]. Tuan, A, Tang., Lotfi, Mhamdi., Des, McLernon., Syed, Ali, Raza, Zaidi., Mounir, Ghogho. (2018). Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks. 202-206. doi: 10.1109/NETSOFT.2018.8460090

[13]. Suleman, Khan., Abdullah, Gani., Ainuddin, Wahid, Abdul, Wahab., Mohsen, Guizani., Muhammad, Khurram, Khan. (2017). Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-Art. IEEE Communications Surveys and Tutorials, 19(1):303-324. doi: 10.1109/COMST.2016.2597193

[14]. Reza, Mohammadi., Reza, Javidan., Mauro, Conti. (2017). SLICOTS: An SDN-Based Lightweight Countermeasure for TCP SYN Flooding Attacks. IEEE Transactions on Network and Service Management, 14(2):487-497. doi: 10.1109/TNSM.2017.2701549

[15]. Celio, Trois., Marcos, Didonet, Del, Fabro., Luis, C., E., Bona., Magnos, Martinello. (2016). A Survey on SDN Programming Languages: Toward a Taxonomy. IEEE Communications Surveys and Tutorials, 18(4):2687-2712. doi: 10.1109/COMST.2016.2553778

[16]. Cheng, Li., Zhengrui, Qin., Ed, Novak., Qun, Li. (2017). Securing SDN Infrastructure of IoT–Fog Networks From MitM Attacks. IEEE Internet of Things Journal, 4(5):1156-1164. doi: 10.1109/JIOT.2017.2685596

[17]. Kubra, Kalkan., Levent, Altay., Gurkan, Gur., Fatih, Alagoz. (2018). JESS: Joint Entropy-Based DDoS Defense Scheme in SDN. IEEE Journal on Selected Areas in Communications, 36(10):2358-2372. doi: 10.1109/JSAC.2018.2869997

[18]. Changhoon, Yoon., Seungsoo, Lee., Heedo, Kang., Taejune, Park., Seungwon, Shin., Vinod, Yegneswaran., Phillip, Porras., Guofei, Gu. (2017). Flow Wars: Systemizing the Attack Surface and Defenses in Software-Defined Networks. IEEE ACM Transactions on Networking, 25(6):3514-3530. doi: 10.1109/TNET.2017.2748159

[19]. Taimur, Bakhshi. (2017). State of the Art and Recent Research Advances in Software Defined Networking. Wireless Communications and Mobile Computing, 2017:1-35. doi: 10.1155/2017/7191647

[20]. Christos, Tselios., Ilias, Politis., Stavros, Kotsopoulos. (2017). Enhancing SDN security for IoT-related deployments through blockchain. 303-308. doi: 10.1109/NFV-SDN.2017.8169860