2022 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) | 978-1-6654-7340-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ANTS56424.2022.10227717

Hierarchical Classic Controllers (HCC) with an Enhanced SVM Method for DDoS Attack Detection in SDN

Shideh Yavary Mehr[1] and Byrav Ramamurthy[2]

[1]Department of Computer Science, University of Wisconsin-Milwaukee
[2]School of Computing, University of Nebraska-Lincoln

***Abstract*—A centralized Software-defined Network (SDN) controller, due to its nature, faces many issues such as a single point of failure, computational complexity growth, different types of attacks, reliability challenges and scalability concerns. One of the most common fifth generation cyber-attacks is the Distributed Denial of Service (DDoS) attack. Having a single SDN controller can lead to a plethora of issues with respect to latency, computational complexity in the control plane, reachability, and scalability as the network scale increases. To address these issues, state-of-the-art approaches have investigated multiple SDN controllers in the network. The placement of these multiple controllers has drawn more attention in recent studies. In our previous work, we evaluated an Entropy-based technique and a machine learning-based Support Vector Machine (SVM) to detect DDoS using a single SDN controller. In this paper, we extend our previous work to further decrease the impact of the DDoS attacks on the SDN controller. Our new technique called Hierarchical Classic Controllers (HCC) uses SVM and Entropy methods to detect abnormal traffic which can lead to network failures caused by overwhelming a single controller. Determining the number of controllers and their best placement are major contributions in our new method. Our results show that the combination of the above three methods (HCC with SVM and Entropy), in the case of a network with 3 controllers provides greater accuracy and improves the DDoS attack detection rate to 86.12% compared to 79.03% and 81.33% using Entropy-based HCC and SVM-based HCC, respectively.**

I. Introduction

Software Defined Networking (SDN) technology is an approach to manage the network by a control plane comprising of one or more centralized controllers. The control plane of the SDN is where the intelligence resides. However, this centralization has its own drawbacks when it comes to security, scalability and elasticity [1]. The centralized controller presents a single point of failure in the SDN and potentially decreases the overall network availability. Some works have suggested incorporating multiple controllers to improve reliability, scalability and elasticity of the network. Even though using multiple controllers can bring some benefits, it also increases the network complexity. This imposes some new challenges to the network management aspects of the SDN; therefore, an efficient technique to utilize multiple controllers for security attack detection, without increasing the network complexity, is needed. Typically the controllers are logically centralized [1]; however, they may be physically distributed. While there have been earlier approaches that use multiple controllers, our approach is unique in the use of controllers with SVM and Entropy methodologies to prevent the Distributed Denial of Service (DDoS) attacks. The placement and communication of the controllers used are what differentiates this from previous works using multiple controllers. Specifically in our method, named Hierarchical Classic Controllers (HCC), the controllers follow a strict hierarchy. The top-level controller is updated by all the downstream controllers. No controller is able to communicate with its peers, it only being able to communicate with controllers that are managed by it or controllers that manage it. In other words, the information comes from all the downstream controllers to the top-level controller and is disseminated by the top level controller. This allows the top level controller to better react to changes in the network and provide a degree of isolation for other branches of the network in the event of a fault. There are several outstanding challenges regarding the use of multiple controllers in SDN, including optimal placement, allocation (how many controllers should be used) and configuration (capacity, protocol usage, etc). We have chosen the use of multiple controllers with different capacities, arranged such that we minimize the latency between controllers. This choice allows us to distribute the load on the network, utilizing the increased aggregate packet processing capabilities [2], [3], without needlessly increasing the overhead required to operate. It also allows us to have an alternate controller to fall back to in the event of unexpected outage. The workflow of our approach, HCC, is illustrated in Figure 1. In this work, we use three different topologies to evaluate our model. In the first topology (Figure 2a), HCC contains a Root Controller (RC) which is located at the top level and some Local Controllers (LCs) which are located at the bottom level and connected directly to the switches. The second topology (Figure 2b), is the same as the first topology and the only difference is having five controllers. In the third topology (Figure 2c), which is slightly larger than the first and second topologies, we employ an extra level called the "intermediate level." This level contains two Intermediate Controllers (ICs) and their responsibility is to get the information from the first level of controller and update the root controller [4]. In our approach the RC is updated at regular time intervals based

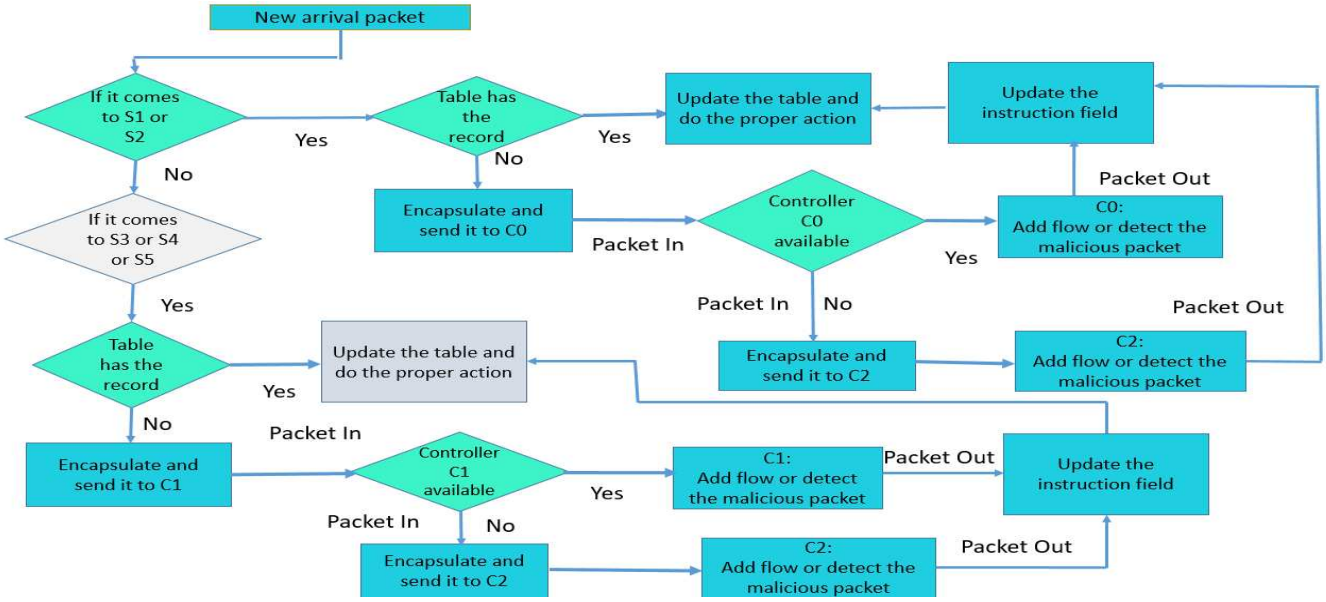
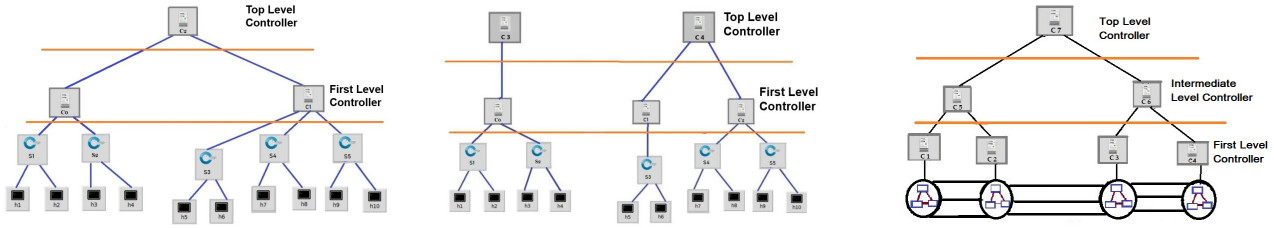


Figure 1: Flow entry process



(a) (b) (c)

Figure 2: a: Three controllers; b: Five controllers; c: Seven controllers with three levels

on the information from and the status of the ICs and LCs. When one of the LCs encounters an unknown destination, it will communicate with its respective hierarchical ICs or RCs to find a route. Our contributions can be summarized as follows:

- We study several methodologies (SVM, Entropy individually and together) to avoid DDoS attacks in an SDN network and how they can be integrated into an SDN controller.
- We propose a combination of SVM and Entropy methods on top of the Hierarchical Classic Controllers (HCC) that leverages the concept of distributed SDN controllers to provide efficient network failure recovery, availability, and robustness against DDoS attacks.
- We implement our proposed method using the Holland Computing Center and data storage named Attic at the University of Nebraska, Lincoln to show our results proving the efficacy of our proposed method.
- We implemented the best placement of the controller considering all possible combinations of choosing C controllers from among S switches. The number of the controllers should be less than or equal to the number of switches ($C \leq S$).

The rest of the paper is structured in the following order: Section II covers the background and related work. Our proposed HCC strategy is described in Section III. Section IV provides the details of the HCC with SVM and Entropy methods. Section V presents the experimental results of our proposed method. Section VI presents our conclusions and outlines our future work.

II. RELATED WORK

The authors in [1] introduced a distributed model in SDN architecture where hierarchical SDN controllers are used to avoid problems that a single controller presents in terms of complexity for the packet control. In their method, inter-domain traffic is managed by the root controller and the intra-domain traffic is managed by subordinate controllers.

Another study in [2], introduces an architecture based on hierarchical distributed controllers on SDN. Their study concentrates on different levels of controllers to decrease the latency of the system. In their work, the authors categorize the controllers into two tiers.

In [3], the authors proposed a method of DDoS attack detection implemented using random forests and support vector machines (RDF-SVM). This technique randomly selects m samples from KDD99 dataset [5] for training. In this tech-

nique, features are first extracted and then SVM is applied to re-screen the features based on their importance. This feature subset extracted was shown to not only help detect several types of attacks but also distinguish between those attacks.

A comprehensive review presented in [6] systematically reviews 70 detection and protection methods for DDoS attacks in SDN architecture. The paper also reviews various types of DDoS attacks on an SDN network. The research works reviewed in the paper are based on different information-theoretic and and Machine learning (ML) methods such as Decision Tree, Artificial Neural Networks, and SVM. The authors of the study also summarized that SDN-based architectures are relatively more secure compared to other network architectures against several types of security attacks.

In contrast to these earlier methods, our approach uses multiple synchronized controllers as standbys in order to address the vulnerabilities stemming from overwhelming a single controller. Also, to prevent DDoS attacks, HCC combined with SVM and Entropy is shown to be more reliable and accurate with a higher detection rate compared to when each method is used individually.

III. Hierarchical Classic Controllers (HCC) strategy

A centralized single SDN controller is inherently susceptible to DDoS attacks. To address this issue, we propose the idea of having multiple controllers called Hierarchical Classic Controllers (HCC) teamed with SVM and Entropy to reduce DDoS attacks and the consequent performance impacts as well as network failures. DDoS attacks consume the resources of the controller by flooding it with packets, thereby leading to exhaustion of its resources and eventually causing loss of the management over the entire SDN network. In HCC, each controller is logically centralized over its sub-network but physically distributed over the entire network [7]. Due to this peculiar property, if one controller fails, then the top-level controller steps in, effectively replacing the downed controller, preventing the attack from causing a network failure for that sub-network. In HCC, all the downstream controllers share their information and their status with the top-level controller; they do not however share information with other local controllers directly. If local controllers need to communicate, they do so through the top level controller. As an example, if C1 does not have any information about a particular packet and wants to know if the other controllers have that information, it communicates with the top-level controller. The top level controller can then determine if any of the other local controllers know anything. Should any of the controllers fail due to a DDoS attack, the top controller steps in. Since all the local controllers update the top controller in real-time, the top level controller can respond with any information about the failure. HCC integrates multiple controllers, each at a layer of the SDN, i.e. sub-network working in tandem with each other and the top-level controller. Since the top controller maintains the global view and manages all the traffic in the network, it has the highest capacity. All the downstream controllers have less capacity than those upstream. We combine HCC with Entropy, a statistical analysis method to detect any changes in the network traffic. We also employ a SVM Machine Learning solution, a popular technique for anomaly detection. Figure 2 presents some examples topology of HCC. In HCC, the first level controllers periodically update their status to the controller above them. If a controller fails to report its status for a period of time, it is assumed to be unreachable. At this instant, the controller above the unreachable controller will take over the control management responsibilities of the sub-network.

We present the flow chart describing the flow of control for our HCC methodology in Figure 1. In HCC, when a new packet arrives at a switch, the search process for finding a match into the flow table begins. If the table does not have any record match, the packet will be sent to the first-level controller. The first-level controller processes the packet and will add a new flow for similar packets. If the first-level controller does not find any match, it queries the top controller [6] which maintains a global view of all the controller tables.

A key challenge in implementing such a hierarchical controller model is that having a relatively large number of controller layers increases the complexity and hence adds inter-controller latency as we move higher into the controller hierarchy due to the large number of incoming packets. On the other hand, when there are too few controller layers, the probability of packet dropping and eventually the network failing due to DDoS attacks increases. Therefore, there is a need to optimize the number of layers in order to not impact the performance of the control plane while also addressing the occurrence of DDoS attacks. For example, in the HCC topology presented in Figure 2a, all first-level controllers are connected together via a top-level controller. The first-level controllers, C0, C1 send their local information to the top controller C2 periodically. The top-level controller maintains a global view of the network and updates the status of the all controllers that have recently reported to it as “alive”. In the next section, we explain how to combine HCC with Entropy and SVM to protect the network from DDoS attack.

IV. HCC with SVM and Adaptive Entropy-based Detection Mechanism

We combined the techniques of Entropy and SVM methods for DDoS attack detection and integrated them into HCC. This is a two-step method where the Entropy-based DDoS attack detection mechanism runs on the incoming packet flows and determines the probability of that flow being an attack or not. At this stage, the output of the Entropy-based DDoS detection is fed into a pre-trained SVM-based DDoS attack detection mechanism for the purpose of training the SVM model. The main motivation of such an ensemble-like detection mechanism is that it leverages the simplicity of Entropy-based method combined with the advantages of the SVM-based method such as accuracy, faster response time and ease of adaptability.

A. HCC with Entropy-based Detection Mechanism

Some of the effective methods to detect DDoS attacks are Entropy-based mechanisms. Entropy is a quantitative measure of randomness associated with a random variable. In this algorithm (see Algorithm 1), we consider a window size equal to 50 to collect packets for Entropy analysis. The packets are classified based on their destination IP address. A hash table from each window contains two fields, namely IP addresses, and the number of times it has occurred. The probability can be calculated for each unique destination IP address via the formula in line 4 of the algorithm. The entropy of network can be calculated using the formula in line 12 of the algorithm. Window size and threshold are two main components to DDoS detection using entropy method. Window size is determined either by the time period or based on the number of packets. The threshold value of entropy for certain characteristics of the data is used to detect suspicious packets. In our method, we monitor the entropy of the traffic patterns in the SDN at each of the controllers to identify any suspicious flow of packets that could be a potential DDoS attack. The entropy is calculated using a sliding window over the incoming packets at the controller [8]. Assuming a certain probability for each node under the controller to be the forwarded destination for an incoming packet, we compute the randomness and compare the computed value against the threshold to deem whether this flow of packets might potentially be a DDoS attack.

The window size and threshold value can be flexibly defined as a time unit or the number of packets. Additionally, window size and threshold values are adaptive i.e. they are updated with the run time dynamics of the incoming network traffic. We describe our adaptive Entropy-based mechanism in Algorithm 1. Our topology shown in Figure 2a consists of 10 hosts and 3 controllers across two control layers. In an ideal scenario, the probability of receiving new packets on each of the hosts are reasonably close and the entropy is assumed to be a maximum value following the principle of maximum entropy [9]. When any of the hosts in the network receives an unusually high number of packets, the entropy at that host is assumed to be low and this situation is deemed as an outlier.

Upper and lower threshold values are established for what the entropy should be; a value above or below the respective threshold is considered an attack. In this work, a predetermined threshold is used to compare with the computed entropy value to detect the attack. However, since SDN is a programmable architecture, the threshold value or window size at each controller can be adjusted on-the-fly based on the run-time network traffic behavior. We could further consider the network structure or IP headers (such as destination IP address, destination port, etc.) when computing the entropy, allowing for a highly customized solution.

The process of using entropy is shown in Algorithm 1. The new packet-in arrives with a new source IP address. The destination IP address is inspected for any existing instance in the window. If it can be found, then the counter is incremented. In case the window becomes full, the entropy is calculated and it will be compared with the set threshold value to identify an attack.

Algorithm 1 Detection process using entropy

1: **procedure** INITIALIZE
2: $num\_packets \leftarrow$ total packets in the current time interval $\triangle t_{current}$
3: $X = x_1, x_2, x_3, ..., x_n$, where $x_i \forall\ i \in [1, n]$.
4: $P_i = \frac{X_i}{n}$
5: P_i : The probability of i^{th} destination IP.
6: X_i : The packet count on i^{th} destination IP.
7: n = Total number of destination IPs.
8: $num\_attacks_{s_j} \leftarrow$ counter for DDoS attacks at switch s_j
9: **end procedure**
10: **procedure** COLLECT STATISTICS
11: Calculate entropy of j^{th} switch (H_{s_j}).
12: $H_{(sj)} = -\sum_{i=1}^{n} P_i log\ P_i$; H_{s_j}
13: **end procedure**
14: **procedure** ENTROPY COMPUTATION
15: **if** $H_{s_j} > th_{entropy}$ **then**
16: $num\_attacks_{s_j}++$
17: **if** $num\_attacks_{s_j} >$ minimum DDoS attacks detected in W **then**
18: DDoS attack detected!
19: **end if**
20: **end if**
21: **end procedure**

B. HCC with SVM-based Detection Mechanism

We designed and implemented an SVM-based detection mechanism that adapts to the changing run-time behavior of the incoming traffic at the controllers. The training data for the SVM is sampled from the run-time traffic in the network. Due to the presence of a large number of features in the training data set, we use Linear Kernel Function for feature reduction [10]. A Regularization parameter C is used to set the threshold for misclassification of each training example. In other words, larger values of C leads the SVM optimization to the smaller-margin hyper-plane [11]. In contrast, a small value of C leads to a larger-margin separating hyper-plane. Once the training stage of SVM finishes, the information gained from training is used to classify the future incoming traffic on the likelihood of being a DDoS attack. Our SVM model continually adapts to the incoming traffic by periodic repeated training process. We describe our SVM-based detection mechanism in Algorithm 2.

V. SIMULATION AND RESULTS

We use the NSL-KDD data for the training and testing of our models [5]. In this paper, we consider TCP network connections and select both normal and DDoS attack data. Our data set contains both training and testing data. Packet header processing and collecting statistics are the two main parts of detection. In our model, the traffic data is collected

from packet-in messages. Then, we extract salient features from the incoming packets where source and destination IP addresses and source and destination ports are some of the key features. The distribution of each feature is measured by the Entropy method. We then trained our models over normal and malicious traffic data. Our experimental results show that in the case of a network with 3 controllers, an Entropy enhanced SVM model with HCC provides more accuracy, increased detection rate, and lower response time compared to HCC + SVM and HCC + Entropy approaches. For the sake of brevity, we chose to only include the results for 1, 3, and 5 controllers.

A. Implementation

Our experiment was conducted in the Holland Computing Center, hosted at the University of Nebraska, Lincoln. We simulated the NSL-KDD data set over two experimental topologies. One experimental topology consists of three SDN controllers (ONOS) in two control layers 2a while the other consists of five SDN controllers. ONOS controllers were used with 64GB memory for the local and top controllers. The data set from the switches will forward the header for any packets received and will buffer the payload. In our example topology, the network contains of three layers, the switches and hosts are in layer 0, a cluster of ONOS [12] controllers are in layer 1 and layer 2 includes the top controller. In SDN, the incoming packets will not go to the top controller unless the match cannot be found. In addition, the SDN controller does not process all traffic, just new unmatched packets. Hence to reduce the latency in the system we should connect a limited number of switches and hosts to each controller. An abnormally high amount of traffic due to an DDoS attack induces additional delays at the switches and consequently decreases the network performance [13]. The detection solution for any DDoS is based on these abnormalities in the network. To find the anomalies, we consider the characteristics of the data in the network such as the delay, packet size and packet header information. We calculate the entropy using two parameters, window size and threshold. A window size of 50 chose in this work to compute the entropy for incoming packets. We chose a relatively small window size of 50, rather than 500 or 5000, to reduce the computational requirements and therefore detect attacks earlier. The entropy value will then be compared to a threshold value and used to determine if a DDoS attack was launched. We used the following expression to calculate the DDoS attack detection rate for each of the methods proposed in this paper:

$$Detection\_Rate = \frac{\#malicious\_pkts}{\#malicious\_pkts + \#misclassified\_pkts} * 100 \quad (1)$$

$$Accuracy = \frac{\#ofpredict\_attacks}{\#oftotal\_attacks} * 100 \quad (2)$$

Algorithm 2 The traffic distinguishing process

procedure INITIALIZE
  $C \leftarrow$ regularization parameter
  $X = x_1, x_2, x_3, ..., x_n$, where $x_i \forall\ i \in [1, n]$ and x_i indicates a TCP connection with some features which are the host-based and time-based network traffic.
  $C > 0$; $c = 1$ was chosen in this work.
  K(x,y,C) is a Linear Kernel Function
  $\alpha^*$is the Lagrange multiplier vector defined as $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_N^*)^T$
end procedure
procedure SVM METHOD
  Select a positive component (b^*) from $\alpha^*(0 \leq \alpha_i^*)$
  Compute the function:
  $b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i k(x_i.x_j)$
  Compute the decision function:
  $$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i^* y_i K(x, x_i, C) + b^*\right)$$
  where $0 \leq \alpha_i \leq C, i = 1, 2, ..., N$
  if f(x) = -1 **then**
    DDoS attack detected!
  else
    Normal packet!
  end if
end procedure
procedure *K(x,y,C)*
  return $X^T.Y + c$
end procedure

B. Results and Discussion

Our results show that in the case of a network with 3 controllers, when HCC is used with Entropy-based and SVM-based detection methods, it improves the DDoS attack detection rate to 86.12% compared with 79.03% and 81.33% using Entropy-based HCC and SVM-based HCC methods, respectively. Figures 3a, 3b and 3c show that the average response time for 10 hosts on the HCC + SVM + Entropy model gives a lower response time as compared to SVM and entropy for all 10 hosts. As is seen in Figure 3b, there is a better response for all the methods when we use the appropriate number of controllers. Figure 3d, 3e and 3f show that our proposed method has a lower rate of dropped packets due to a direct improvement in attack detection. Figures 3d, 3e and 3f show the comparison between entropy-based, SVM-based, and entropy-enhanced SVM-based detection methods with HCC over a SDN topology of 1, 3 and 5 SDN controllers respectively. Entropy-enhanced SVM with HCC method experiences the least packet loss over 1, 3 and 5 SDN controllers. Figures 3d, 3e and 3f also show that an SDN network with 3 controllers presents an optimal scenario for packet loss rate and hence the network performance. This is because it strikes a fine balance – a large number of controller layers increases the complexity and latency, while the probability of packet loss due to DDoS attacks increases when there are too few controller layers, eventually leading to network failure. Through our experiments, we found that having two controller layers is optimal for our topology.

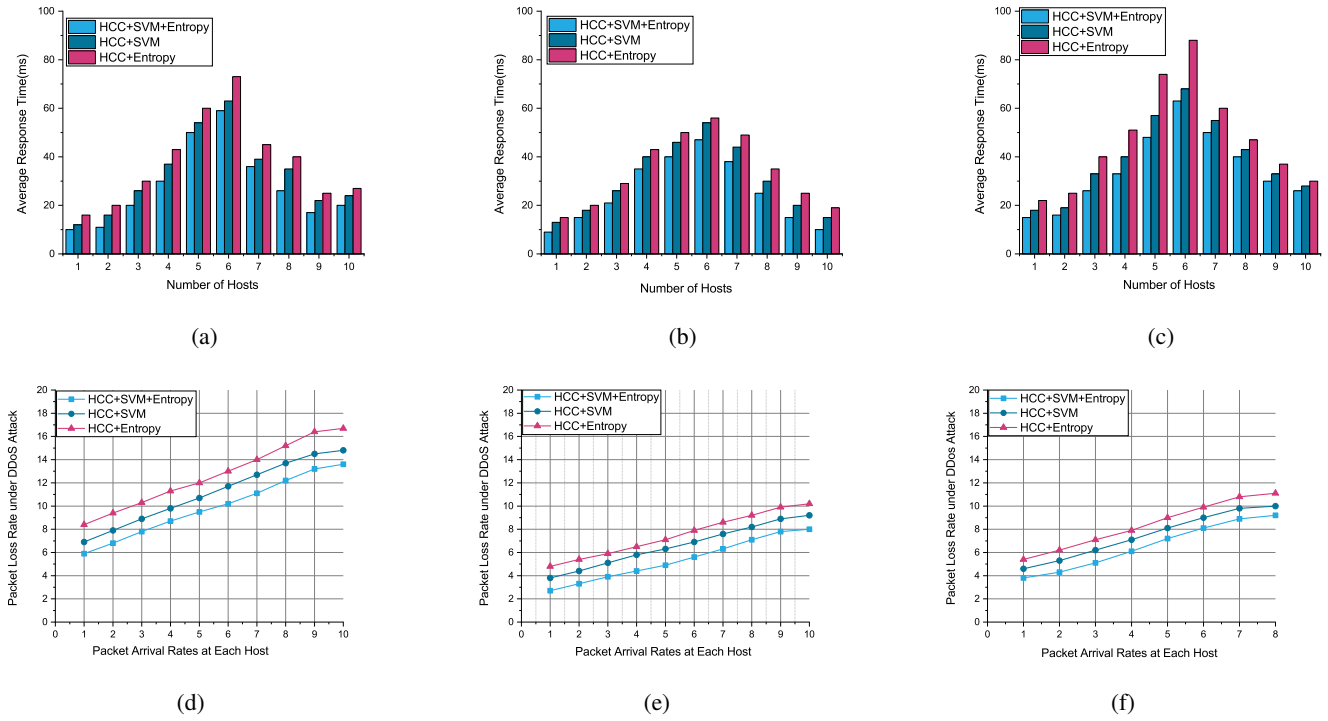


Figure 3: Sub-figures a, b, c: Response time versus number of hosts with one/three/five controller(s). Sub-figures d, e, f: Packet loss versus packet arrival rates at each host using one/three/five controller(s).

VI. Conclusions and Future Work

Single and distributed controllers are the two types of the control plane implementations over an SDN architecture. Our proposed method Hierarchical Classical Controllers (HCC), is a type of Distributed SDN controller model that improves the DDoS attack detection accuracy compared to other methods. Using entropy and an SVM-based algorithms we presented our proposed method, Hierarchical Classical Controllers (HCC). We used an entropy monitoring algorithm and an SVM-based method for DDoS attack detection with HCC and show that HCC improves the average response time and detection accuracy. These gains directly lead to reduction in the packet loss. Additionally, we analyzed the impact of the number of controller layers on the attack detection accuracy and network performance. The number of controller layers was deemed to be an important factor for not only improving the attack detection but also for reducing the network complexity. In future work, we will focus on finding an optimal machine learning algorithm for detecting DDoS attacks even earlier. Additionally, we will explore deep learning methods for attack mitigation following the detection.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation grant (NSF CNS-1817105).

References

[1] E. Amiri, E. Alizadeh, and K. Raeisi, "An efficient hierarchical distributed SDN controller model," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019, pp. 553–557.

[2] K. S. Kalupahana Liyanage, M. Ma, and P. H. Joo Chong, "Controller placement optimization in hierarchical distributed software defined vehicular networks," *Computer Networks*, vol. 135, pp. 226–239, 2018.

[3] C. Wang, J. Zheng, and X. Li, "Research on DDoS attacks detection based on RDF-SVM," in *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2017, pp. 161–165.

[4] A. Koshibe, A. Baid, and I. Seskar, "Towards distributed hierarchical SDN control plane," in *2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)*, 2014, pp. 1–5.

[5] Canadian Institute for Cyber security, http://nsl.cs.unb.ca/NSL-KDD/, 2019.

[6] L. Mao, https://leimao.github.io/blog/Maximum-Entropy/, 2021.

[7] Y. Chen, J. Pei, and D. Li, "Detpro: A high-efficiency and low-latency system against ddos attacks in sdn based on decision tree," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[8] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *2015 international conference on computing, networking and communications (ICNC)*. IEEE, 2015, pp. 77–81.

[9] M. Conti, A. Gangwal, and M. S. Gaur, "A comprehensive and effective mechanism for DDoS detection in SDN," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2017, pp. 1–8.

[10] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks," *IEEE access*, vol. 6, pp. 44 570–44 579, 2018.

[11] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, p. 100279, 2020.

[12] R. Li and B. Wu, "Early detection of DDoS based on φ-entropy in SDN networks," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, 2020, pp. 731–735.

[13] C.-J. Ong, S. Shao, and J. Yang, "An improved algorithm for the solution of the regularization path of support vector machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 3, pp. 451–462, 2010.