VOL. 12, NO. 11, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SBS-SDN BASED SOLUTION FOR PREVENTING DDoS ATTACK IN CLOUD COMPUTING ENVIRONMENT

T. V. Sindia and Julia Punitha Malar Dhas
Department of Computer Science and Engineering, Noorul Islam University, Kumaracoil, India
E-Mail: sindia.niu@gmail.com

ABSTRACT

Cloud computing is an emerging technology that provides several services to the cloud users. However, the adversaries intend to make certain services unavailable by triggering DDoS attacks. Hence, a system to prevent or detect DDoS attack is the need of the hour. This paper proposes a SDN based solution to prevent DDoS attack in the cloud computing environment. In order to achieve the goal, the proposed approach is decomposed into knowledge gaining and operational phase. The knowledge gaining phase intends to extract features from the samples and the entropy of the destination address is computed. The variance of the entropy is computed and the feature representatives are formed. The median of the feature representatives is calculated and is fixed as threshold. In the operational phase, the system can classify between the normal and the DDoS attack scenarios by computing Euclidean distance between the test sample feature representative and the knowledge base. The performance of the proposed approach is found to be satisfactory in terms of detection rate and detection time. Additionally, the proposed approach shows the least FNR and FPR, when compared to the existing techniques.

Keywords: cloud computing, DDoS attack, SDN.

1. INTRODUCTION

Cloud computing is a boon to the today's business world. Cloud computing provides a range of services on the basis of software, platform and infrastructure. Most of the companies started to exploit cloud services by paying for the usage of the service. For instance, banks, healthcare organizations and many other sectors utilize cloud services from the cloud service providers such as Amazon, Microsoft, IBM and so on. This distributed environment is prone to several security attacks. Among all sorts of attacks Distributed Denial of Service (DDoS) attack is more popular and is the distributed version of the Denial of Service (DoS) attacks. The DDoS attack makes the service unavailable to the corresponding users by depleting the entire system or a particular source.

For instance, the DDoS attack is observed in the Olympics site, which has suffered from almost 540 Gbps. The DDoS attack is executed by unknown hackers and these hackers are destined to halt the normal functionality of the system. To launch this attack, the hackers forward excessive amount of traffic to the server either manually or by an automated system. At this juncture, the server cannot handle the superfluous traffic and it arrests the service for legitimate users also. The DoS attack is carried out by a single system. On the other hand, the DDoS attack is accomplished by several systems, which are distributed across several geographical locations. DDoS attacks are the serious security threats to the cloud services at present [1-4].

Software Defined Networking (SDN) is a popular networking paradigm, which intends to manage the dynamic functionality of the future networking generations, while reducing the computational cost [6, 7]. The prime advantages of SDN are it isolates the control logic from the networking equipments and the switches involve no implicit computations. This implies that the operational logic is embedded in the SDN controller [8]. This way of functionality partitioning results in better performance and reduces the computational complexity involved.

An Application Programming Interface (API) acts as an intermediary between the SDN controller and the networking equipments. One of the famous APIs for SDN is Openflow [9]. The API enables the controller to communicate with the networking equipments and vice versa. As the operational logic and the physical equipments are segregated, it is easy for the system to control and manage the networking activity without any hassles.

Though there are several solutions in the existing literature to deal with DDoS attacks, the SDN based solutions for DDoS attack in cloud computing are countable. Understanding the efficiency and the need for versatile SDN based solutions for DDos attack in cloud computing, this paper proposes a new networking framework that can prevent the DDoS attack by taking the traffic data into account.

The features of the traffic data are continuously tracked by the network switches and are passed to the controller for every period of time. The controller measures the data flow in terms of count and size. Based on the opinion of the controller, the request is processed or blocked, so as to safeguard the system from DDoS attacks. In order to achieve the goal, this work utilizes the correlation features of the traffic data, to distinguish between the normal and the attack scenario.

The reason for the choice of correlation features is that these features are observed to be effective for the system to classify between the normal and the abnormal traffic. However, the correlation features do not rely on the basic correlation measures such as Kendall or Pearson but depend on variance of the entropy being calculated

between the features. Some of the highlights of this work are listed below:

- The incorporation of SDN makes the attack detection and management processes easier.
- The computational complexity involved in the attack detection is overthrown.
- An SDN based solution that is proven to be accurate and efficient is proposed.
- The proposed approach learns about normal/abnormal traffic and applies the gained knowledge to preventing the DDoS attack.
- This work ensures the safety of the system, as the DDoS attack is prevented.

The rest of the paper is organized as follows. The review of related literature is presented in section 2. Section 3 elaborates the working principle of the proposed approach along with the overall idea. The performance of the proposed approach is analysed in section 4. Finally, the conclusions of the proposed work are presented in section 5.

2. REVIEW OF LITERATURE

This section intends to present the related review of literature with respect to DDoS attack and SDN. The main objective of this work is to prevent DDoS attacks rather than to detect and manage the attack. Although there are several systems to address DDoS attacks, the SDN based solutions are deficient. Taking this as a challenge, this work presents a SDN based solution for DDoS attack prevention.

Some of the popular ways to prevent DDoS attacks are activity profiling and source ID distribution [10-12]. However, these solutions depend on the nature of the DDoS attacks and suitable for specific DDoS attacks. Hence these solutions cannot prevent the DDoS attacks in a successful manner. Several works based on Self-Organising Maps (SOM) have been proposed to defend against DDoS attacks [13, 14]. The major drawback of this technique is its maximum time consumption. In [15], a covariance based attack detection model is presented. However, this model is found to be suitable on SYN flooding DDoS attacks alone. In [16], a proactive DDoS attack detection technique is proposed and is suitable for bandwidth DDoS attack. In this work, the IP addresses are utilized to track the network traffic and so the attacks are detected earlier.

Openflow is the state-of-the-art protocol that is based on SDN paradigm [17, 18]. This protocol possesses several internal flow tables which are controlled and monitored by external controller. Every flow table consists of multiple traffic flow entries and all the flow entries have a header. Besides this, a column is dedicated for the action to be performed on the flow entries, which makes sense that the node has to be served or blocked.

All the packets that reach the switch are compared against the flow tables. In case, if the packet matches with any entry, then the action which has been taken already is repeated. In case of the absence of the matching entry, the packet is passed to the controller, such that the controller can take necessary actions. Afterwards, the controller updates the switch with the updates. Today's research focuses on developing an efficient controller, which is capable of managing the switch. Several controllers have already been presented already in the literature. The central theme of the controllers is to manage the entire network and to take necessary actions when something goes wrong. The controllers can make decisions by applying the previously gained knowledge.

Motivated by the existing works and SDN, this paper intends to present a SDN based solution for preventing DDoS attacks for cloud computing. This work takes the features of the packets into account and computes the multi-variational correlation among the features. Finally, the normal and the abnormal features are classified by the system and necessary actions are taken. Based on the controller's decision, the request is served or blocked. This results in continuous operation of the system along with improved performance. The following section elaborates the working principle of the proposed approach.

3. SDN BASED SOLUTION FOR PREVENTING DDoS ATTACK

DDoS attack is one of the crucial challenges placed in front of cloud computing. As cloud computing follows distributed environment, DDoS hit the scene and is unavoidable. Though the motives of the DDoS attack are different, the primary objective is to hinder the operation of the cloud by forwarding so many packets in a repetitive fashion. At a point, the server cannot handle too many requests, which results in system crash or the system becomes unavailable. It is difficult to resume the normal mode of the server. Instead of handling the attack after its occurrence, it is better to prevent the attack by imparting some knowledge to the system. Taking these points into account, this work intends to prevent the DDoS attack by feeding knowledge to the controller. This makes the controller to operate efficiently by comparing the current scenario with the knowledge base.

3.1 Overall flow of the work

The objective of the article is to present a SDN based solution for DDoS attack prevention in cloud computing. The proposed approach is decomposed into two categories, which are training and testing stage. The goal of the training phase is to impart knowledge to the controller, such that it can classify between the normal and abnormal requests. As soon as the training phase is done, the knowledge obtained is stored in the knowledge base and the system is now ready to handle any type of request. The knowledge is imparted to the system by considering the features of the request. The overall workflow is depicted in Figure-1.

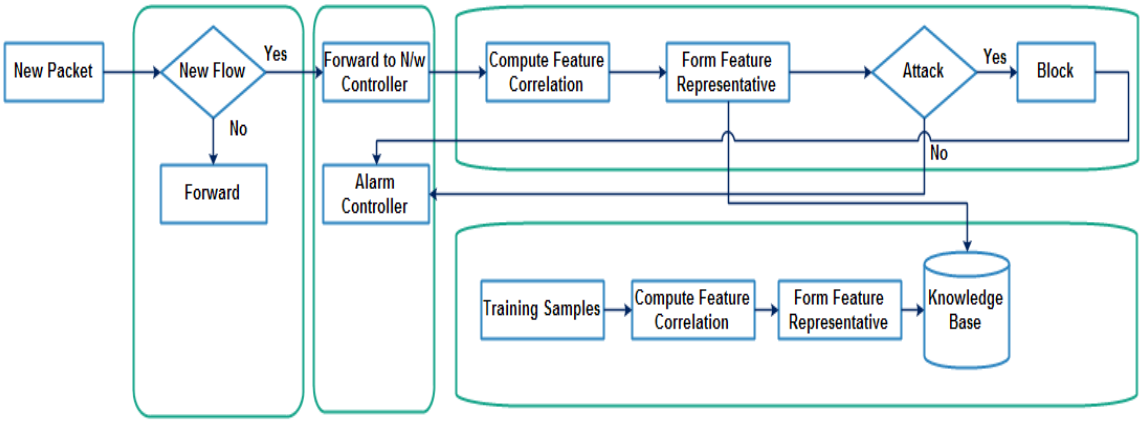

Figure-1. Overall flow of the work.

In the training phase, the correlation between the features of the entities is found on the basis of multi-variational analysis. The entropy and the variance of the entropy are calculated to form the feature representative and for every feature vector the mean and the deviation are computed. The median of the calculated values is computed and is considered as the threshold.

In the testing phase, the same procedure is followed, but finally the Euclidean distance is employed to find the degree of similarity between the testing entity and the knowledge base. By this way, the controller is imparted with the knowledge, through which the normal and the abnormal input can be classified. Based on the nature of the entity, the actions are taken by the controller, which can either be serve or block. This approach provides better solution to the DDoS attack prevention. The main merits of this work are reduced time consumption, computational complexity along with better work performance.

3.2 Stages involved in the proposed approach

The proposed approach consists of two major phases, which are knowledge gaining phase and operational phase. The knowledge gaining phase intends to impart knowledge to the system, such that it can differentiate between the normal and abnormalities. The operational phase is meant of real operation of the system. The knowledge gaining phase is achieved by several features of multiple entities. Though there are several features, the system requires some necessary features to gain knowledge. The feature representative is formed by computing the entropy and variance of the destination address along with the rate of packet.

The entropy is one of the standard measures, which is dynamic by nature. Here, the value of entropy is found to be high, whenever the packet is destined for various hosts. On the other hand, the entropy value falls down in the case of numerous packets transmission to a same host. This work utilizes the destination address to compute the entropy, as the attackers can spoof the source IP address. This is followed by the computation of variance of entropy, which denotes the rate of variation in the destination address for a time interval. The rate of packet is the total count of packets being transmitted in a particular span of time.

3.2.1 Knowledge gaining phase

As mentioned earlier, this section aims to collect some samples from the dataset for computing the entropy and variance of entropy of the destination address along with the packet count. The obtained values help the system to differentiate between the normal and the abnormal scenario. The entropy of the destination address is computed as follows.

The value of entropy (Ent_{da}) presents in between 0 and $log_2 cd$. Here, cd is the total count of the destination address. The value of Ent_{da} is observed to maximum, when the packets are intended to be transmitted to different destinations. However, the smaller value of Ent_{da} indicates that all the packets are intended to be transmitted to the same destination. The entropy value depends on the occurrence frequency of a particular destination address over a specific interval of time. For instance, the destination address may seem to be repeated, however with different source address. This entropy measure aims to compute the unique destination addresses for a particular period of time. Consider a be the total count of destination addresses for b number of packets. The occurrence frequency of a particular destination address is computed by the following equation.

$$OF_i = \frac{b_i}{b} \quad (1)$$

Where, b_i is the count of packets with the destination address DA_i. Based on (1), the entropy of the destination address is computed by

$$Ent_{da} = -\sum_{i=1}^{a} OF_i \log_2 OF_i \quad (2)$$

The greater value of Ent_{da} indicates that the packets are intended to be transmitted to different destination addresses. After the computation of entropy of the destination address, the variance of the entropy is calculated. Let $a1$ be the count of entropy values being

considered and m_{ent} be the mean value of the entropy. The variance is computed by

$$var_{ent} = \frac{\sum_{i=1}^{a}(Ent_{da} - m_{ent})^2}{a1} \quad (3)$$

When the variance value is greater, then it indicates that the destination address changes frequently. Finally, the packet rate pkt_r is also taken into account for forming the feature representative. The feature representative is formed by

$$f_{rep} = (Ent_{da}, var_{ent}, pkt_r) \quad (4)$$

When the feature representatives are formed, the median of each and every feature is calculated to find the threshold. This threshold value plays an important role in differentiating between the normal and the abnormal scenarios. Thus, the system is trained and the knowledge gained so far is stored in the knowledge base.

3.2.2 Operational phase

This phase work on the basis of the knowledge gained from the previous phase. As soon as the samples are passed, the feature representative is formed. The formation of feature representative is achieved by following the same way that involves entropy and variance calculation. The feature representative of the test samples are formed, as per eqn.4. The feature representative so formed is compared with the knowledge base by computing the Euclidean distance. Based on this, the test samples are classified as normal and abnormal scenarios. In addition to this, whenever the feature representative of the test sample is smaller than the threshold value, then an alarm is generated. The overall algorithm for the proposed approach is presented below.

SDN based Solution for DDoS Attack Prevention – Algorithm

Input: Network traffic entites $e_1, e_2, \ldots e_n$
Output: DDoS attack detection
//Knowledge Gaining Phase
Begin
Extract features from the network traffic entities;
For $(i = e_1 to\ e_n)$
do
Compute entropy Ent_{da};
Compute variance of the entropy var_{ent};
Extract packet rate pkt_r;
Form feature representative $f_{rep} = (Ent_{da}, var_{ent}, pkt_r)$;
End
For $(f_{rep} = 1\ to\ n)$
Compute the median and store it as threshold thr;
Compute max and min f_{rep} and store;
End;
End;
// Operational Phase
Extract features from the network traffic entities;
For $(i = e_1 to\ e_m)$
do
Compute entropy Ent_{da};
Compute variance of the entropy var_{ent};
Extract packet rate pkt_r;
Form feature representative $f_{rep} = (Ent_{da}, var_{ent}, pkt_r)$;
End
For $(f_{rep} = 1\ to\ m)$
Compute the Euclidean distance between the sample and the knowledge base;
If $(dist \geq thr)$
Declare DDoS attack;
Else
Declare normal;
End;
End;

Whenever the controller sniffs that something is going out of control or when an abnormal scenario is detected, the requests are immediately blocked. Thus instead of handling the attack, this work prevents the attack by imparting knowledge to the controller. This work is observed to be uncomplicated and involves minimal computation overhead. The upcoming section presents the experimental analysis of the proposed approach.

4. EXPERIMENTAL ANALYSIS

The simulation is carried out on a stand alone machine with Intel i7 processor, 1.80 GHZ and 16 GB RAM. The experiments are carried out in Java by utilizing publicly available dataset namely CAIDA [23]. This dataset contains unknown traffic traces from a DDoS attack in the year 2007. The performance of the proposed approach is evaluated against three existing techniques, which works by cluster analysis, wavelets and location separation [24-26]. The performance is analysed in terms of detection rate, time to detect attacks, true positive and false positive rates.

The detection rate of any attack detection system must be as great as possible. This tightens the security of the system, as a good attack detection system acts as a wall to the entire system. The time consumption for detecting DDoS attacks must be relatively low. However, achieving lesser time consumption is a bit difficult thing. When the DDoS attack detection system involves more computation, then obviously the time consumption goes high. Hence, it is always better to use lightweight but effective features to minimize the time consumption.

It is important for all the DDoS attack systems to show the least False Positive (FPR) and False Negative Rates (FNR). When the FPR and FNR go high, the system fires unnecessary alarms and this halts the operation of the system. Besides this, when the FPR and FNR are more, then the computational efficiency becomes poor. All the above mentioned measures are utilized to check the performance of the proposed work and the results obtained are compared with the existing works.

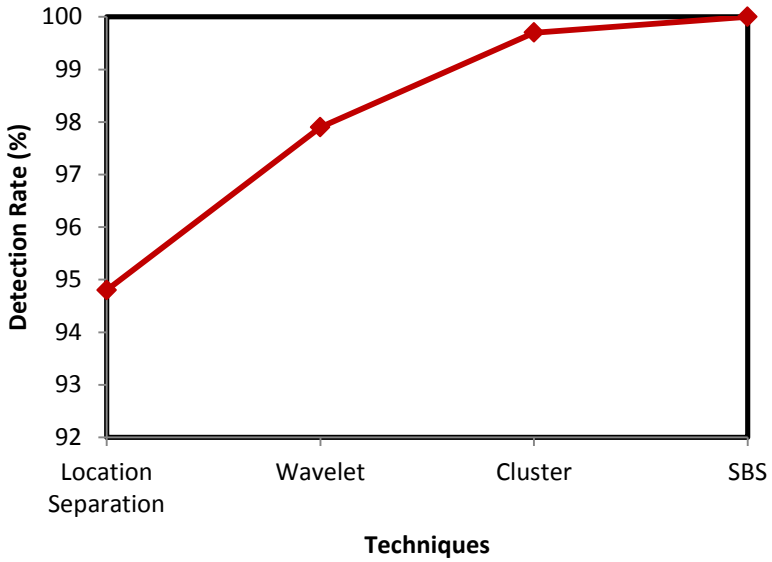

Figure-2. Detection rate analysis.

The detection rate of the proposed work is analysed and compared with the existing analogous techniques. The location separation technique separates the identifiers from the locators and the nodes IDs are used by the network layer for node placement. By this way, the DDoS attack is detected. As the location separation technique is originally proposed to address Internet scalability issues, it cannot serve well for DDoS attack detection. In the wavelet based work, the wavelet based features are extracted and semi-supervised learning is applied to detect the DDoS attack. The wavelet feature based DDoS detection system cannot serve well, owing to the increased computational complexity. The cluster analysis based DDoS attack detection technique utilizes the centroid based rules. This work claimed that the approach is suitable for real time applications, as it raises minimal false alarm rates. On comparison of the detection rate, the proposed work shows cent percent attack detection. The FPR and FNR are analysed and the results are presented in Figure-3.

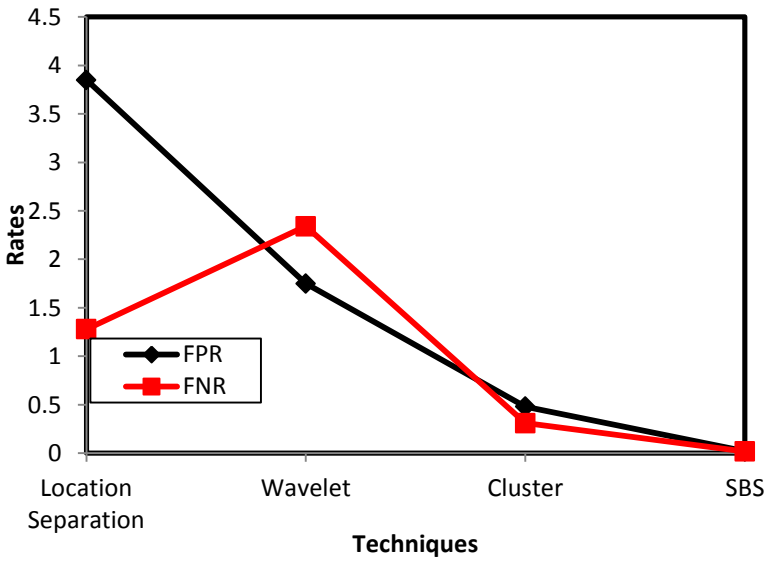

Figure-3. Analysis on FPR and FNR.

An ideal attack detection system must show lesser false alarms. False Positive Rate (FPR) is the state that claims a normal traffic as a DDoS attack. False Negative Rate (FNR) is the measure that interprets the original DDoS attack as normal. Both these situations are equally harmful, as the FPR consumer unnecessary computational power, time consumption. Besides this, FPR interrupts the normal work process of the system. FNR interprets the attack as normal, which is dangerous. This FNR broadens the loophole of the system security.

The purpose of DDoS attack detection system fails, in case of more FNR and the system is claimed to be inefficient. On analysis, the FPR and FNR of the proposed approach is the least, when compared to all other existing techniques. The FPR and FNR shown by the proposed approach are 0.02 and 0.018 respectively. The experimental results on detection time analysis are shown in Figure-4.

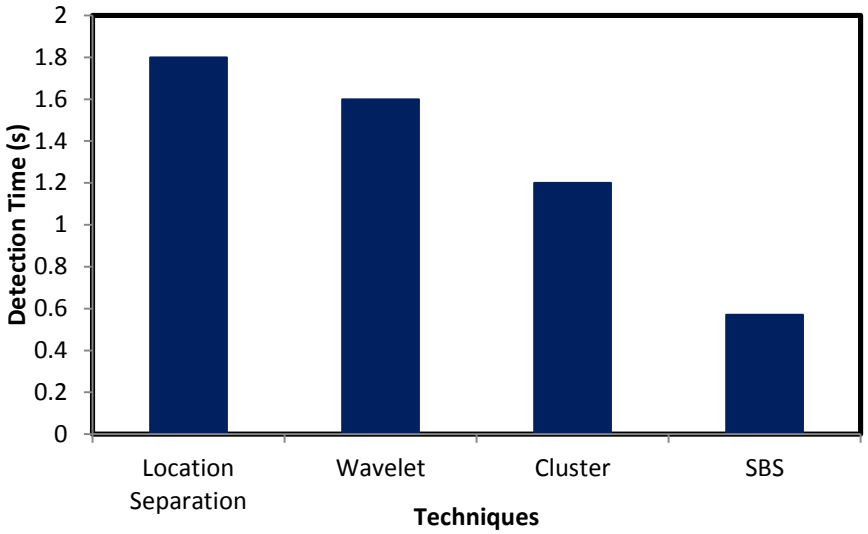

Figure-4. Analysis on detection time.

The time consumption for detecting the abnormal behaviour must be relatively low for any attack detection system. The attack detection system must be vigilant against the operations of the network. The attack has to be detected all at once, such that the security of the system is safeguarded and the normal functionality of the system is continued.

When the proposed approach is analysed with respect to detection time, it shows lesser time when compared to the other techniques. This is because, the proposed approach utilizes crispy features for computing the feature representative and the matching process is done easier. This paves way for the proposed approach to detect the abnormal scenario immediately.

In substance, the proposed approach detects the DDoS attacks effectively, whenever it happens. In spite of the cent percent attack detection, the FN and FP rates are found to be the least and tolerable. Finally, the time it takes to detect attacks is comparatively low, when compared to the other techniques. Thus, the main objectives of a DDoS prevention or detection system are satisfied.

5. CONCLUSIONS

This article presents a SDN based solution to prevent DDoS attack for cloud computing. The DDoS attack is prevalent in cloud computing and is necessary to build an effective solution. The proposed approach decomposes the entire work into knowledge gaining phase and operational phase. In the knowledge gaining phase, the attack prevention system is imparted enough knowledge to distinguish between the normal and the attack scenario.

In the operational phase, the attack detection system classifies between the normal and the attack scenario, based on the knowledge it gained from the knowledge gaining phase. Finally, the performance of the proposed approach is analysed with respect to detection rate, FNR, FPR and detection time. The experimental analysis shows the efficacy of the proposed approach. In future, this work is planned to be extended by varying the feature selection techniques.

REFERENCES

[1] Wang B.; Zheng Y.; Lou W.; Hou Y.T. 2015. DDoS attack protection in the era of cloud computing and Software-Defined Networking. Comput. Netw. 81: 308-319.

[2] McGregory S. 2013. Preparing for the next DDoS attack. Netw. Secur. pp. 5-6.

[3] Vissers T.; Somasundaram T.S.; Pieters L.; Govindarajan K.; Hellinckx P. 2014. DDoS defense system for web services in a cloud environment. Future Gener. Comput. Syst. 37: 37-45.

[4] Malecki F. 2012. Simple ways to dodge the DDoS bullet. Netw. Secur. pp. 18-20.

[5] Software Defined Networking Definition. 2015. [Online]. Available: https://opennetworking.org/sdn-resources/sdn-definition.

[6] N. Mckeown. 2011. How SDN will Shape Networking. [Online]. Available: http://www.youtube.com/watch?v=c9-K5O qYgA.

[7] S. Schenker. 2011. The Future of Networking, and the Past of Protocols. [Online]. Available: http://www.youtube.com/watch?v=YHeyuD89n1Y.

[8] H. Kim and N. Feamster. 2013. Improving network management with software defined networking. Communications Magazine, IEEE. 51(2): 114-119.

[9] OpenFlow: Enabling Innovation in Campus Networks. 2015. [Online]. Available: http://archive.openflow.org/documents/openflow-wp-latest.pdf.

[10] Z. Duan, X. Yuan, J. Chandrashekar. 2008. Controlling IP spoofing through interdomain packet filters. IEEE Trans. on Dependable and Secure, Computing. 5: 22-36.

[11] F. Yi, S. Yu, W. Zhou, J. Hai, A. Bonti. 2008. Source-based filtering algorithm against DDoS attacks. International Journal of Database Theory and Application. 1: 9-20.

[12] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker and S. Savage. 2006. Inferring internet denial-of-service activity. ACM Transactions on Computer Systems. 24(2): 115-139.

[13] S. Ostermann, B. Tjaden M. Ramadas. 2003. Detecting anamalous network traffic with self-organizing maps. Recent Advances in Intrusion Prevention. pp. 36-54.

[14] E. Mota, A. Passito R. Braga. 2010. Lightwight DDoS flooding attack Prevention usingNOX/Openflow. IEEE 35th conference on Local Computer Networks, 2010, pp. 408-415.

[15] Jin S, Yeung DS. 2004. A covariance analysis model for DDoS attack detection, 2004 IEEE International Conference on Communications, IEEE, Paris, France. 1882-1886.

[16] Peng T, Leckie C, Ramamohanarao K. 2004. Proactively detecting distributed denial of service attacks using source IP address monitoring, 2004, Springer, Third International IFIP-TC6 Networking Conference, Athens, Greece. 771-782.

[17] Open flow specification version 1.0.0, http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf

[18] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J.2008. Open flow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38: 69-74.

[19] Gude N., Koponen, T., Pettit, J., Pfa B., Casado M., McKeown N., Shenker S. 2008. Nox: towards an operating system for networks. SIGCOMM Comput. Commun. Rev. 38: 105-110.

[20] Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., Ramanathan R., Iwata Y., Inoue H., Hama T., Shenker S. 2010. Onix: a distributed control platform for large-scale production networks. In: Proceedings of the 9th USENIX conference on Operating systems design and implementation. pp. 1-6. OSDI'10, USENIX Association, Berkeley, CA, USA.

[21] Zheng Cai, Alan L. Cox, T.S.E.N.: Maestro: A system for scalable open flow control, http://www.cs.rice.edu/~eugeneng/papers/TR10-11.pdf.

[22] Tootoonchian A., Ganjali Y. 2010. Hyper flow: a distributed control plane for open flow. In: Proceedings of the 2010 internet network management conference on Research on enterprise networking. pp. 3-3. INM/WREN'10, USENIX Association, Berkeley, CA, USA.

[23] http://www.caida.org/data/.

[24] Luo H, Lin Y, Zhang H, Zukerman M. 2013. Preventing DDoS attacks by identifier/locator separation. IEEE Network. 27(6): 60–65.

[25] Srihari V, Anitha R. 2014. DDoS detection system using wavelet features and semi-supervised learning, Springer Second International Symposium on Security in Computing and Communications (SSCC), Delhi, India. 291-303.

[26] Bhaya W, Manaa ME. 2014. A proactive DDoS attack detection approach using data mining cluster analysis. Journal of Next Generation Information Technology. 5(4): 21-36.