I.J. Information Technology and Computer Science, **2015, 11, 34-42**
Published Online October 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.11.05

Accurate Anomaly Detection using Adaptive Monitoring and Fast Switching in SDN

Gagandeep Garg[1]
Research Scholar, dept. of I.T., U.I.E.T., Panjab University, Chandigarh, 160014, India.
E-mail: gagandeepgarg900@gmail.com

Roopali Garg[2]
Supervisor, dept. of I.T., U.I.E.T., Panjab University, Chandigarh, 160014, India
E-mail: roopali.garg@gmail.com

***Abstract*—**Software defined networking (SDN) is rapidly evolving technology which provides a suitable environment for easily applying efficient monitoring policies on the networks. SDN provides a centralized control of the whole network from which monitoring of network traffic and resources can be done with ease. SDN promises to drastically simplify network monitoring and management and also enable rapid innovation of networks through network programmability. SDN architecture separates the control of the network from the forwarding devices. With the higher innovation provided by the SDN, security threats at open interfaces of SDN also increases significantly as an attacker can target the single centralized point i.e. controller, to attack the network. Hence, efficient adaptive monitoring and measurement is required to detect and prevent malicious activities inside the network. Various such techniques have already been proposed by many researchers. This paper describes a work of applying efficient adaptive monitoring on the network while maintaining the performance of the network considering monitoring overhead over the controller. This work represents effective bandwidth utilization for calculation of threshold range while applying anomaly detection rules for monitoring of the network. Accurate detection of anomalies is implemented and also allows valid users and applications to transfer the data without any restrictions inside the network which otherwise were considered as anomalies in previous technique due to fluctuation of data and narrow threshold window. The concept of fast switching also used to improve the processing speed and performance of the networks.

***Index Terms*—**Anomaly detection, SDN, flow-counting, traffic- aggregation, Adaptive traffic monitoring, Network management, fast switching, bandwidth utilization.

I. Introduction

With the emergence of the SDN's technology, the concept of programmable networks has again come into the picture. Network flow path decisions are made by the controller present in the control plane of SDN. SDN can improve the functionality of the networks and change the way they operate. Also, OpenFlow is treated as a new revolutionary idea in networking that provide secure transfer of control flows [1]. SDN is currently capturing a greater amount of academia and industry attraction. A number of network operators, vendors, and service providers had recently laid down an industrial driven organization named "Open Network Foundation" [2] to promote SDN. Security of SDN is the most important area of concern for its rapid deployment in today's fast growing enterprises. With the totally changed security architecture, it is a challenge to learn the whole new security paradigms and implement them. SDN open interfaces and known protocols also provide opportunities to invaders for various attacks. Including with its benefits of higher innovation, separation of the control plane from the forwarding plane also being a security problem for various enterprises using SDN. All three layers of SDN can be targeted for attack due to that extra level of complexity added to it. Centralized SDN controller still provides us the whole view of the network, that view can be used to easily apply various security policies and intrusion detection system inside the network for making it more secure. Using SDN it is easier to monitor the network traffic and make the dynamic decision, according to the traffic patterns. But due to new encapsulation and overlying infrastructure techniques for hiding the data-flows, most of the existing security tools are unable to monitor and inspect the SDN traffic. Various new monitoring policies and mechanisms need to be implemented on the network traffic while maintaining the performance of the network. Virtualization and abstraction hide the underlying infrastructure so it is harder for an attacker to attack the forwarding elements directly because they are not exposed directly to the attacker.

Centralized control of SDN also makes it easier for attackers to attack central controller with certain attacks like DOS attacks, malicious traffic insertion etc. So to mitigate such attacks efficient monitoring of the network traffic for different intrusions; implementation of certain efficient policies for intrusion/anomaly detection are required. In [3] important security issues for SDN are discussed. Different possible attacks on SDN controller like DOS (Denial of service) attack, fake packets insertion, and running unauthorized programs are discussed and several techniques already proposed to tackle such issues are also listed. Securing SDN from these kinds of attacks is really a challenging task. Efficiently monitoring the network traffic flow and analyzing all the traffic while maintaining the performance of the network can solving such security issues. For better monitoring and management of the network, accurate statistics needs to be gathered using detection queries and other policies to detect malicious data packets or anomalies in network traffic.

Traffic monitoring over the controller can significantly increase the overhead on the SDN controller. It can result into the degradation of network performance. Also applying complex queries can further increase the overhead to the higher extent which can result into loss of legal data packets. So to overcome such problems, a mechanism is required which maintain a proper balance between complexity of detection policies and monitoring overhead over the controller. For that certain existing work proposed random packet sampling technique [4] to choose a sample of packets from the whole flow of network traffic to analyze. But this method can be implemented on small scale network and likely to miss small flow entries during heavy traffic. So small flow entries remained unmatched with detection query policies which result in the residence of anomalies in the network. Other proposed techniques are port scan and enhancement of sampling techniques [5]. These enhanced sampling techniques failed to efficiently implement the anomaly detection behavior while balancing the overhead cost and query policy complexity.

An adaptive flow counting method for anomaly detection proposed in [6] which provided such efficient mechanism to detect anomalies while controlling the overhead cost. According to the malicious detection, it updates its flow counting rules for aggregation of traffic adaptively. The modified version of this algorithm with reduced complexity is proposed in [7] and further implemented in [8] and generated similar results to verify the reduced complexity in [9]. This work is mainly focused on accurate anomaly detection by dynamic threshold range calculation using efficient bandwidth utilization, to overcome the fluctuations considered as anomalies. It also maintains the performance of the network by using fast switching which enhances the processing speed of the network.

The rest of paper organized as follows: In II, some work related to anomaly detection by other researchers is given. In III, the methodology used for our algorithm is defined for accurate anomaly detection. In IV, implementation details and gathered statistics is shown. V represents the accuracy of the results obtained using our algorithm. In VI, the effect on performance is evaluated. Finally in VII, we conclude this research work by discussing the importance of adaptive technique and future work on its further improvement.

II. Related Work

Dynamic flow path updating in SDN motivates us towards the work of adaptive anomaly detection using dynamic flow-counting. Our work also draws from the inspiration from existing methods for monitoring and detecting anomalies inside the network. In [10, 11] various methods different network topologies using NetFlow for anomaly detection implemented. In this work, entropy was used as a summarization tool for classification and aggregation of traffic of the network. Also, supervised learning for the detection of DOS attack was used. In [12] methods used for anomaly detection motivated us for applying statistic collection methods for monitoring and measurement of network traffic. In [13] rate limiting and maximum entropy detection methods used for four prominent traffic anomaly detection algorithms. These four algorithms are used for detection of anomalies by limiting the rate of network traffic data. In [14] SDN-Scanner provided with fingerprinting method for detection malicious traffic and detect the DOS (denial of service) attack. This method used the reporting interval as parameter for capturing the time difference between two transfers and attacking the network. But this method also not provide any dynamic rule for the case of heavy traffic and failed to fingerprint the anomalies in that case. DIFANE [15] focuses on the general scalability problem. The generated fake packets may still cause large communication and computation burden. DIFANE needs to work all the time to monitor and keeps flow rule update in data plane. However, this may lose information about new incoming flows from controller's point of view.

In [16] time trade-off between statics collection overhead and performance provided and analyzes the effect on performance on applying tighter monitoring. AvantGuard [17] proposed new architecture as data plane extensions to protect our network from control plane saturation attack that disrupts network operations. AvantGuard introduces connection migration, actuating triggers on the data plane’s available statistics gathering services. It provides detection of changing flow dynamics including the responses to it, within the data plane. Connection migration allows the data plane to protect the control plane from these kind of saturation attacks. Control layer applications insert triggers to register for the asynchronous call backs, and insertion of conditional flow rules. These triggers were fired when certain condition is detected at collected statistics. AvantGuard increases the scalability and responsiveness towards threats, with connection delay overhead of 1%. AvantGuard overcomes the intrinsic bottleneck presented by the interface between the control plane and data plane that expert hackers can exploit.

In [8] an approach towards improving the efficiency and reducing the complexity of the anomaly detection algorithm using adaptive flow-counting was implemented. Another work in [9] redefined this adaptive anomaly detection algorithm and further implemented it with gathered statistics. This algorithm is also evaluated in this work by comparing the results with the existing results. On behalf of this given technique, another algorithm for dynamic calculation of certain parameters like threshold range for making decision and fast switching are implemented in this paper. This work effectively helps in accurate detection of anomalies in the network while maintaining the performance of the network using fast switching concept.

III. Methodology

Adaptive monitoring approach used in [7] reduces the complexity of the dynamic rule updating algorithm given in [6]. But this approach still unable to detect anomalies accurately. In case of authentic or legitimate users who want to send large amount of data flows inside the network and causes larger fluctuation in data flow counts, this algorithm considers such flow entries as detected anomalies inside the network and raises an alarm of detected anomalies for these data flows. However, such data flows can be from some legitimate users or internal data servers inside the organization containing larger data flow counts. Hence, another extension of this algorithm is required to accurately detect anomalies. For detecting anomalies accurately and allowing the internal servers and other legitimate users to send data, a new method for calculating the values of dynamic threshold range (RL – RU) of the dynamic rule update algorithm is implemented in this work. In the existing algorithm, threshold range value RL and RU are calculated as static values and can be given as:

$$RL = Mean - (3 * standard\ deviation)$$
$$RU = Mean - (3 * standard\ deviation)$$

However, this range of values considers the fluctuations of data as anomalies when applied on the aggregated data as used in the existing algorithm and causes a problem for valid users. So using this approach anomaly count comes out to be much higher than the original anomalies present in the network. To overcome such problem, a new paradigm is implemented which replaces this method for calculating the threshold range and helps in accurate detection of anomalies. In this paradigm, this dynamic threshold range values can be calculated using the efficient bandwidth utilization. This approach is represented in algorithm 1 and implemented for improving the accuracy of the adaptive anomaly detection method.

In this algorithm, a threshold range of values for making the decision for dynamic rule updating algorithm is calculated by utilizing the bandwidth efficiently. These values of range RL and RU will be re-calculated each time a new user will connect to the network or whenever any user leave the network or isolated from the network. Initially available total network bandwidth is assigned to the network and minimum required per node is set. After this, this find out the total number of active data streams inside the network. Available network bandwidth per node is calculated by deducting the essential data header field bandwidth form total allocated bandwidth and then bandwidth for each current data flow is calculated. According to these obtained values, the value of the dynamic threshold range RL and RU are updated by comparing the currently active data stream bandwidth for each data flow so that internal data flow fluctuations can be ignored and allows them to safely transfer data without raising alarms for detection of anomaly in case of legitimate users. This mechanism helps in detecting anomalies accurately inside the network.

Algorithm 1. Calculation of values for upper and lower range R_L, R_U in dynamic rule update algorithm.

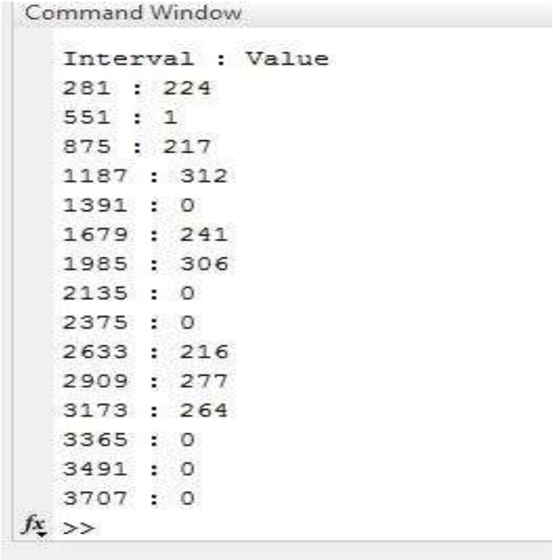
```
Procedure:

Step 1. Some total bandwidth is assigned to the network and also set a minimum bandwidth required for each node.
Step 2. Calculate the number of active streams inside the network.
Step 3. Calculate the essential bandwidth for essential header data for the data flows inside the network.
Step 4. Available bandwidth be calculated for data flows for the whole network and can be given as:

        Available BW= Total BW – Essential BW

Step 5. Available average bandwidth per node computed as:

        Available average BW= Available BW/ Total Nodes

Step 6. Average of currently active bandwidth per node be computed by available bandwidth and active nodes as:

        Average Active BW= Available BW/ total active nodes

Step 7. The current stream of data flow bandwidth is calculated using the packet history.
Step 8. If (Current data stream BW) > (Average Active BW)

        RU= Average Active BW

Else-if (Current data stream BW) < (Average Available BW)

        RU= Average Available BW

Step 9. If (Current data stream BW) < (Minimum Required BW)

        RL= Current data stream BW

    Else

        RL= Minimum Required BW

END
```

IV. Implementation

For the implementation of this algorithm, a controller based simulator is created using MATLAB environment. In MATLAB, different modules created for implementing this scenario, as due to large scale of network traffic it's not feasible to capture the real network's traffic aggregates. Different simulator modules created are the controller module, topology builder module, packet generator module, data propagation and entropy calculation modules. These modules perform different tasks like the topology builder module creates different topologies for our network and configured for dynamically choosing different topology each time, so that it can be related to real time network scenario. The controller module is the kernel of the simulator model. It handles all the data being transferred between the nodes in the current network topology. The controller module collect the information about the data flows and capture other different statistics and apply various network policies accordingly. Packet generator module generates packet of any number of bytes packet size randomly and also covers very small or very large data packets. It can choose the payload size of data dynamically to relate it with real time scenario. In data propagation module, data is propagated from one node to controller and then that data will be analyzed and forwarded to different nodes. This module encapsulates data and convert it into binary form and vice versa. Data transferred between two subnets using random gateway decided dynamically. This gateway link will be created while creating topology by topology builder module. Controller gathers statistics on transfer of data as data paths are decided by the controller.

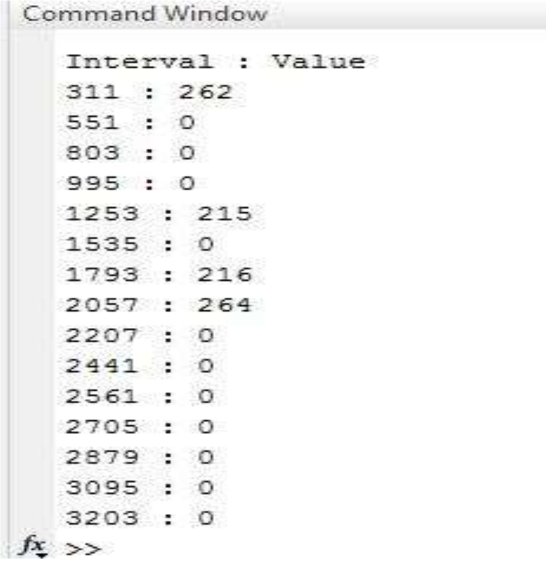
These captured statistics can be used for dynamic calculation of predicted range of values and making decisions according to these values. This work is the further extension to our work given in [9] for improving the accuracy of anomaly detection using adaptive monitoring. In this implementation, it is noticed that fluctuation of data flow count inside the network can be considered as anomalies and detection rate for anomalies comes out to be higher than the actual amount of anomalies present in the network. This case of fluctuations for previous implementation of dynamic rule updating algorithm for detected anomalies are shown by captured snapshot in figure 1. However, the numbers for detection of anomalies for the fluctuations have reduced to the certain extent on using the dynamic threshold calculation algorithm given by us as algorithm 1. This algorithm for calculating the dynamic threshold ranges of values RL and RU have reduced the anomaly count by allowing the transfer of data flows for the legitimate users while considering their bandwidth requirement for the particular amount of time spans. Values of anomaly detection count of data are shown in figure 2 using the snapshot captured in MATLAB. It shows the reduction in count than the previous result of figure 1 and able to consider fluctuations of data from legitimate users as valid data.

In figure 1, interval represents the number of packets transferred for the first interval in the first row and on each interval these number of packets added up to the existing value. Value column in this figure represents the number of anomalies detected for that particular interval of time. These stats represent the values for data flows for fifteen intervals. Initially larger number of anomalies can be detected due to the controller going through the learning phase. After the learning, detected anomaly count keep on decreasing as it accurately detects the anomalies. In the existing mechanism fluctuations in the data will be detected more which causes the controller to raise more number of alarms even for the valid users. This reduces the accuracy of the existing algorithm. However, our dynamic threshold range calculation algorithm can be able to overcome this problem by correctly detecting anomalies and considering fluctuation as required by the proper working of the network on applying various anomaly detection rules on the basis of decision made according to the new threshold range.



```
Interval : Value
281 : 224
551 : 1
875 : 217
1187 : 312
1391 : 0
1679 : 241
1985 : 306
2135 : 0
2375 : 0
2633 : 216
2909 : 277
3173 : 264
3365 : 0
3491 : 0
3707 : 0
>>
```

Fig.1. Snapshot of stats captured using existing algorithm used in [7] and static calculation of range values (RL and RU)



```
Interval : Value
311 : 262
551 : 0
803 : 0
995 : 0
1253 : 215
1535 : 0
1793 : 216
2057 : 264
2207 : 0
2441 : 0
2561 : 0
2705 : 0
2879 : 0
3095 : 0
3203 : 0
>>
```

Fig.2. Snapshot of stats captured on implementing proposed algorithm for calculation of range values (RL and RU)

Further, this methodology is implemented with higher performance of the network by reducing the process of recalculation of threshold values for the data flows. This dynamic threshold dynamic range algorithm is applied to only suspicious data flows which uses higher data bandwidth and introduces the risk of DOS or DDOS attack. This improvised implementation also improves performance of the network.

V. Accuracy

For accuracy of anomaly detection using dynamic threshold range (RL is the lower range and RU is the upper range) calculation algorithm, the graphs for detection of anomalies with respect to prefix aggregation mask length are generated. These graph are obtained using the dynamic rule updating algorithm and also upon implementing threshold range calculation algorithm for different simulation times using different capturing intervals. These graphs are compared with the graphs obtained by using existing static threshold range calculation to the graphs obtained using new dynamic threshold range calculation algorithm for dynamic rule update algorithm.

In figure 3.1 fraction of anomaly detection graph with respect to prefix aggregation mask length is generated for the existing static range value algorithm at simulation time = 30 seconds using capturing interval of 2 seconds. Similarly, other graph is also generated in figure 3.2 for accurate anomaly detection implementing our range calculation algorithm. Also, in figure 4.1 and 5.1 similar graphs are generated at simulation time 60 seconds and 90 seconds respectively using the capturing interval of 3 and 5 seconds respectively for existing technique. Similarly, in figure 4.2 and 5.2 improved accurate detection graphs are generated at simulation time (ST) = 60 seconds and 90 seconds respectively using the capturing interval of 3 and 5 seconds using new algorithm. In all these graphs, it is found that anomaly count with respect to prefix mask length decreases in case of improved algorithm using our range calculation algorithm i.e. the fluctuations caused by the internal server large data flows or fluctuations in traffic by the data flows of others legitimate users are not considered as anomalies. Hence, it can be able to detect anomalies more accurately than the previous existing anomaly detection algorithm using adaptive flow counting with the static calculation of range (RL-RU).

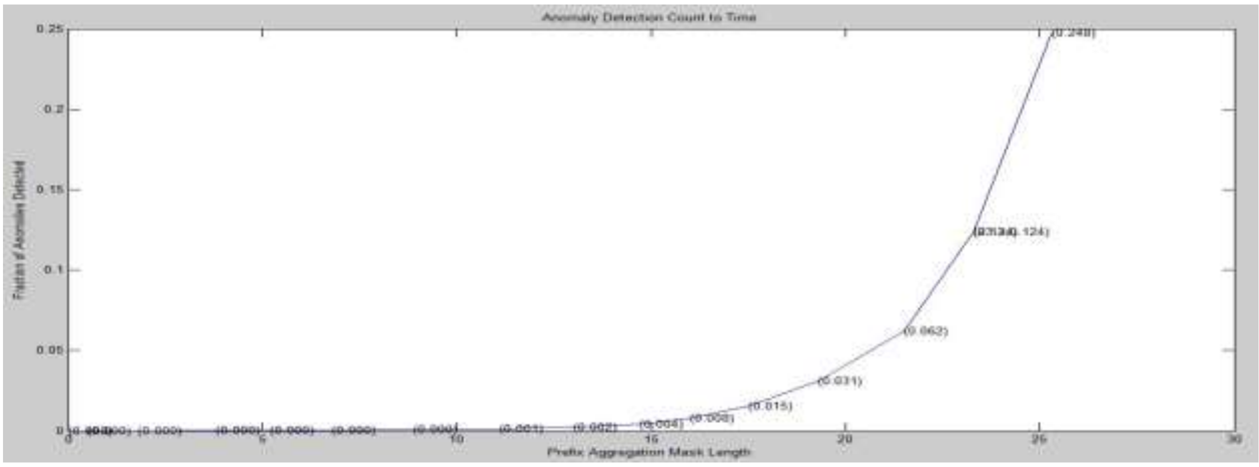


Fig.3.1. Existing anomaly detection using static range calculation at (ST= 30s)

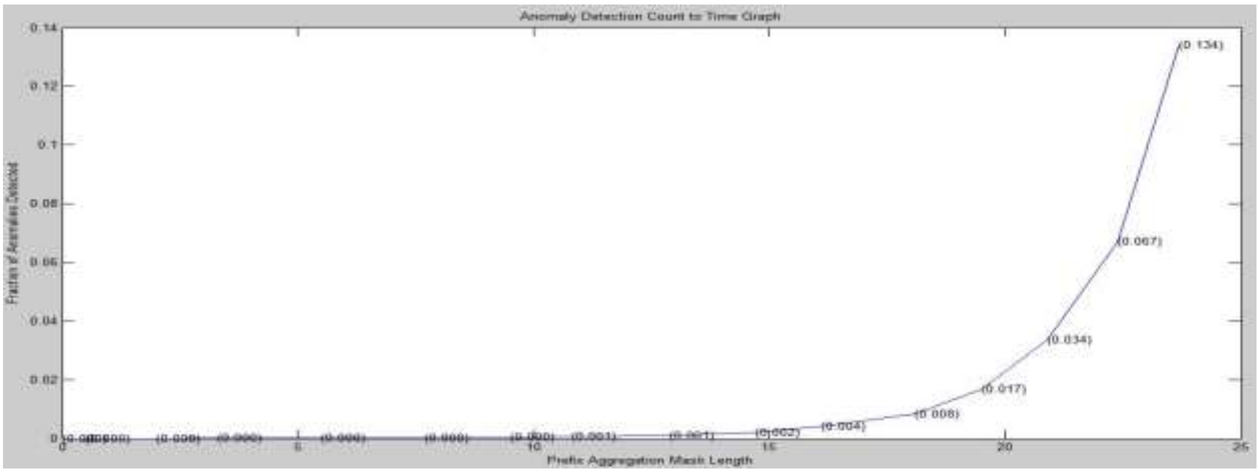


Fig.3.2. Accurate anomaly detection implementing range calculation algorithm at (ST= 30s)

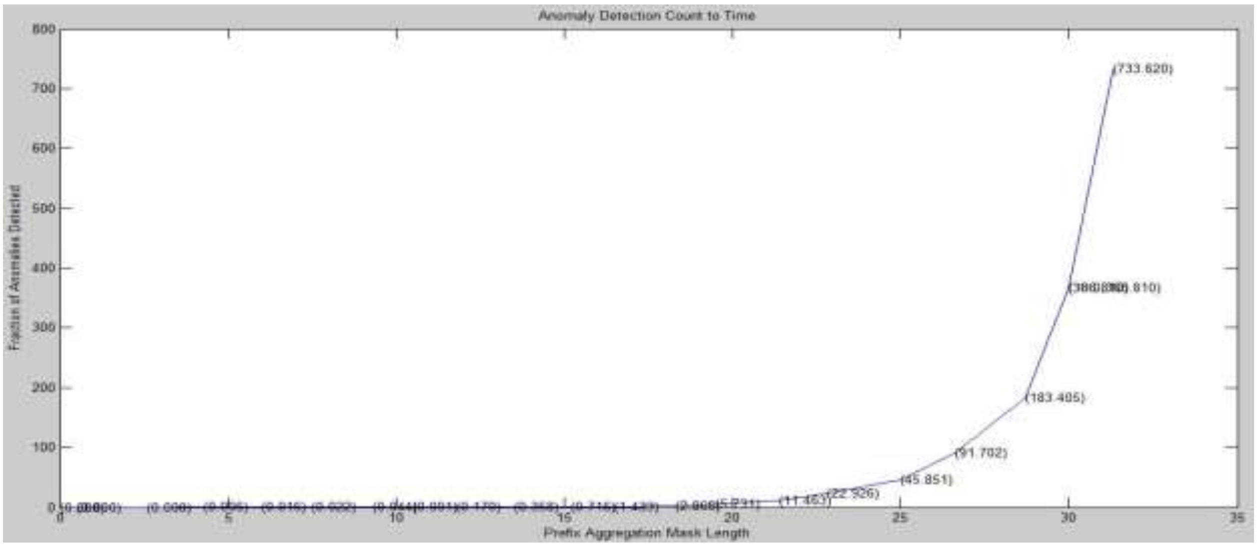


Fig.4.1. Existing anomaly detection using static range calculation at (ST= 60s)

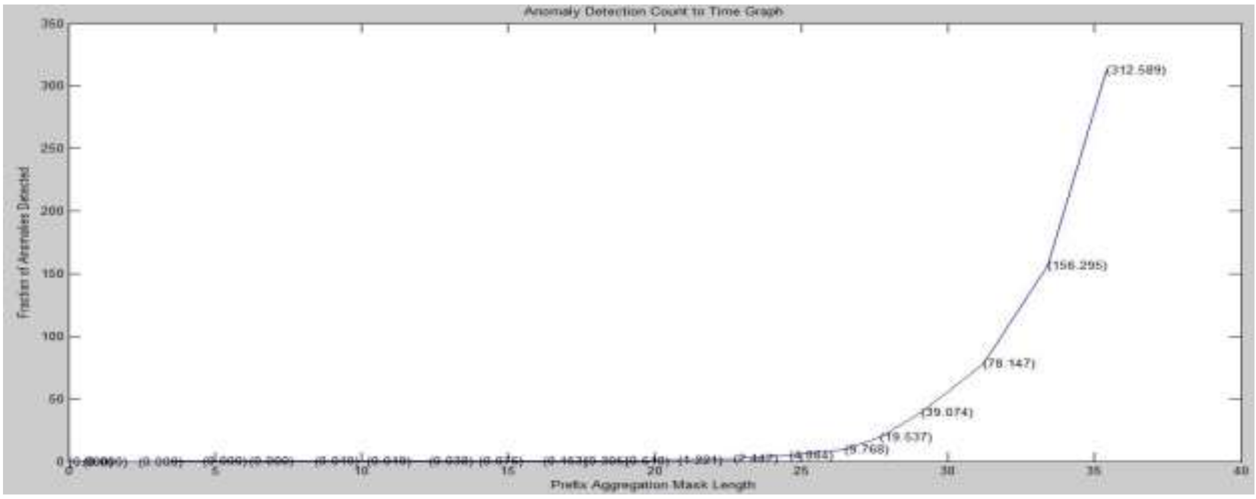


Fig.4.2. Accurate anomaly detection implementing range calculation algorithm at (ST= 60s)

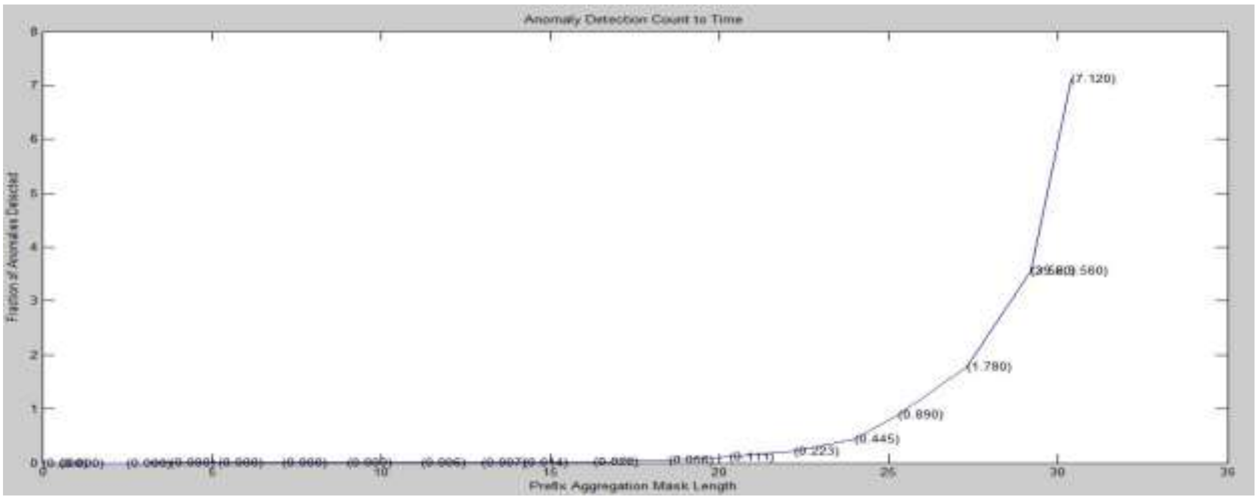


Fig.5.1. Existing anomaly detection using static range calculation at (ST= 90s)

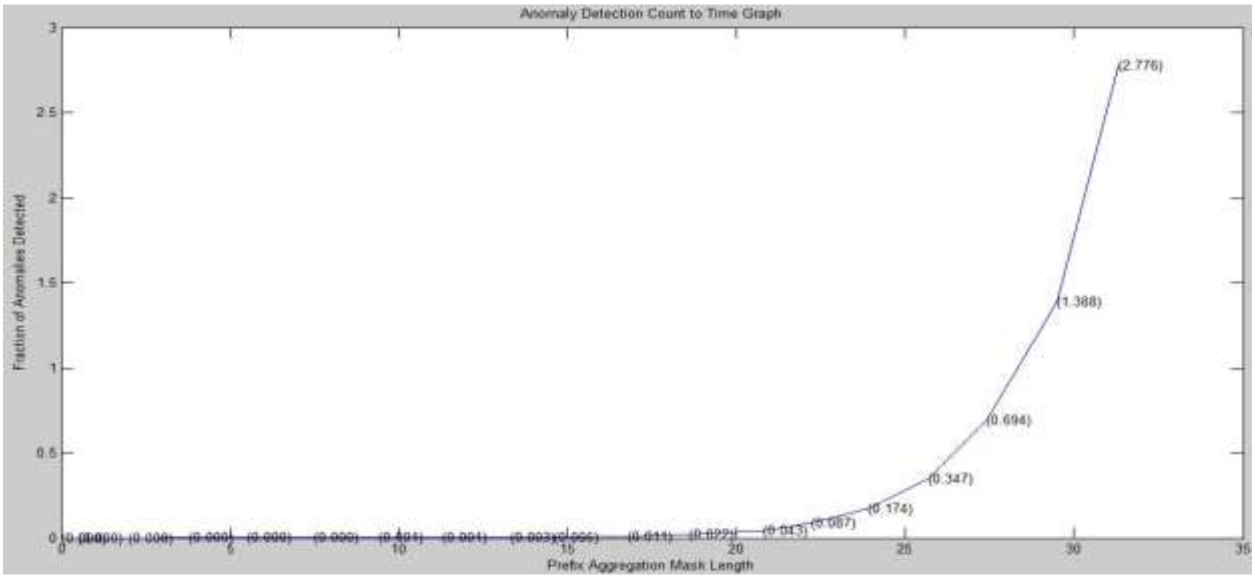


Fig.5.2. Accurate anomaly detection implementing range calculation algorithm at (ST= 90s)

VI. PERFORMANCE

The performance of the network is the most important aspect to be taken care of, as networks need to perform better for proper and real time data delivery. Packet loss must need to be minimum for real-time data flows while monitoring the statistics for making the decision on the basis of these statistics. In this algorithm, adaptive parameters are re-calculated each time for new data flow entry, this can results into the greater overhead over the controller. So, to overcome this problem, it is required to implement such scenario that only mandatory or suspicious data flows can be monitored and values of range and other adaptive or dynamic parameter must need to be recalculated in case of suspicious data flows only.

In this work, it further improvises the range calculation algorithm to the higher extent to maintain the performance of the network. The improvised algorithm is given as algorithm 2.

Algorithm.2 Improvised range calculation-Fast switching

```
Procedure:

Step 1. Calculate available bandwidth per node in the network.
Step 2. If (Flow-Counter) =1
            Call range calculation algorithm 1
Else-if (Current data stream BW) < (Average Available BW)
            Do Nothing
Else
            Call range calculation algorithm 1
END
```

In this improvised range calculation algorithm, adaptive parameters and other calculation will be made only on the suspicious data flows in the traffic. According to this algorithm, initially for the first data flow entry in the network whole parameters for the calculation of range (RL- RU) and making the decision will be computed. But for the next data flows, it will check for the current active bandwidth i.e. if the current data stream bandwidth is less than the available bandwidth per node then there is no need to recalculate all the parameters as there are no chances of attacks like DOS (denial of service), DDOS etc. due to the fact that bandwidth consumption is well within the legitimate range of available bandwidth. This mechanism implements the concept of fast switching and fast forwards the data flows which in-turn reduces the overhead of recalculation over the controller and increases the performance of our network. However, if the bandwidth utilization by the current flow is higher than the available bandwidth, only then adaptive parameters will be recalculated dynamically and values will be updated for the anomaly detection for the next data flows to maintain accuracy for detection.

The performance of this algorithm is evaluated by generating the similar graphs on implementing the improvised range calculation over the above algorithm1. Similar graphs generated by implementing algorithm 2 upon algorithm 1 are shown in figure 6.1 and 6.2 at simulation times 30s and 90s respectively. The similarity of these graphs with the improved accurate detection graphs represent that we had achieved higher performance of the network while keeping the accuracy of the anomaly detection intact.

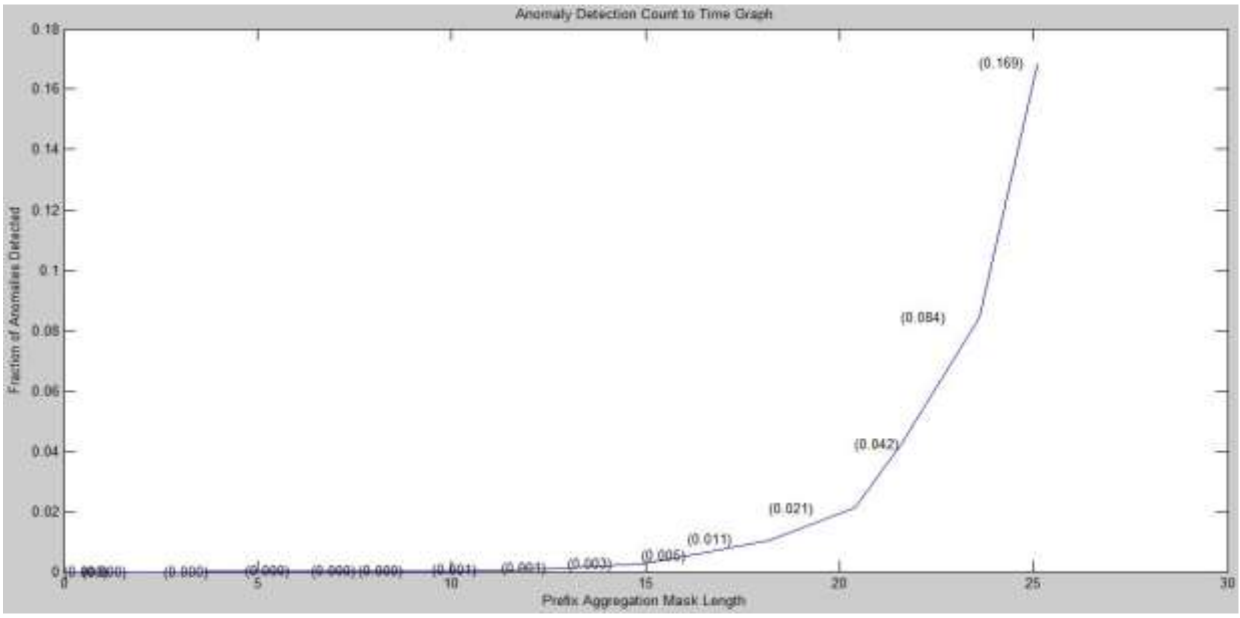


Fig.6.1. Anomaly detection graph using improvised range calculation at ST=30s

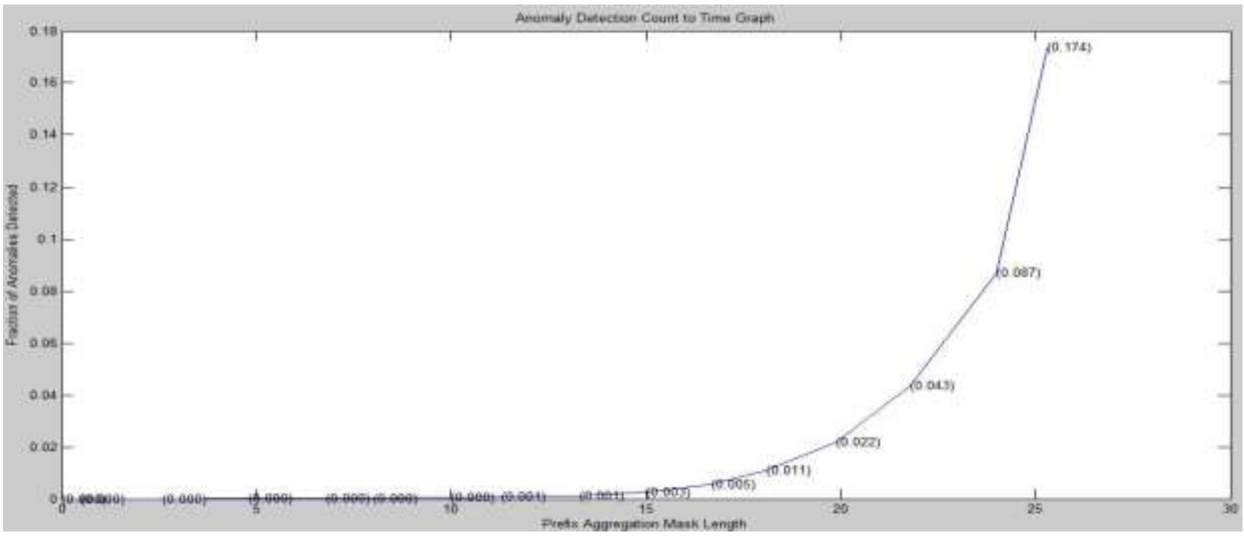


Fig.6.2. Anomaly detection graph using improvised range calculation at ST=90s

VII. Conclusion and Future Work

Security of modular SDN architecture relies on ensuring the integrity of network traffic. To ensure the integrity of the data, it is required to find out various malicious flow entries present in the network. Hence, it is required to implement efficient anomaly detection algorithms. But on implementing such algorithm on the network, it is also required to manage the monitoring overhead to keep performance matches. For that, an adaptive rule update algorithm is provided in this work, with greater efficiency and reduced complexity. Its dynamic nature provides zooming of some flow aggregates on expanding them which helps in applying finer granularity rules on our network traffic and detect anomalies with greater accuracy. The existing algorithm also considers fluctuations of data as anomalies even for legitimate users and raise the alarm. However, a dynamic threshold range calculation algorithm is provided in this work, which dynamically update the threshold range values according to the bandwidth utilization and accurately make decisions of anomaly detection. Results show that upon implementing this improved algorithm, the accuracy of anomaly detection increases, and performance is also improved using fast switching concept by recalculation of dynamic parameters in case of suspicious flows only. In future, other certain parameters like header space divider and sampling rate can be considered for further improvement of this technique. Also, trusted users like internal server, managerial data flows from management servers etc. can be added to a trusted IP addresses list on the basis of some parameters. So that, they can also be fast forwarded even without applying any monitoring policies on internal data so that processing speed can be further increased.

ACKNOWLEDGMENT

I would like to thank my co-author and supervisor Ms. Roopali Garg for providing me her valuable guidance and assistance for the completion of this research work. I would also like to thank all my colleagues and friends without whose support this research could not be completed.

REFERENCES

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar , L. Petrson , J. Rexford, S. Shenker , and J. Turner , "OpenFlow :Enabling innovation in campus networks", ACM SIGCOMM Computer Communication, Vol. 38, Issue 2, pp. 69-74, 2008.

[2] M. Betts, S. Fratini, N. Davis, R. Dolin and others, "SDN Architecture". Open Networking Foundation ONF SDN ARCH, Issue 1, June 2014.

[3] G. Garg and R. Garg "Review on architecture and security issues in SDN", International Journal of Innovative Research in Computer and Communication Engineering" Vol. 2, Issue 11, pp. 6519-6524, November 2014.

[4] T. Zseby, T. Hirch and B. Claise, "Packet sampling for flow accounting: challenges and limitations", Passive and Active Network Measurement Lecture Notes in Computer Science, Vol. 4979, springer, 2008. pp. 61-71.

[5] J. Mai, A. Sridharan, C. N. Chuah, H. Zang, and T. Ye "Impact of Packet Sampling on Portscan Detection", Selected Areas in Communications, IEEE Journal, Vol. 24, Issue: 12 pp. 2285 – 2298, December, 2006.

[6] Y. Zhang, "An adaptive flow counting method for anomaly detection in SDN", ACM Proc. of CoNEXT, Santa Barbara, California, USA December, 2013, pp. 25-30.

[7] G. Garg and R. Garg: Detecting anomalies efficiently in SDN using adaptive mechanism. In: IEEE, International conference on advance computing and communication technologies (ACCT2015) pp. 367-370, Rohtak, INDIA. Feb. 2015, doi: 10.1109/ACCT.2015.98.

[8] G. Garg and R. Garg, "Lecture Notes in Computer Science: Security of Networks Using Efficient Adaptive Flow-Counting for Anomaly Detection in SDN". In Springer: AISC, International Conference on Communication, Computing and Power Technologies (ICCPT-2015) Chennai, April, 2015, pp. 536-544.

[9] G. Garg and R. Garg, "Efficient anomaly detection using adaptive monitoring in SDN". In International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5 Issue 6 pp. 498-501, June, 2015.

[10] P. Banford, J. Kline, D. Plonka, and A. Ron.: A signal analysis of network traffic anomalies. ACM Digital library. Proc. of SIGCOMM IMW'02 (2002) pp. 71-82.

[11] A. Lakhina, M. Crovella, and C. Diot.: Mining anomalies using traffic feature distributions. ACM Digital library. Proc. of SIGCOMM, Philadelphia Pennsylvania, USA, 2005 pp. 217-228.

[12] K. Giotis, G. Androulidakis, and V. Maglaris.: Leveraging SDN for efficient anomaly detection and mitigation on legacy networks. In Proc. of third European Workshop on Software Defined Networks (EWSDN), Budapest, Hungary 2013.

[13] S. A. Mehdi, J. Khalid and S. A. Khayam.: Revisiting Traffic Anomaly Detection using Software Defined Networking. In: Springer. Recent Advances in Intrusion Detection, 2011.

[14] S. Shi, G. Gun, "Attacking Software-Defined Networks: A First Feasibility Study", ACM Proc. of HotSDN, Hong Kong, China, 2013 pp. 165-166.

[15] M. Yu, J. Rexford, M. J. Freedman, and J.Wang., "Scalable flow-based networking with DIFANE", In Proceedings of ACM SIGCOMM conference SIGCOMM'10, pp. 351-362, Vol. 40, Issue 4, October 2010.

[16] M. Moshref, M. Yu, and R. Govindan, "Resource/Accuracy Tradeoffs in Software-Defined Measurement", ACM, Proc. of HotSDN'13, pp.73-78, Hong Kong, China, August 2013.

[17] S. Shin, V. Yegneswaran, P. Porras, G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks", In Proceedings of ACM SIGCOMM conference CCS, pp. 413-424, Berlin, Germany, 2013.

Authors' Profiles

Gagandeep Garg a post graduate student of Information Technology, is undergoing a mandatory research in the innovative networking technology known as software defined networking. He is pursuing his master's degree from U.I.E.T, Panjab University and has pursued his bachelor's degree from Lovely Professional University. There is a review and three research papers published to his credit which have been published in international journals and IEEE X-plore. He holds a strong interest in the area of Networks and working towards the security and anomaly detection in the field of SDN.

Roopali Garg is working as Assistant Professor at department of Information Technology Engineering at U.I.E.T., Panjab University Chandigarh. She has an experience of 12 years in academics. She has done M. tech in Electronics and B. Tech in Electronics & Electrical Communication from Punjab Engineering College. She has been awarded Administrator's Gold medal by Chandigarh Administration in 2000 for her supreme performance in curricular, co- curricular and extra- curricular activities. There are more than thirty research papers to her credit which have been published in good indexed international journals and have been presented in reputed international conferences. Her focused research area is Wireless communication and has guided more than a dozen M. thesis in this area.

How to cite this paper: Gagandeep Garg, Roopali Garg,"Accurate Anomaly Detection using Adaptive Monitoring and Fast Switching in SDN", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.11, pp.34-42, 2015. DOI: 10.5815/ijitcs.2015.11.05