Computer Systems Science & Engineering
DOI:10.32604/csse.2022.020993
Article

Tech Science Press

Grey Hole Attack Detection and Prevention Methods in Wireless Sensor Networks

Gowdham Chinnaraju* **and S. Nithyanandam**

Department of Computer Science and Engineering, PRIST Deemed to be University, 613403, India
*Corresponding Author: Gowdham Chinnaraju. Email: gowdhamphd12@gmail.com

Received: 18 June 2021; Accepted: 27 August 2021

Abstract: Wireless Sensor Networks (WSNs) gained wide attention in the past decade, thanks to its attractive features like flexibility, monitoring capability, and scalability. It overcomes the crucial problems experienced in network management and facilitates the development of diverse network architectures. The existence of dynamic and adaptive routing features facilitate the quick formation of such networks. But flexible architecture also makes it highly vulnerable to different sorts of attacks, for instance, Denial of Service (DoS). Grey Hole Attack (GHA) is the most crucial attack types since it creates a heavy impact upon the components of WSN and eventually degrades the performance of network. In current study, a simple attack detection, prevention and reduction approach is proposed. This is to secure the WSN from GHA and other such attacks by warning and blocking the malicious suspensions and by examining the storage table. Instead of blocking the entire host, the presented approach specifically eliminates the malicious nodes. Further, in case of no malicious traffic detection, the host gets unblocked. In current study, the researchers simulated the model under MATLAB environment and the outcomes showed an enhanced performance and increased utilization of Central Processing Unit (CPU) and packet delivery ratio.

Keywords: Grey-Hole Attack; WSN; detection; prevention; reduction; network performance

1 Introduction

Wireless Sensor Networks are decentralized in nature and are appropriate for different kinds of applications since the centralized nodes cannot be trusted. WSNs also exhibit network scalability feature i.e., it can be connected with networks of huge size, whereas the whole networks should be recognized [1]. Rapid deployment and minimal configuration make WSNs, a preferable option for emergency crises like disaster and military conflicts [2]. With the availability of dynamic and adaptive routing facilities, one can create the network quickly. Wireless networks tend to have high architectural variations, compared to wired networks [3]. WSN frequently encounters packet drop-based attacks in which the host may broadcast the packets in case of encountering a short path [4]. However, the traffic which is directed at the host, is compromised. Hosts are susceptible to collaborative attacks and may be compromised due to which it may pass wrong information to the network. Some existing protocols may exhibit resistance towards selective drops, through node thwarting, in order to avoid from being overloaded [5]. It acquires routing reliability by disabling the link as defective or by attaining newer effectual routes toward the destination. In order to resolve this selective drop attack, reliability factors are chosen after the evaluation of link weight list [6]. For instance, if a sum of weights to certain route is higher, it specifies lower reliability in which the attack nodes can be identified [7]. Every node has to maintain a certain weight and the attained weight is added to the payload route request. Through the evaluation of consistency rate, the defected nodes are differentiated from normal ones. Fig. 1 shows the architectural diagram of Grey Hole Attack scenario. When a node fails, it has a heavy impact on packet routing [8]. Therefore, these kinds of nodes have to be identified and isolated to eliminate the partition of network, since it influences the survival of network itself. Node failure is identified by routing protocols.

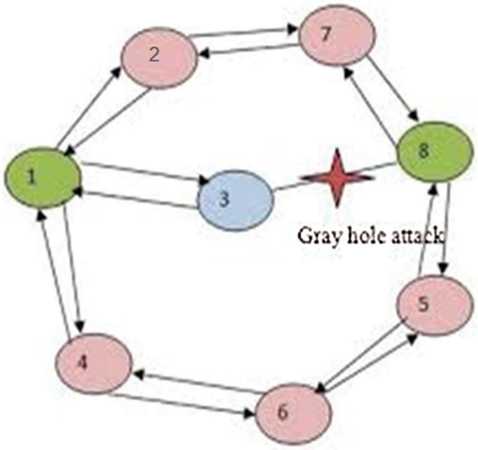

Figure 1: Grey hole attack scenario

In a scenario, where the node fails and another node commences its route discovery process, the failed node may not be able to transmit the packets received from downstream nodes [9]. When neighborhood nodes fail, the initiation nodes are incapable of communicating with one another. In such case, the commencing node gets isolated by its neighbors. This failed node is considered as a selfish node. When this node starts its route discovery process, the selfish node may disincline to forward the request from initiation node [10]. This node discards the data packets forwarded to it. Therefore, the communication between these two nodes can be fulfilled. If neighbors are selfish, it may not be able to communicate with other nodes in a single hop. However, selfish node may communicate with other nodes and distinguish it from failure nodes [11]. The anticipated model is designed to resist the Grey Hole Attack through by node thwarting and avoid from being overloaded. It may acquire routing reliability with the help of reliable factors and by disabling a link as either defective or by attaining an effectual route towards the destination [12]. One of the significant contributions of the current study is the detection of malicious nodes in Grey Hole Attack. The current research work selected an effectual technique to provide security. This technique provides superior network security by fulfilling availability, integrity and performance in WSN.

In this study, simple attack detection, prevention and reduction approaches are described in detail. These approaches are used to secure WSN from GHA by warning and blocking the malicious suspensions and attacks and by examining the storage table. The anticipated approach specifically eliminates the malicious nodes, instead of blocking the entire host. Further, it also unblocks the host if there is no malicious traffic present. The simulation was conducted in MATLAB environment.

Rest of the paper is organized as follows. Section II provides an overview of the existing works and techniques. Section III discusses about the theoretical approach to eliminate Grey Hole Attack. Section IV details about the experimental set up and simulation and Section V concludes the research paper with future research direction.

2 Related Works

The current section is a review of studies conducted earlier and the section deals with methodologies used in the mitigation of WSN attacks.

Pal et al. [13] conducted a study to mitigate attacks by facilitating the switches to validate the source using TCP handshake approaches. After effectual validation, the controller installs the flow rates to initiate data transfer. This can be applied in TCP to transfer data which needs variations in switches. Lee et al. [14] proposed a load balancing approach to improve network survival rate in which the overloaded paths are partitioned among switches to reach successive routes. Different ways are followed to validate this approach and reduce the overload.

Pal et al. [15] developed an application to preserve IP and MAC addresses and connect the hosts with table so that its location can be tracked completely. While joining or leaving the host, the information has to be updated properly. Lee et al. [16] transferred a limited number of packets to the controller so as to avoid DoS attacks. To get rid of this table overflow, the researcher recommended optimal timeout value and flow aggregation for flow rules. Though this is the most essential one, there exists some variations in legitimate packet drops too. Armenia et al. [17] recommended an observable architecture and deployed the network to identify packets with required signature. Those packets are transmitted to the controllers when the port is jammed and the presence of malicious nodes is proved thereafter. Chang et al. [18] proposed an approach to eliminate attacks in which an abnormal user is considered to transmit less number of packets than a normal user during the induction of a session. Some tend to block the flow, if the transmitted packets are lesser than the provided number.

Aijaz et al. [19] utilized a trust-based approach to avoid attacks by providing IP-based trust values for every packet, based on communication. According to the researchers, queuing is a possible way to eliminate attacks, since the queues can be created from different switches. Weighted round robin method can be utilized to get the request from various queues, based on its size. Further, the latter characteristic is also used in the partition of queue switches into port queue. Razzak et al. [20] proposed different approaches to gather packets from controller and switch. These approaches compute the ratio between total and packet message by host. When the ratio exceeds a pre-defined threshold, then the controller that maintains the switch transmits less messages. Ferraz et al. [21] presented an effectual DoS mitigation and detection method termed ‘Anti-DoS’ comprised of four approaches namely, mitigation, trace back, detection and attack trigger. Attacks are identified with the help of back propagation NN which further tracks the source and blocks it by installing flow rules.

TCP / IP (Transmission Control Protocol / Internet Protocol) is indeed a set of protocols that are independent of the transmission object used during wireless communication. However, most data transmission functions start and end with Ethernet frames, besides internet communication. Ethernet can be used both as a bus topology and a star topology. A bus topology connects all the devices to a long wire in pattern. All the issues are typically wired to the main hub inside a star topology. Star-shaped bus topology is a combination of two topologies and are applied in 10Base-T. This is because the data that comes from the cable can be decided for sharing, by an attached device i.e., star-shaped wiring.

Xia et al. [22] proposed a method with regards to spoofing attack in which the information on attacker location is gathered, packets are collected from switches, features are hauled out and organized classifier is used. Data cache, stored in the table, is missed during saturation point. Cheelu et al. [23] proposed a detection model that stores the table statistics and identifies the fixed interval to determine abnormalities. Komninos et al. [24] proposed a data plane solution termed ‘Line switch’ to fight against the attack. It eliminates buffer saturation and TCP limitation over the guard and protects the controller through migration. Proxies may give initial connection from IP address which are prioritized on some probability.

Khare et al. [25] proposed flow installation technique to diminish DoS attacks effectually over the control channel and controller. This may install rules all over the switches from source to destination to attain the request. Peng et al. [26] presented a DoS detection system based on centralized controller. It comprises of anomaly and pre-processing detection models. Guo et al. [27] proposed a dynamic routing method that distinguishes the flow from resource utilization, gathers routing paths to eliminate overflow and allocates ineffectual bandwidth. It is appropriate for multiple paths and links. Wang et al. [28] proposed a load balancing approach to diminish the response time and channel utilization using different controllers.

3 Proposed Method

The proposed method comprises of four diverse functionalities such as Network Data Collection (NDC), Grey Hole Detection (GD), Grey Hole Prevention (GP) and Grey Hole Reduction (GR). GD and GR processes are executed only once, when executing the system. However, other functionalities are iterated based on the dependencies of analysis. For instance, network data collection gets triggered only when a network controller acquires the packet from neighbourhood nodes, whereas prevention process gets triggered, when the flow crosses a threshold. All these functionalities are explained below. The flow diagram of the proposed model is shown in Fig. 2.

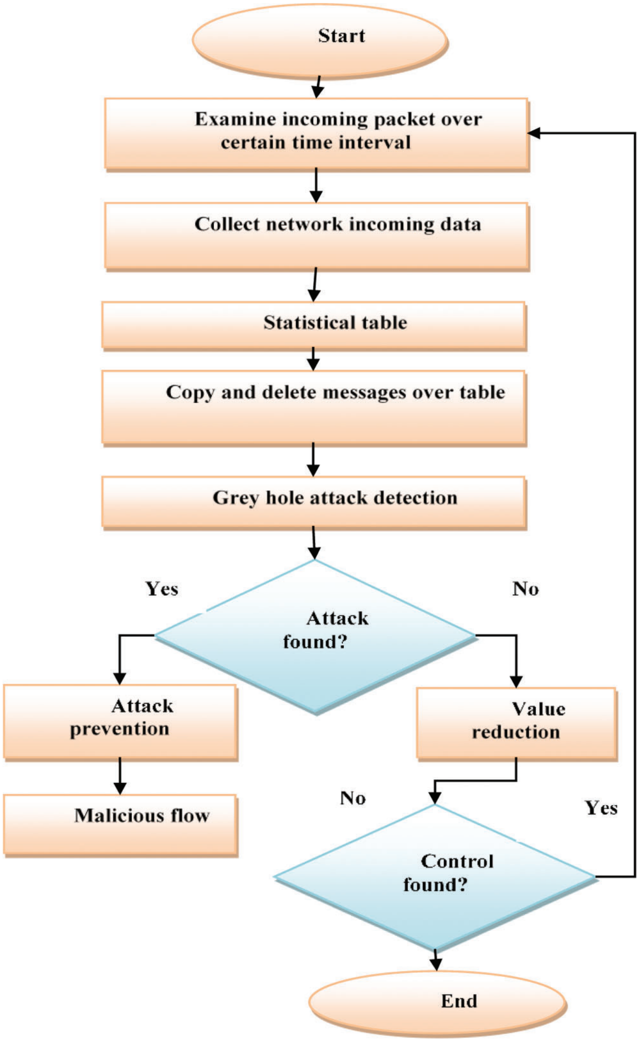
Start
Examine incoming packet over certain time interval
Collect network incoming data
Statistical table
Copy and delete messages over table
Grey hole attack detection
Yes
No
Attack found?
Attack prevention
Malicious flow
Value reduction
No
Yes
Control found?
End

Figure 2: Flow diagram of the proposed model

1) Network Data Collection (NDC): The messages are collected here, when nodes encounter an incoming message. Then it gets stored in the table. The table may be modelled with data structure so as to process further. When initiating an iteration, these incoming messages are deleted after generating a copy for GD process. The information, gathered from incoming packets, may comprise of Ethernet types, switch ID, port ID, source and destination addresses. The information connected with network, data link and transport layers get stored over three diverse dictionaries. Here, various header file combinations are used to store the information. NDC may access and aggregate the data, when incoming packets are stored in the field. It may not influence the flow rule installation of controllers. The process may take a constant time, whereas the complexity is predicted in the order of $O(1)$.

2) Grey Hole Detection (GD): This is an initial function that describes the characteristics of incoming traffic, after the anomalies are determined in NDC. It validates whether the total number of requests, from a certain combination port id, switch id, IP or port, during iteration interval, may exceed the flow limit or not. If the flow limit is exceeded, then prevention process bypasses the contents of arguments. Then, GD initiates the analysis from higher level by validating MAC and flooding. This process is primarily performed to determine IP spoofing and flooding, when malicious attackers are identified. Similarly, detection process may determine fine-grained factors involved in flooding/spoofing. Therefore, it may limit legitimate packet drop.

Under every normal condition, the controllers may receive incoming message from switches and hosts. Then, the complexity is measured as $O(|s|.|h|)$ while it is not executed owing to its normal value, $'h'$. During attack, ‘h’ value gets enhanced owing to flooding or spoofing. This process may go deeper to identify the fine-grained information about the attacks.

Algorithm 1:

```
//statistical process of data collection
For all table do
If request > flow rate limit then
Prevention (switch, portin))
End if
End for
If MAC > flow rate limit then
Frequency < Flow rate/2 then
Prevention (MAC< switch, port_in)
Elseif MAC > flow rate limit then
  If source IP > flow rate limit then
  Attack prevention (MAC, switch, portin)
  Endif
  For all switch, MAC< portin
  If total_port > flow rate limit then
  Attack prevention (MAC, portin, switch)
  End if
  End for
  End if
  End for
  End if
  End for
  End for
```

3) Grey Hole Prevention (GP): This step gets triggered every time, when GD is identified during Grey Hole Detection process. It may generate threat entries, handle values, take decisions toward traffic and compute the block interval. Every statistic, maintained in the table, is termed as ‘malicious flow’. This function may generate an entry for parameter flow, received from threat detection procedure, while there is no such entry during malicious flow. Wildcard field in flow is specified by ‘any’.

If an entry identifies the same flow, the process gets incremented with threat value. The value may be 0.5 to 1 based on port type. It may be either external or internal. If the updated value is higher, then the warning is given through flow rule installation to eliminate the suspected traffic for a short period of time. If the updated value is higher than the blocking value itself, then the flow rate is made fit to eliminate the traffic for a certain period of time, based on threat value. Flow rate is computed as an aggregate of random numbers on timeout value. Here, the duration of blocking flow rate is evaluated by aggregating the squares of threat values in flow duration. A random value is computed to eliminate the rule at certain period, which further blocks the port at the time of flooding. Finally, a timestamp is included, when updating the record.

Algorithm 2:

```
//blocking malicious traffic
flow → incoming message
if flow.portin == external then
    threat = yes;
else
    threat = partial
endif
if flow.portin == malicious data then
    flow.time → totaltime ();
    threat value → flow value + threat value
    duration = 0;
    if threat value > warning threshold
    then
        Duration = random value + timeout;
        if flow. Block == true
        Then
            Flow.block → yes
            Duration = $(\text{threat value})^2$ + duration
            Aggregate flow (idle, timeout, drop time);
        Else
            Add flow (idle, timeout, drop time)
        End if
    End if
    Flow.time → time
Else
    Flow → match
    Time → present time ():
    Threat value → threat;
    Duration → 0;
End if
return
```

4) Grey Hole Reduction (GR): This process gets executed after the completion of threat detection, with iterations of the proposed model. This function gets gradually diminished with values and blocks the duration that remains inactive. If the flow is inactive to reach a timeout period, then it gets eliminated automatically from an appropriate switch. Then, it is moved to malicious flow table. Therefore, no traffic flows through it. Further, duration and value are eliminated after the expiry of malicious flow duration table. This gets continued until the outcome of inactive block entry is eliminated from the table.

Algorithm 3:

```
//Threat reduction
Flow ∈ malicious node do
    If block = = yes then
        If (duration + idle) < present time () then
            Flow. Threat → threat value/2;
            Flow → duration/2;
            Time → present time ();
            If duration <= 1 then
                Block = 0;
            End if
        End if
    Else
        If duration = = 0;
        Then
            Eliminate flow;
        Elseif (time+idle) < present time ()
        Then
            Eliminate flow;
        End if
    End if
End for
```

It determines data collection with header data from every packet and preserves the collected data in table format for all the intervals. Threat detection provides the table after copying data from source. These processes run for certain period, when a threat is determined. Then, attack prevention and related malicious flow values are raised. Moreover, if a threat value crosses over threshold, then flow rule is made to fit the malicious traffic over switches. Subsequently, if there is no threat, the threat value gets reduced by blocking the flow rules based on activeness.

The parameters related to this function are discussed herewith. Iteration period is determined as the time between two successive iterations. This may work iteratively by collecting data to identify the attack. The value may be in the range of 2 to 10 s based on network preferences. In other words, the value is deterred based on how it identifies and blocks the suspicious node with processing power. A pictorial representation explains it elaborately.

Request limit is defined as the maximal number of requests generated after flow. Here, host may transmit the request by controlling the iteration interval. If a host transmits the flow request, then it may start considering the host as a suspicious node. This may get restricted based on network size and is set as 50 for every small network connection. There are 15 to 100 host networks present in the setup. Then, threshold is denoted by the time computation of the host to cross flow request limit within a certain period of time. Based on the threshold, this system is installed with traffic as a warning message for short interval. The threshold warning value is set between 2 and 6.

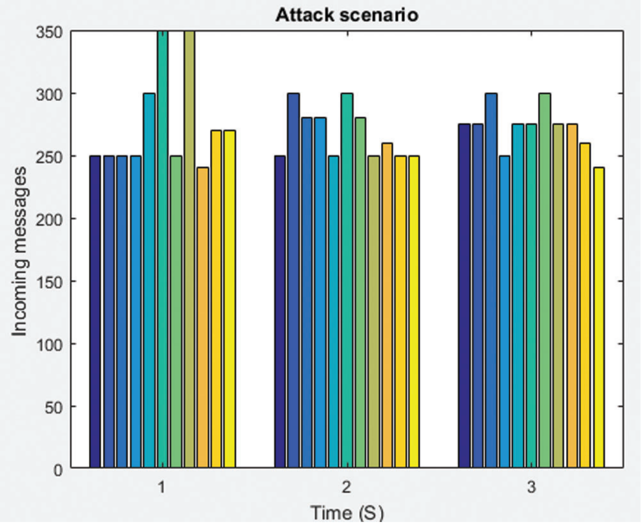

Figure 3: Attack scenario with incoming message

Blocking the threshold specifies the limit to show the number of times. The host crosses the flow rate limit for a certain period of time. Based on the threshold value, traffic is measured because the malicious flow is installed to block for a certain period of time. Blocking threshold value is set as a double value for warning threshold.

Finally, idle timeout is inactive and removed from the switch in case of no flow of packets through it. Timeout is measured as the total period after the removal of switch, irrespective of whether it is active or inactive. These are considered as default parameters.

4 Numerical Results and Discussions

This section explains the experimental setup, simulation outcomes and performance analysis conducted in the study. The researcher simulated the model in a PC with configuration such as 3.4 GHz, 8 GB RAM and windows operating system. Random traffic was established with TCP connection that can transmit 100 data packets to the destination host. The total number of Grey Hole Attack hosts was provided for various scenarios that cause constant flooding to consume high power and resources. This was remotely connected with multiple resources. The attack scenario with incoming messages is shown in Fig. 3. The attack scenario is provided in the Figs. 4–6. Tab. 1 shows the simulation set up parameters.

To evaluate the performance of the proposed system, default routing algorithm was considered with the following parameters namely, CPU utilization, bandwidth, response time, PDR and total amount of request flows towards the controller. The analysis was conducted as per the network scenario shown in the Figs. 7–9.

Scenario 1: Here, the proposed system was simulated using an easy topology consisting of four hosts and three switches. The analysis was conducted using both attack and non-attack scenarios. When attack scenario was considered, the processing power of CPU got increased with constant flow rules and was available for lesser limits in case of legitimate requests. The average utilization of CPU, in case of non-attack scenario, was 7.66%. In attack scenario, CPU utilization went up to 8.50%. Therefore, the utilization of CPU by the proposed model was comparatively less than the detection and mitigation of Grey Hole Attack. Further, the researcher considered a time flow of 5 s between malicious traffic and fit flow rule to eliminate the malicious traffic at high timeout values.

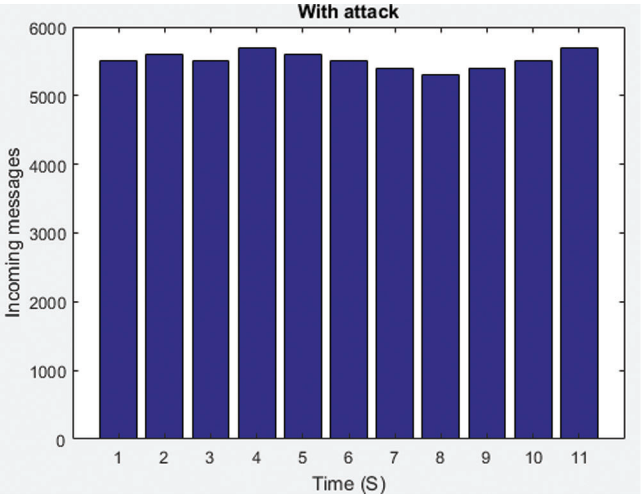

Figure 4: Incoming messages with attack

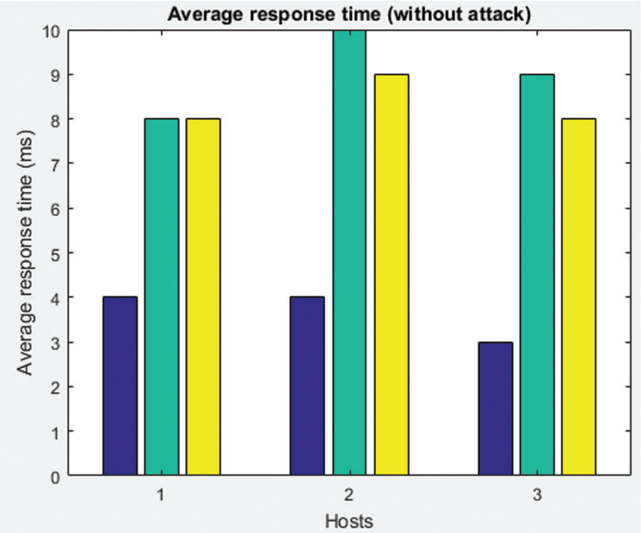

Figure 5: Average response Time

Channel bandwidth needs flow request through control channel and is considered as a bottleneck during attacks. It diminishes the load control channel by eliminating malicious traffic. In line with this, the proposed model was more active in the bandwidth of 14.24 kb/s and was lesser compared to the existing models. Incoming message-based flow requests were considered for major traffic in network. Grey hole was launched using features to overload the network, control channel and table value. The proposed model blocked the malicious requests in attack scenario. The total number of malicious requests got reduced in the proposed model than other models. In case of attacks, the response time of the host turned out to be a huge overload with fake requests to the network. The proposed model showed lesser response time in contrary to the prevailing models. This is because the proposed model eliminated malicious traffic to provide the normal hosts with free service. To compute the response time, it transferred the ICMP messages to hosts and heavily reduced the response time.

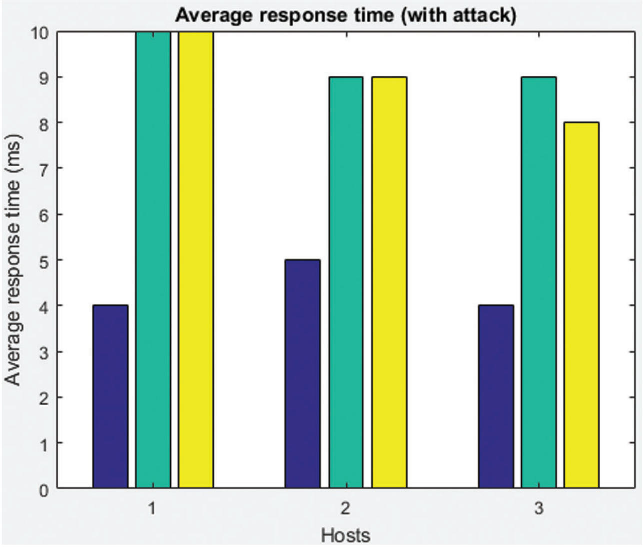

Figure 6: Average response time with attacks

Table 1: Parameter setup

Parameter	Value
Timeout	30 s
Idle time	15 s
Blocking threshold value	6
Warning threshold value	3
Iteration interval	5 s
Flow request level	75

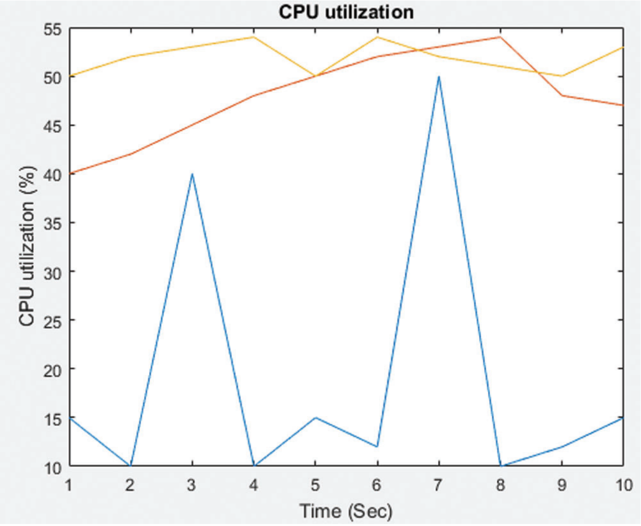

Figure 7: CPU utilization

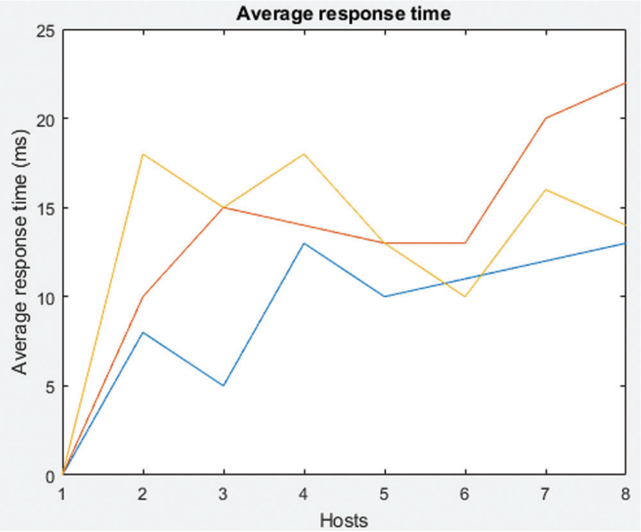

Figure 8: Average response time

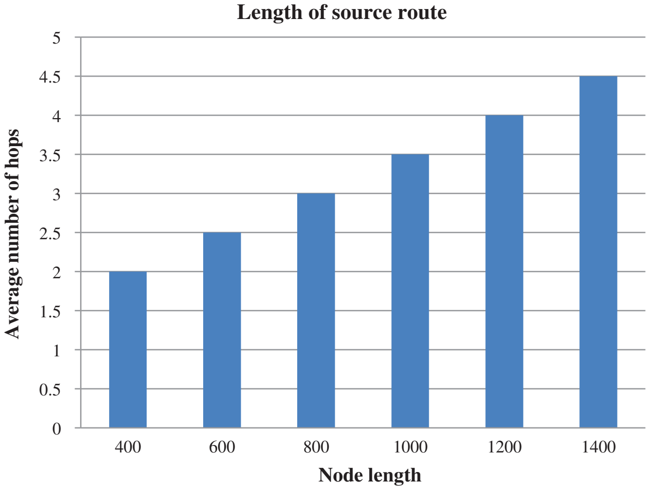

Figure 9: Average number of hosts

DR is defined as the ratio of total data packets transmitted by a source to the number of packets received at the destination. In this study, normal traffic flow was ensured using network connectivity. It transmitted TCP packets to the destination host. In case of encountering no response, it tends to retrieve it again. In this initial scenario, PDR was high as shown in Tab. 2.

Table 2: PDR computation

S.No	Message sent	Delivered	Packet drop	PDR
1	65800	65800	0	100
2	34600	12380	22480	35.21
3	63800	63800	0	100

The proposed model showed no additional delay, copied some information from the packets arrived at the network and did not interfere in the routing process. The proposed model also identified heavy traffic and flooding scenario based on iteration interval and warning threshold. This rule may block the flooding traffic based on 15 s installation which may offer immediate relief to the network. Threat identification process may generate flow that blocks the traffic using IP address and port numbers. The proposed model did not block the traffic of that port alone. However, with the proposed method, the IP address of that specific block alone can be blocked instead of complete port. If any system is intended to be blocked, due to multiple illegitimate activities, then it can be blocked after the expiry of flow rate level. Therefore, there is no need to generate a flooding traffic. The attacker modifies their behaviour to original state and gets unblocked after sometimes.

There is no need for statistics to mitigate the attacks. However, it is completely important to have the data collected during path finding. It may not generate traffic for verification or detection process. But it works over any anomaly detection process and there is no signature of database attacks. Therefore, the proposed model saves the costs incurred from processing, storage and time.

5 Conclusion

The attractive features of WSN network makes it highly vulnerable too, especially in case of Grey Hole Attack. This attack tends to degrade the performance of network completely. So, the current research paper developed a statistical model to mitigate malicious traffic using certain flow rules instead of blocking the complete IP or port address. The proposed system does not block the suspicious traffic rather it performs two processes. Initially, it ensures the traffic associated to have a complete flow and eliminates it for some time. When the attack continues, it declares the entity as a blocked or malicious user. Based on the proposed model, the researcher conducted investigations. The extensive outcomes demonstrated the overcoming of Grey Hole Attack through lesser CPU utilization, flow requests, response time, bandwidth and PDR in contrary to other approaches. Further, the proposed method showed some additional benefits i.e., no added delay and quick response to attacks. The proposed method further eliminated certain flows without any signature of database attacks. Further, it does not require any statistics from switches to detect the attacks. In future, the model can be tested in real-time applications and can be implemented in the development of effective clustering techniques.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Muthumayil, T. Jayasankar, N. Krishnaraj, M. Sikkandar, P. N. Balasubramanian *et al.,* “Maximizing throughput in wireless multimedia sensor network using soft computing techniques,” *Intelligent Automation & Soft Computing*, vol. 27, no. 3, pp. 771–784, 2021.

[2] S. Y. Dorbala and R. S. Bhadoria, “Analysis for security attacks in cyberphysical systems,” in *Cyber-Physical Systems: A Computational Perspective*, pp. 395 2015.

[3] I. Ahmad, S. Namal, M. Ylianttila and A. Gurtov, “Security in software defined networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[4] S. Shin, V. Yegneswaran, P. Porras and G. Gu, “AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks,” in *Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security - CCS ’13*, Berlin, Germany, ACM Publisher, New York, pp. 413–424, 2013.

[5] R. Kandoi and M. Antikainen, “Denial-of-service attacks in OpenFlow SDN networks,” in *2015 IFIP/IEEE Int. Symp. on Integrated Network Management (IM)*, Ottawa, ON, Canada, pp. 1322–1326, 2015.

[6] S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. R. Basha and T. Jayasankar, “An optimized deep neural network based DoS attack detection in wireless video sensor network,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2021. https://doi.org/10.1007/s12652-020-02763-9.

[7] P. Bera, A. Saha and S. K. Setua, “Denial of service attack in software defined network,” in *2016 5th Int. Conf. on Computer Science and Network Technology (ICCSNT)*, Changchun, China, pp. 497–501, 2016.

[8] T. Wang and H. Chen, “SGuard: A lightweight SDN safe-guard architecture for DoS attacks,” *China Communications*, vol. 14, no. 6, pp. 113–125, 2017.

[9] M. Imran, M. H. Durad, F. A. Khan and A. Derhab, “Reducing the effects of DoS attacks in software defined networks using parallel flow installation,” *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 69, 2019.

[10] H. Peng, Z. Sun, X. Zhao, S. Tan and Z. Sun, “A detection method for anomaly flow in software defined network,” *IEEE Access*, vol. 6, pp. 27809–27817, 2018.

[11] H. Wang, H. Xu, L. Huang, J. Wang and X. Yang, “Load-balancing routing in software defined networks with multiple controllers,” *Computer Networks*, vol. 141, no. 2, pp. 82–91, 2018.

[12] A. T. Mzrak, S. Savage and K. Marzullo, “Detecting malicious packet losses,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 191–206, 2009.

[13] S. Pal, B. Sikdar and J. Chow, “Real-time detection of packet drop attacks on synchrophasor data,” in *2014 IEEE Int. Conf. on Smart Grid Communications*, Venice, Italy, pp. 896–901, 2014.

[14] D. Lee and D. Kundur, “Cyber attack detection in PMU measurements via the expectation-maximization algorithm,” in *2014 IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, USA, pp. 223–227, 2014.

[15] S. Pal, B. Sikdar and J. H. Chow, “Detecting malicious manipulation of synchrophasor data,” in *2015 IEEE Int. Conf. on Smart Grid Communications (SmartGridComm)*, Miami, FL, USA, pp. 145–150, 2015.

[16] D. Lee, B. Carpenter and N. Brownlee, “Media streaming observations: Trends in udp to tcp ratio,” *Int. Journal on Advances in Systems and Measurements*, vol. 3, no. 4, pp. 147–162, 2010.

[17] A. Armenia and J. H. Chow, “A flexible phasor data concentrator design leveraging existing software technologies,” *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 73–81, 2010.

[18] J. M. Chang, P. C. Tsou, I. Woungang, H. C. Chao and C. F. Lai, “Defending against collaborative attacks by malicious nodes in MANETs: A cooperative bait detection approach,” *IEEE Systems Journal*, vol. 9, no. 1, pp. 65–75, 2015.

[19] A. Aijaz and A. H. Aghvami, “Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective,” *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[20] F. Razzak, “Spamming the internet of things: A possibility and its probable solution,” *Procedia Computer Science*, vol. 10, pp. 658–665, 2012.

[21] L. H. G. Ferraz, P. B. Velloso and O. C. M. B. Duarte, “An accurate and precise malicious node exclusion mechanism for ad hoc networks,” *Ad Hoc Networks*, vol. 19, no. 4, pp. 142–155, 2014.

[22] H. Xia, Z. Jia, X. Li, L. Ju and E. H. M. Sha, “Trust prediction and trust-based source routing in mobile ad hoc networks,” *Ad Hoc Networks*, vol. 11, no. 7, pp. 2096–2114, 2013.

[23] D. Cheelu, M. R. Babu and P. Venkatakrishna, “A fuzzy-based intelligent vertical handoff decision strategy with maximised user satisfaction for next generation communication networks,” *Int. Journal of Process Management and Benchmarking*, vol. 3, no. 4, pp. 420–440, 2013.

[24] N. Komninos, D. Vergados and C. Douligeris, “Detecting unauthorized and compromised nodes in mobile ad hoc networks,” *Ad Hoc Networks*, vol. 5, no. 3, pp. 289–298, 2007.

[25] K. Khare, J. L. Rana and R. C. Jain, “Detection of wormhole, blackhole and DDOS attack in MANET using trust estimation under fuzzy logic methodology,” *Int. Journal of Computer Network and Information Security*, vol. 9, no. 7, pp. 29–35, 2017.

[26] H. Peng, Z. Sun, X. Zhao, S. Tan and Z. Sun, “A detection method for anomaly flow in software defined network,” *IEEE Access*, vol. 6, pp. 27809–27817, 2018.

[27] Z. Guo, Y. Xu, R. Liu, A. Gushchin, K. Y. Chen *et al.,* “Balancing flow table occupancy and link utilization in software-defined networks,” *Future Generation Computer Systems*, vol. 89, no. 2, pp. 213–223, 2018.

[28] H. Wang, H. Xu, L. Huang, J. Wang and X. Yang, “Load-balancing routing in software defined networks with multiple controllers,” *Computer Networks*, vol. 141, no. 2, pp. 82–91, 2018.