Human-centric Computing and Information Sciences
a SpringerOpen Journal

RESEARCH Open Access

ColShield: an effective and collaborative protection shield for the detection and prevention of collaborative flooding of DDoS attacks in wireless mesh networks

I Diana Jeba Jingle[1*] and Elijah Blessing Rajsingh[2*]

* Correspondence: dianajebajingle@gmail.com; elijahblessing@karunya.edu
[1]Department of Computer Science and Engineering, LITES, Thovalai, India
[2]KSCST, Karunya University, Coimbatore, India

Abstract

Wireless mesh networks are highly susceptible to Distributed Denial-of-Service attacks due to its self-configuring property. Flooding DDOS attack is one form of collaborative attacks and the transport layer of such networks are extremely affected. In this paper we propose *ColShield*, an effective and collaborative protection shield which not only detects flooding attacks but also prevents the flooding attacks through clever spoof detection. *ColShield* consists of Intrusion Protection and Detection Systems (IPDS) located at various points in the network which collaboratively defend flooding attacks. *ColShield* detects the attack node and its specific port number under attack. In order to reduce the burden on a single global IPDS, the system uses several local IPDS for the collaborative mitigation of flooding attacks. The evaluation of *ColShield* is done using extensive simulations and is proved to be effective in terms of false positive ratio, packet delivery ratio, communication overhead and attack detection time.

Keywords: Collaborative; Bandwidth; Traffic; Timer; IPDS; Spoofing

Introduction

Wireless mesh networks (WMN) has a wired-cum-wireless semi-centralized infrastructure that allows an end host to easily join the network and communicate with any host by exchanging packets. WMN uses a high speed back-haul network that can transmit packets at high bandwidth in large range. WMN consists of gateways that optimize the network performance and integration with other wireless networks, intermediate mesh routers that are stationary and mesh clients that are mobile. The mesh routers must be synchronized [27] as it is the optimal feature of WMN. These mesh routers operate as bridging points in inter-network and can be integrated with other wireless devices. However, the mobility and self-configuring property of wireless mesh networks (WMN) makes the attackers to prevent the internet's service to legitimate users by flooding excess amount of messages to the corresponding server thereby forming a Denial of Service (DoS) attack. The main objective of DoS attacks is either to completely tie up certain resources or to bring down an entire network so that the legitimate users are not able to access service(s).

DoS attackers mainly use IP spoofing as a moderator for launching flooding attacks. Such spoof-based flooding attacks can be traced easily if launched by a single attacker. The most sophisticated type of DoS attack is the flooding attack [28] that occurs at all the layers of WMN [11]. In case, if multiple attackers are collaboratively involved in launching flood packets at the victim, it will lead to a Distributed Denial of Service attack which is one form of collaborative flooding attack. The collaborative flooding DDoS attacks [42] are spread by natural distributed processing architecture of the network. It normally floods the mesh clients and the intermediate mesh routers using hierarchal control points [37] to congest the WMN traffic communication. Collaborative flooding DDoS attacks exploits the huge resource asymmetry between the internet and the victim. Collaborative flooding attacks can bring the entire network down and they are very hard to detect because the attack is distributed. Also it is impossible to trace the attacker. The attackers use a large number of machines to collaboratively flood packets simultaneously at the victim. These machines are ready to participate in the attack and are called as compromised machines [31] or zombies. To avoid these issues, this paper focuses on spoof-based collaborative detection of collaborative flooding DDoS attacks.

Intrusion detection systems [34] can be used to detect such collaborative flooding attacks; however, they may have a high incidence of false alarms. Current rules-based and anomaly-based intrusion detection systems detect intrusions either by matching patterns of network and users activities with pre-defined rules or they define the normal profile of system usages and then look for deviation. These approaches have their consequences and drawbacks. The former is well suited for known intrusions but it cannot detect new intrusions. The latter relies on deviation from normal usage and sometimes fails to detect well known intrusions. This paper presents an effective intrusion protection and detection system (IPDS) that detects and prevents collaborative flooding attacks against clever spoofs at the mesh client level. *ColShield* comprises of a distributed two-level architecture with group of local IPDS at the mesh router level and a single global IPDS at the gateway router level. All these IPDS collaboratively involve in protecting the source network from collaborative flooding of DDoS attacks. This informative paper aims to be an opening to a research that could hopefully end up with a mechanism to prevent flooding attacks.

This paper proceeds as follows. Section Related work summarizes the related work. Section The proposed system describes the architecture and operation of *ColShield* system and its metrics and algorithms. Section Performance results presents the simulations [29] we conducted to evaluate *ColShield*. Finally Section Conclusion concludes the paper.

Related work

DDoS attacks are quite advanced and powerful methods to attack a network system and to make it either unusable to the legitimate users or downgrade its performance. They are increasingly mounted by professional hackers in exchange for money and benefits. Yet there seems to be no silver bullet to the problem. This survey examines the possible solutions to this problem and analyzes the feasibility of those approaches. Based on the analysis of existing solutions, we proposed a desirable solution to defend collaborative flooding of DDoS. *Firecol* [1] uses Intrusion Prevention Systems (IPS) which form virtual protection rings around the hosts to defend flooding attacks collaboratively

by exchanging selected traffic information. However, FireCol cannot detect the specific port under attack. $SACK^2$ [2] detects SYN flood attacks [5] against skillful spoofs. It does by identifying the victim server and the TCP port being attacked by exploiting the behavior of the SYN/ACK-CliACK pair. $SACK^2$ has low and controllable false positive and false negative rates as well as short detection delay. However, $SACK^2$ can detect only SYN flood attacks against skillful spoofs. TVA [24] uses capabilities to discard unauthorized traffic floods on a single autonomous system. TVA achieves high throughput, but the problem is TVA stores all capability information of each user on routers and a router with limited number of queues may not be able to protect all the legitimate users.

DWARD [13] autonomously detects and filters attack traffic from legitimate traffic by dropping the excess traffic by limiting the traffic rate to and from the victim thereby reducing the overload at the victim. But DWARD cannot detect attack traffic until connection buffer fills up thereby causing increased time delay to detect an attack and it causes more communication overhead. DARB [4] uses an active probing detection method and a TTL based rate-limit counteraction method to detect and filter SYN flooding attack [26] traffic accurately and independently on the victim side. DARB consumes more amount of the victim's bandwidth and causes computation overhead for both detecting and counteracting methods. Ge Zhang et al. [8] proposes a priority mechanism for blocking attacks on SIP proxies caused by external processing. But this mechanism causes time delay [41] and decreased throughput when SIP proxies interact with external servers. Haidar Safa et al. [9] proposed CDMS that is implemented at the edge routers of spoofed IP address' networks to defend the victim. CDMS also a communication protocol is used to encourage collaboration between various networks to protect each other. This mechanism is very efficient and it prevents the routers from being overloaded. However this mechanism causes time delay to detect and filter an attack. Sudip Misra *et al*. [20] proposed DLSR which uses the concept of Learning Automata (LA) and prevents the server being overloaded with excess amount of illegitimate traffic from crashing and keeps the server functioning. However DLSR cannot effectively differentiate valid user's IP address and spoofed user's IP address and it also causes excess time delay to detect and filter an attack. Patrick P.C. *et al*. [17] proposes an online early detection algorithm based on the statistical CUSUM method for detecting signalling DoS attacks on wireless networks in a timely manner. This approach does not detect the attack traffic that has a spoofed IP address and causes signaling load on the control plane. This mechanism detects signaling DoS attacks by monitoring inter-setup time samples and blocks both benign and malicious traffic when the signaling load reaches a threshold. Supranamaya Ranjan et al. [22] proposed DDoS-Shield to detect the attack packets that overwhelm the system resources such as bandwidth. DDoS Shield consist of a suspicion assignment mechanism that examines requests belonging to every session (TCP,UDP,ICMP) and assigns suspicion values to sessions and a DDoS-resilient scheduler that schedules the sessions based on the values assigned to the sessions and decides which session to be forwarded and when. The scheduler also performs rate-limiting. DDoS shield improves the victim's concert by consuming less memory for buffering requests and responses. However DDoS Shield consumes more processing time and cannot produce good throughput.

Joseph Chee Ming Teo et al. [14] proposes a group key agreement protocol to protect heterogeneous networks against DoS attacks. But it causes more communication overhead in heterogeneous networks. Wei Chen et al. [23] proposes a storage-efficient data structure

and a change-point detection method to distinguish complete three-way TCP handshakes from incomplete ones. This mechanism leads to large memory consumption. Sungwon Yi et al. [15] introduced a two-level cache Content Addressable Memory (CAM) to dynamically detect and quarantine the unresponsive TCP flows [18]. But it leads to large memory comsumption. Dimitris Geneiata et al. [7] proposed a two-part bloom filter based monitor to detect and filter flooding attacks against proxy servers. The monitor's main task is to record the state of any incoming session in 3 different filters and the filter is indexed through a hash function. This mechanism uses an alarming system to trigger an alarm and report if any entries in the filter exceed the threshold value. This mechanism is very efficient and cost-effective and causes reduced time delay to detect an attack. However, hashing of entries in the filters leads to computation overhead and more CPU utilization. Dimitris Geneiatakis et al. [6] proposes a new header to overcome signaling DoS attacks in SIP servers. But the scheme uses a pre-shared key which when explored leads to password-based attacks and also it is vulnerable to man-in-the-middle attacks. It is observed in [9] that collaborative flooding attacks (DDoS) depend heavily on IP spoofing; therefore clever IP spoof detection might contribute to solving the problem. A common way for preventing IP spoofing is by using ingress and egress filters on firewalls [19]. But it fails in wireless networks where legitimate packets could have topologically incorrect addresses. In this paper, we have introduced a spoof-based collaborative detection of collaborative flooding attack (DDoS).

The proposed system

The ColShield system

The *ColShield* system (Figure 1) uses a semi-centralized architecture maintaining a group of local IPDS that is installed near the local routers and a global IPDS that is installed

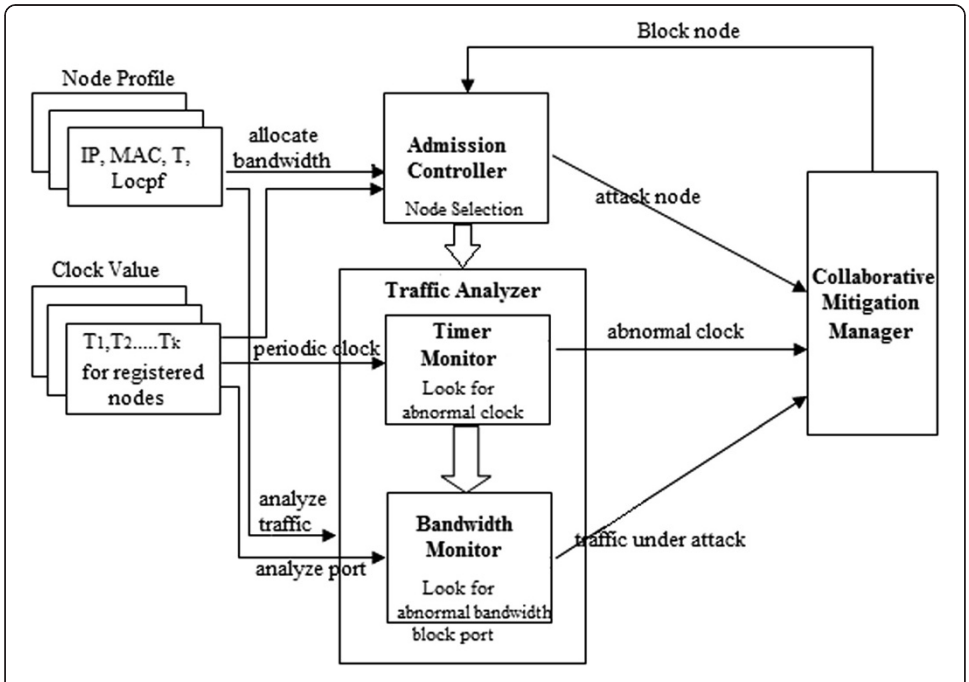

Figure 1 *ColShield* **architecture.**

near the gateway router. This paper focuses on spoof-based collaborative detection of collaborative flooding DDoS attacks. The *ColShield* system consists of four main components which mutually involve in mitigating collaborative flooding DDoS attacks. The Figure 1 shows the architectural view of the *ColShield* system. The ColShield components are described as follows: The *admission controller* is responsible for allocating initial bandwidth for each node using a bandwidth allocation algorithm. The *admission controller* accepts the node that completes the registration process successfully. The nodes have to initially register with the network by sending few confidential information. At the end of registration process, the *admission controller* allocates a bandwidth b_n and a bandwidth validity time, i.e., *TTL* for each node. The *traffic analyzer* component comprises of two components namely the *timer monitor* and the *bandwidth monitor*. The *timer monitor* maintains the clock values [21,25] being sent periodically by each node. These clock values are compared with the threshold value. The nodes that match the threshold value are forwarded to the *bandwidth monitor* for analyzing the traffic abnormalities. Finally, the *admission controller*, the *timer monitor* and the *bandwidth monitor* altogether informs the *collaborative mitigation manager* about their observation in abnormalities of each node. The *collaborative mitigation manager* decides whether to accept or to reject the node and its traffic. However, since the entire traffic cannot be possibly monitored altogether by a single global IPDS component, we promote the usage of multiple IPDS components for efficient detection and filtering of the attack.

The global IPDS maintains a node profile which consist of the following information namely the client node's IP address, the client node's MAC address, the client node's timer value, the client node's location proof information [30], the client node's allotted bandwidth and the TTL value. The global IPDS also maintains a local profile which consists of the IP address of the local IPDS, the total number of client nodes connected to it and its neighboring local IPDS. The local IPDS maintains a profile which consists of the timer values of each client node, the number of flows within each client node, its corresponding port number and the corresponding client node to which the flow is being transmitted or received.

Clever spoof detection

IP spoofing [10] is the main gateway for collaborative DoS attacks [9] which is considered as a most complex attack in which the attackers create raw IP packets with valid IP and TCP headers. An attacker might spoof a single source address or multiple source addresses. It is a difficult task for the listener to detect and filter the spoofing attacks with multiple source addresses than detecting spoofing attacks with single source address. Spoofing attacks can be prevented by using network ingress filters [3,12,16] and egress filters in proper network locations. *IP Security* (IPsec) also provides an excellent defense against IP spoofing, but this protocol generally cannot be required because its deployment is currently not suitable to work with wireless mesh networks [32]. Filtering does not solve the problem of collaborative flooding of DoS attacks and it is a quite challenging task to block spoofing attacks with multiple source addresses. Hence clever spoof detection is necessary to mitigate collaborative DoS attacks. The clever spoof detection process is depicted in Figure 2 and it is carried out in two phases. The admission controller initiates the detection process in phase 1 (Algorithm 1) and timer manager completes the detection process in phase 2 (Algorithm 2). During phase 1, the

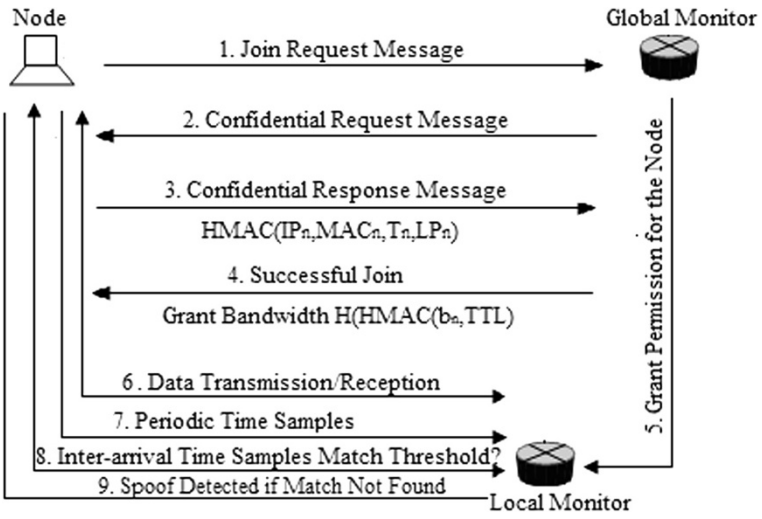

Figure 2 Clever spoof detection protocol.

bandwidth allocation is done for each node and in phase 2, the inter-arrival time samples are monitored for each node. We monitor the inter-arrival time samples at each node in order to detect the presence of IP spoofing in wireless mesh networks thereby providing a way to mitigate collaborative flooding attacks.

Admission controller

We model the backbone of the wireless mesh network (WMN) R as a directed graph $G = (V, E)$ where V represents the set of client nodes in the network and E represents set of directed links. $V = N + M$ where $N = n_1, n_2, \ldots n_r$ is the set of registered nodes in the network and each client node $n \notin N$. M is the set of monitor nodes in the network and it is represented as $M = G_m + \{L_m\}$ where G_m represents the global IPDS and L_m represents the local monitor. The network consists of a group of intrusion protection and detection systems (IPDS) with a single global IPDS, G_m and a cluster of local IPDS L_m. Each client node n before it joins a network has to send a join request message, $R_j(n)$ to the global IPDS G_m. The G_m requests a confidential message $REQ_c(n)$ to client node n to prove its identity. The client in turn replies with its confidential message $RES_c(n)$ to the G_m. The confidential reply message consists of four pieces of information namely, IP address of the client node IP_n, MAC address of the client node MAC_n, Timer value of the client node $z_n(t_i)$ and LP_n, the location proof information [33] of the client node which refers to the actual distance of the client node n from the global IPDS G_m. $z_n(t_i) = z_n(t_c) + K_{sec}$ where $z_n(t_c)$ is the client node's current time and K_{sec} is the client node's secret key. The length of the secret key K_{sec} is 16 bits and its initial value is obtained by adding the least significant 8 bits of IP address with the least significant 8 bits of MAC address along with a 16 bit random number. These 32 bits are hashed into a 16 bit secret key value which forms the length of K_{sec}. The subsequent values of K_{sec} is incremented by 1 bit from the initial value every t_i time interval. The LP_n value is obtained by adding the client node's current distance from the global IPDS D_n with the client node's current available time $z_n(t_c)$. Thus if a client node wants to

prove its identity, the IP_n, MAC_n and LP_n values should match $z_n(t_i)$. The G_m by checking the validity of confidential information, replies with a successful join and grants a bandwidth b_n along with TTL to the client node n. TTL is the bandwidth validity period for client node n. The client node after receiving the bandwidth becomes a part of the network. In this phase, the initial stage of spoof detection is done.

Algorithm 1: *Bandwidth Allocation Algorithm*

$Input$: $set\ of\ client\ nodes\ n\ that\ have\ not\ yet\ registered.$
$i.e., n \notin N.$
$Output$: $bandwidth\ allocation\ b_n\ for\ client\ node\ n.$
$Step\ 1$: $Client\ node\ n\ sends\ a\ join\ request\ message\ R_j$
$to\ G_m$.
$Step\ 2$: $The\ G_m\ asks\ for\ confidential\ request\ message$
$REQ_c\ from\ client\ node\ n.$
$Step\ 3$: $Client\ node\ n\ replies\ with\ a\ confidential\ response$
$RES_c\ to\ the\ G_m.$
$Step\ 4$: $If\ IP_n, MAC_n, LP_n \Longrightarrow z_n(t_i)\ then\ return\ true(n)$
$i.e., client\ node\ n\ is\ not\ spoofed.$
$Step\ 5$: $If\ IP_n, MAC_n, z_n(t_i), LP_n \Longrightarrow true(n) then$
$grant(H(HMAC(b_n, TTL))$
$Step\ 6$: $Client\ node\ n\ successfully\ joins\ the\ network.$
$Now\ n \in N.$

Traffic analyzer

Each *ColShield* IPDS analyzes the traffic within its detection window range. The traffic analyzer consists of two components of which the timer monitor completes the spoof detection process and the bandwidth monitor [35] initiates the flood detection process (Algorithm 3). The timer monitor involve in checking the periodic timer values of each mesh client node. Each mesh client node after joining the network is under the control of the local IPDS. The registered mesh client node, in order to prove its identity to the local IPDS sends periodic timer values to its local IPDS, i.e., L_m. The timer values are the inter arrival time samples of each mesh client node being sent periodically. The local monitor checks the validity of the client node by comparing whether the subsequent inter-arrival timer values match the threshold. The local IPDS concludes the client node as abnormal if the inter-arrival timer values did not match the threshold value by which spoofed node is detected. The timer monitor is described by a timer function,

$$q_n = \max(E(z_n(t)), z_n(t_i)) \tag{1}$$

where $E(z_n(t)$ is the determined threshold value for node n and $z_n(t_i)$ is the actual real-time timer value of node n to be compared with. If $z_n(t_i) = E(z_n(t))$ then $q_n = 0$ and the timer value of node n is benign. If $z_n(t_i) \neq E(z_n(t))$ then the timer value of node n is suspected to be malicious and has to undergo a condition check to confirm the attack. $z_n(t_i)$ values can exceed within an upper limit α and a lower limit β where

α and β are pre-specified constant parameters and $\alpha = \beta = 1$. If the value of $z_n(t_i)$ is greater than $E(z_n(t))$ then the $z_n(t_i)$ value for node n is considered to be malicious if it exceeds the α value. (i.e.) $z_n(t_i) + \alpha = E(z_n(t))$. Likewise, if the $z_n(t_i)$ value is less than $E(z_n(t))$ then the $z_n(t_i)$ value for node n is considered to be malicious if it exceeds the β value. (i.e.) $z_n(t_i) - \beta = E(z_n(t))$. The local IPDS L_m monitors the periodic time samples of all nodes at a given time slot t_i. For a node n the actual real-time timer values is given as,

$$\sum_{\substack{i=1 \\ n \in N}}^{r} z_n(t_i) = \sum_{\substack{i=1 \\ n \in N}}^{r} (z_n(t_i) - \beta) || (z_n(t_i) + \alpha) \quad (2)$$

$z_n(t_i)$ value can be further expressed as,

$$\sum_{\substack{i=1 \\ n \in N}}^{r} Az_n(t_i) \geq \sum_{\substack{i=1 \\ n \in N}}^{r} z_n(t_i) \leq \sum_{\substack{i=1 \\ n \in N}}^{r} Bz_n(t_i) \quad (3)$$

where, $\sum_{\substack{i=1 \\ n \in N}}^{r} Az_n(t_i) = \sum_{\substack{i=1 \\ n \in N}}^{r} (z_n(t_i) - \beta)$ and $\sum_{\substack{i=1 \\ n \in N}}^{r} Bz_n(t_i) = \sum_{\substack{i=1 \\ n \in N}}^{r} (z_n(t_i) + \alpha)$.

The bandwidth monitor has the responsibility to monitor the bandwidth consumption of each client node. During this phase, the local IPDS involve in detecting flooding attacks. The bandwidth monitor categorizes the traffic flow as normal and abnormal. The traffic is said to be normal if the amount of bandwidth consumption adhere to the limit and abnormal traffic consumes a higher bandwidth than the limit. The bandwidth consumption in the sense includes the bandwidth consumed by a single node, per-node per-flow bandwidth and per-node multiple-flow bandwidth. We consider the bandwidth allocation for the global and local IPDS to be stable and predefined. Our aim is to allocate bandwidth for each client node $n \notin N$ and to monitor whether each client node utilizes their allotted bandwidth. Let I_u be the bandwidth update interval which is the time between the last bandwidth allocation and current bandwidth reallocation for each client node. Each client node is permitted to utilize only their allotted bandwidth. Nodes failing to use b_n might have been deviated to $b_{n'}$. The deviation of b_n and $b_{n'}$ must not exceed ϖ. The local IPDS checks whether the fraction of bandwidth allotted for each client node is normal. The local IPDS does this by using the formula, $b_n \leq B_r / N$ where B_r is the total bandwidth allotted to the mesh client nodes in the network. The local IPDS checks whether the fraction of bandwidth utilized per-flow during a single time interval by each client node is within the allotted bandwidth. The per-node per-flow bandwidth is given by, $b_{nf} \leq b_n / C_n$ where C_n is the number of flows established between a mesh client node and another. The local IPDS also checks whether the fraction of bandwidth consumption for all flows per-node during subsequent time intervals. The per-node multiple-flow bandwidth is given by,

$$\sum_{\substack{1 \leq t \leq r \\ 1 \leq f \leq k}} b_{nft} \leq b_n / C_n \quad (4)$$

where f represents the number of flows established between a mesh client node and another node and t represents the time interval of the allotted bandwidth. If any abnormalities were found, the local IPDS detects the attacker node and its port number.

Algorithm 2: *Timer Monitor Algorithm*

$Input: Timer\ values\ of\ each\ clientnode\ z_n(t_i)$
$i = 1 \ldots . k\ and\ n \in N$
$Output: Boolean\ value\ true(n)\ or\ false(n)$
$Timer\ function: q_n = \max(E(z_n(t)), z_n(t_i))$

1. $\forall_{\substack{i=1\ldots.r \\ n\in N}}\ If\ z_n(t_i) = E\big(z_n(t)\big)\ then$
2. $\quad return\ true(n)$
3. $Else\ if\ z_n(t_i) \neq E\big(z_n(t)\big)\ then$
4. $\quad If\ z_n(t_i)) < E(z_n(t))\ then$
5. $\quad\quad q_n = E(z_n(t))$
6. $\quad\quad If\ Az_n(t_i) = E\big(z_n(t)\big)\ then$
7. $\quad\quad\quad return\ true(n)$
8. $\quad\quad Else$
9. $\quad\quad\quad return\ false(n) => spoof\ detected$
10. $\quad\quad Endif$
11. $\quad Else\ if(z_n(t_i)) > E(z_n(t))\ then$
12. $\quad\quad q_n = z_n(t_i)$
13. $\quad\quad If\ Bz_n(t_i) = E(z(t))\ then$
14. $\quad\quad\quad return\ true(n)$
15. $\quad\quad Else$
16. $\quad\quad\quad return\ false(n) => spoof\ detected$
17. $\quad\quad Endif$
18. $\quad Endif$
19. $Endif$

Algorithm 3: *Bandwidth Monitor Algorithm*

$Input: Set\ of\ registered\ client\ nodes\ n \in N$
$b_n, bandwidth\ fraction\ allotted\ to\ node\ n.$
$Output: Boolean\ value\ true(n)\ or\ false(n)$

$Step\ 1: If\ b_n \leq B_r/N \rightarrow true(n)\ then\ go\ to\ step\ 4.$

$Step\ 2: If\ b_n \leq B_r/N \rightarrow false(n) then\ go\ to\ step\ 3.$

$Step\ 3: If\ dev(b_n, b_n') \leq \varpi\ \rightarrow true(n)\ then\ go\ to\ step\ 4\ 4$
$Else\ go\ to\ step\ 6.$

$Step\ 4: If\ b_{nf} \leq b_n/C_n\ then\ return\ true(n)$
$Go\ to\ step\ 5$

$Step\ 5: If\ \sum_{\substack{1\leq t\leq r \\ 1\leq f\leq k}} b_{nft} \leq b_n/C_n\ then\ r\ eturn\ true(n)$

$Step\ 6: return\ false(n) \Longrightarrow flood\ detected$

Collaborative mitigation

We focus on spoof-based collaborative mitigation of collaborative flooding DDoS attacks (Algorithm 4) [36]. All the local IPDS and the global IPDS collaboratively involve in mitigating the flooding attacks (Algorithm 5). The local IPDS L_m executes the bandwidth monitoring algorithm for detecting the attacker client node. Once it detects the attacker client node, it first blocks the port number under attack and then blocks the future traffic to and from the specified port number. It then informs the neighboring local IPDS NL_m about the

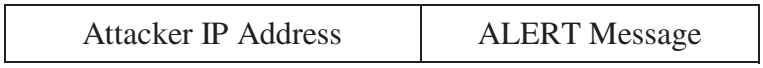

Figure 3 ALERT message format.

attacker client node by sending an ALERT message which contains the IP address of the attacker client node and an ALERT message which is depicted in Figure 3. Now the local IPDS along with its neighbors inform the global IPDS about the attacker. When the global IPDS receives the ALERT message, it blocks future traffic to and from that client node under attack and revokes the allotted bandwidth from that client node. Now the client node under attack is released from the network and it cannot communicate with the nodes in the spoofed network. Thus flooding attack is collaboratively mitigated in this phase. Again if the released node wishes to join the network, it has to re-register and obtain new bandwidth from the network. The attacker in any case cannot bypass the bandwidth monitor test and thus it fails which leads to repeated re-registration process. The effectiveness of *ColShield* lies with the traffic analyzer which aims at analyzing abnormal traffic from the client nodes. Our paper focuses on detecting spoof-based collaborative flooding attacks (i.e., detecting collaborative flooding attacks that occur through IP spoofing). *ColShield* can detect 85% of spoofed nodes and once spoofing attacks are detected, collaborative flooding attacks are easily detected and mitigated because collaborative flooding attacks don't have much effect on spoof free nodes.

Algorithm 4: *Collaborative Mitigation Algorithm*

$Input: f_n, incoming\ traffic\ flow\ at\ node\ n$
$z_n(t_i), the\ periodic\ timer\ values$
$i = 1 \ldots . r\ and\ n \in N$
$Output: DDoS\ free\ network$
$Step\ 1: If\ z_n(t_i) \rightarrow false(n)\ then$
$Step\ 2: L_m \rightarrow block\ (IP_n, P_n)$
$Step\ 3: Else\ if\ f_n \rightarrow false(n)\ then$
$Step\ 4: L_m \rightarrow block\ (f_n)$
$Step\ 5: signal(alert(L_m(n))) \rightarrow recv(NL_m)$
$Step\ 6: signal(alert(L_m(n), NL_m(n))) \rightarrow recv(G_m)$
$Step\ 7: G_m \rightarrow block\ (n)$
$Step\ 8: G_m \rightarrow remove\ (n)$
$Step\ 9: Endif$
$Step\ 10: Endif$

ColShield metrics

ColShield maintains the following metrics:

1) *Traffic flow metric:* This metric helps to calculate the total number of communications taken place in the network when we install the ColShield system in the network. The total traffic flow at the global IPDS is given by,

$$f(G_m) = \sum_{m=1}^{i} f_{out}(L_m) \tag{5}$$

where $f_{out}(L_m)$ is the sum of all outgoing traffic flow coming out from all the local IPDS. All mesh client nodes has to pass through the local IPDS to send and receive messages. Therefore, the total traffic flow at the local IPDS is obtained by adding the total incoming and outgoing traffic flow at each mesh client node. The total traffic flow at the local IPDS is given by,

$$f(L_m) = \sum_{n \in N} f_{in}(n) + \sum_{n \in N} f_{out}(n) \tag{6}$$

where f_{in}(n) is the client node's incoming traffic and f_{out}(n) is the client node's outgoing traffic. The total traffic flow at the mesh client nodes is given by,

$$f(n) = \sum_{c=1}^{i} f_c(n) + \sum_{d=1}^{i} f_d(n) \tag{7}$$

where f_c(n) is the client node's control flow traffic and f_d(n) is the client node's are the control flow traffic and data flow traffic at the mesh client nodes.

The control flow traffic at the mesh client node n is given by,

$$f_c(n) = f_{cin}(n) + f_{cout}(n) \tag{8}$$

where $f_{cin}(n)$ is the client node's incoming control flow traffic and $f_{cout}(n)$ is the client node's outgoing control flow traffic. The data flow traffic at the mesh client node n is given by,

$$f_d(n) = f_{din}(n) + f_{dout}(n) \tag{9}$$

where $f_{din}(n)$ is the incoming data flow traffic at the client node and $f_{dout}(n)$ is the outgoing data flow traffic at the client node. The total number of control messages exchanged between the mesh clients, the local IPDS and the global IPDS are required to calculate the communication overhead.

2) *Throughput metric:* The proposed system guarantees a minimum throughput of λ and all client nodes should adhere within this throughput. i.e.,

$$\sum_{n \in N} b_n \leq \lambda \tag{10}$$

The throughput is affected by the fraction of bandwidth allocated to each client node. The client nodes for which the bandwidth is allocated through the bandwidth allocation protocol are considered for achieving wireless mesh network throughput.

3) *Bandwidth allocation metric:* b_n is the fraction of bandwidth allotted to each client node $n \in N$ and $B_r = B - B_{mb}$ where B is the total bandwidth allotted to the network, B_{mb} is the bandwidth allotted for the local and global IPDS and B_r is the bandwidth allotted to each mesh client nodes who joins the network. The bandwidth constraint is given by,

$$b_n \leq B_r / N \tag{11}$$

4) *Bandwidth deviation metric:* The bandwidth deviation metric is given by,

$$dev(b_n, b_{n'}) \leq \varpi \tag{12}$$

Each client node is allotted a bandwidth b_n within the network and they are permitted to utilize only their allotted bandwidth. Nodes failing to use b_n might

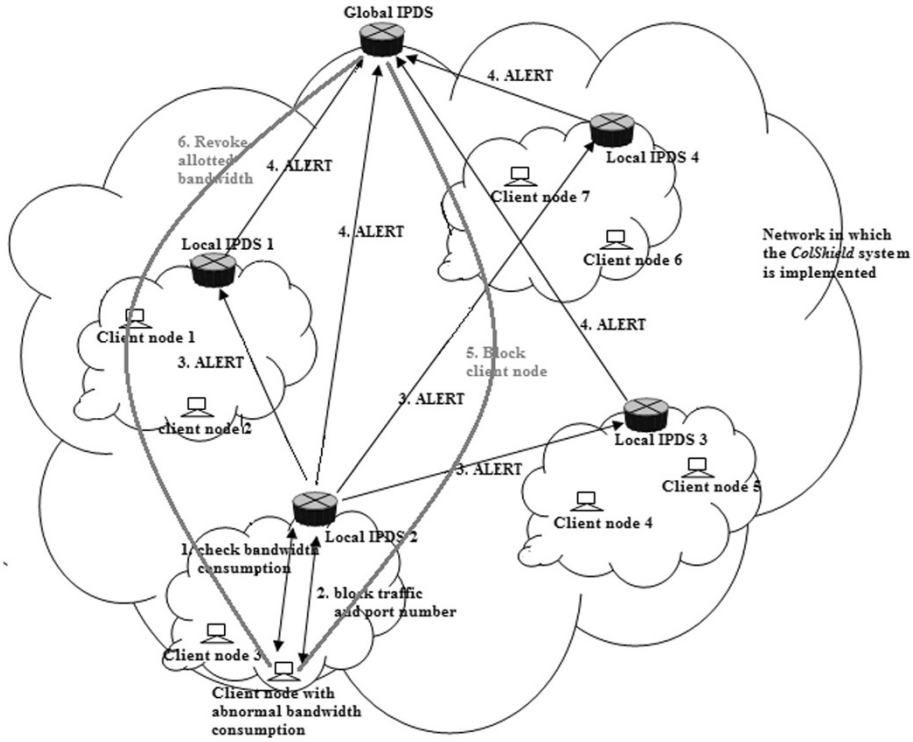

Figure 4 Collaborative mitigation protocol.

have been deviated to $b_{n'}$. The deviation of b_n and $b_{n'}$ must not exceed ϖ whose value is 0.1. If the deviation exceeds ϖ then it leads to rejection of that client node.

Performance results

We used NS-2 simulator for implementing WMN model for security [37,38] against collaborative flooding attack (DDOS). The model is adapted from the IEEE 802.11b/g based adhoc network including the mesh clients that are mobile and backbone mesh routers that are stable. The hierarchical architecture of the WMN was implemented using administrative domain (AD) cluster design. In the model, a gateway router was statically assigned as global IPDS and the local routers are statically assigned as local IPDS while the client nodes are enabled using random waypoint wireless model as mesh clients. In addition, the gateway router is assigned as back-bone router. The adhoc network security standard IEEE802.11i was used for simulation due to the ongoing standardization of WMN security. We have compared the performance of *ColShield* with *FireCol*. We use the following metrics for evaluating the performance of *ColShield*: 1) false positive ratio 2) detection time 3) packet delivery ratio and 4) communication overhead 5) average throughput 6) bandwidth consumption and 7) registration overhead.

1) *Packet delivery ratio (PDR):* It is the ratio of the total number of packets delivered to the mesh client to the total number of packets received at the local IPDS. The local IPDS delivers those packets that wins the timer manager protocol and the bandwidth monitor protocol. Figure 5 shows the packet delivery ratio of *ColShield* with respect to the percentage of local IPDS. The PDR is reasonably good and does not affect the performance of the network.

Figure 5 Packet delivery ratio over percentage of local IPDS.

2) *False positive ratio:* The false positive rate is the amount of legitimate traffic wrongly detected as malicious. Since each IPDS store the full TCP connection information, it can have false rates. However, this will not affect the final detection behavior. Figure 6 shows the false positive rates of *FireCol* and *ColShield* with respect to the percentage of local IPDS. The false positive ratio is roughly increased to 5% which is acceptable and does not affect the final detection results.
3) *Attack Detection Time:* The attack detection time is the delay between the attack occurs and when it is detected. The detection of flooding attack [39,40] is based on detection of increase in a client node's clock inter-arrival times. Figure 7 shows the detection delay for *FireCol* and *ColShield*. The *ColShield* can detect the start of the attack within one detection time interval and end of an attack within two detection time periods. The proposed method can achieve more accurate detection with a shot latency. When the percentage of local IPDS increases, the attack detection time is less.

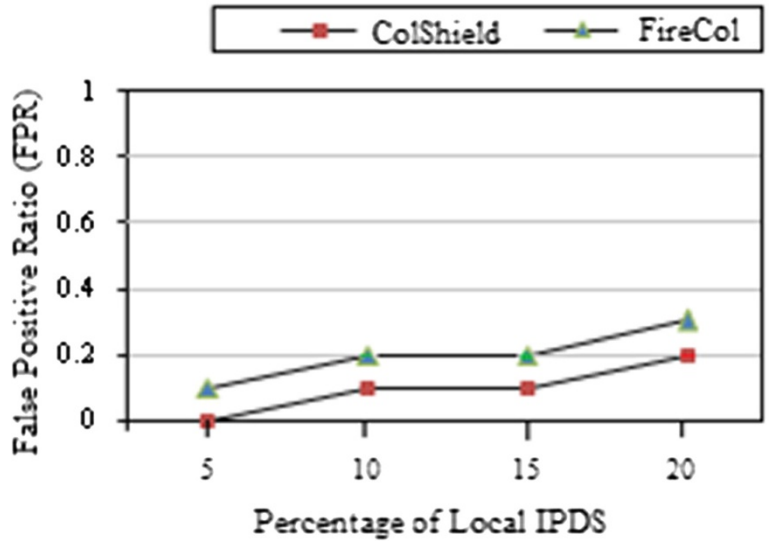

Figure 6 Comparison of false positive ratio.

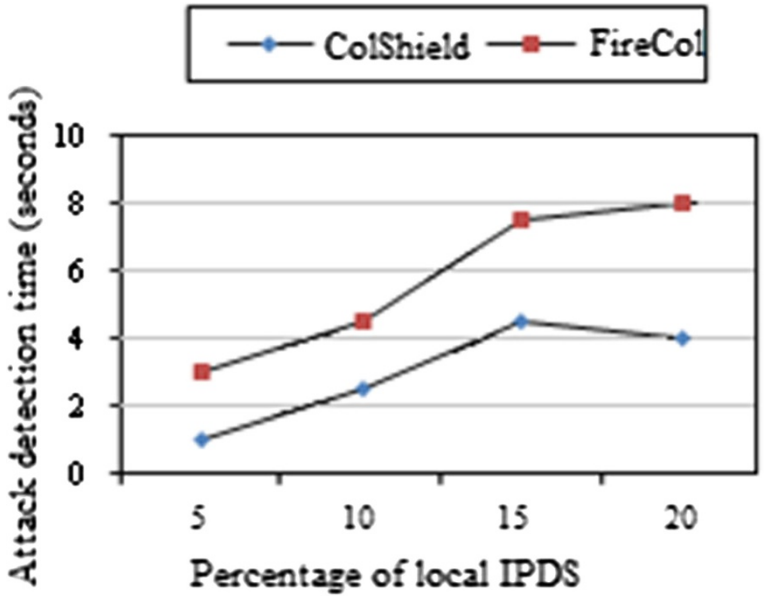

Figure 7 Comparison of attack detection time.

4) *Communication overhead:* It is the total number of control messages exchanged between the mesh clients, the local IPDS and the global IPDS. Compared to the mesh client level and the gateway router level, the maximum number of communications take place at the mesh router level. The communication overhead is obtained by summing up the total traffic flow at the global IPDS and the local IPDS. The communication overhead for *ColShield* is depicted in Figure 8. The figure shows the percentage of data messages and control messages being transmitted in the wireless mesh network. Only 20% of control messages are exchanged within the system which is comparetively less than the total number of data packets exchanged in the system. The communication overhead does not affect the performance of the network.

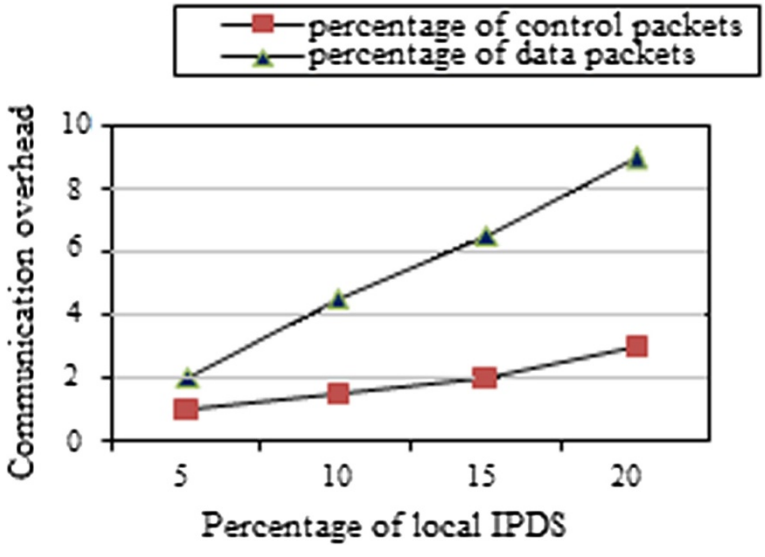

Figure 8 Communication overhead over percentage of local IPDS.

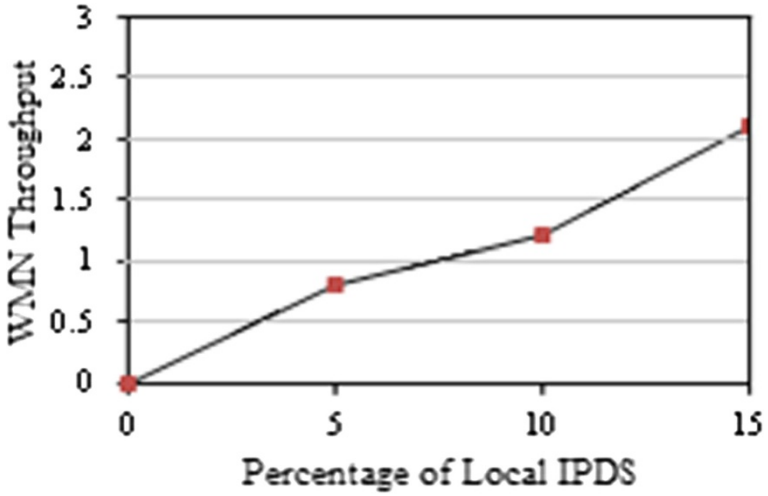

Figure 9 WMN throughput over number of local IPDS.

5) *WMN throughput*: It is defined as the sum of the data delivered to all the client nodes in the network in a given time unit (seconds). The throughput is affected by the fraction of bandwidth allotted to each client node in the network. Figure 9 shows the WMN throughput with respect to the percentage of local IPDS. The client nodes that obtain bandwidth through the bandwidth allocation process are eligible for achieving WMN throughput.
6) *Attack detection ratio:* It is the rate at which the spoofing attacks and the flooding attacks are detected. When the network size increases the percentage of local IPDS increases which leads to the increase in attack detection ratio. Once the spoofing attacks are detected, the flooding attacks are detected easily in a timely manner. Figure 10 shows the attack detection ratio of the *ColShield* system with respect to the percentage of local IPDS in the WMN. The attack detection ratio calculates the

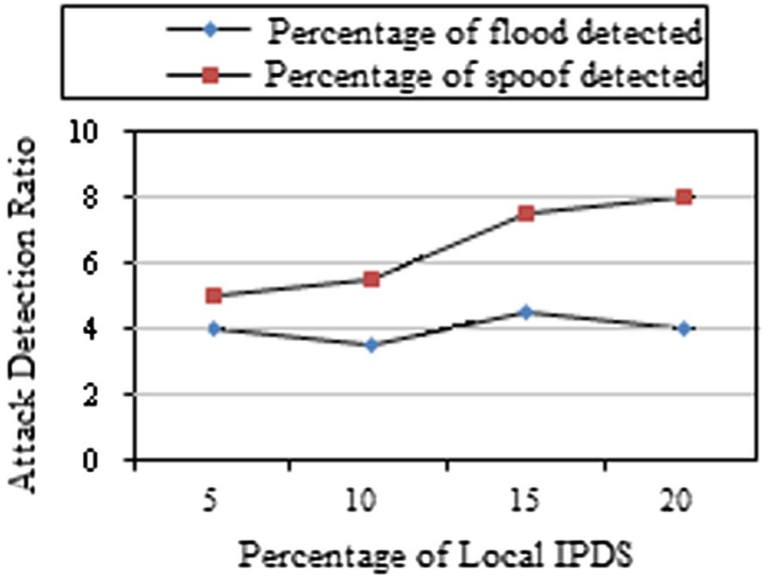

Figure 10 Attack detection ratio over percentage of local IPDS.

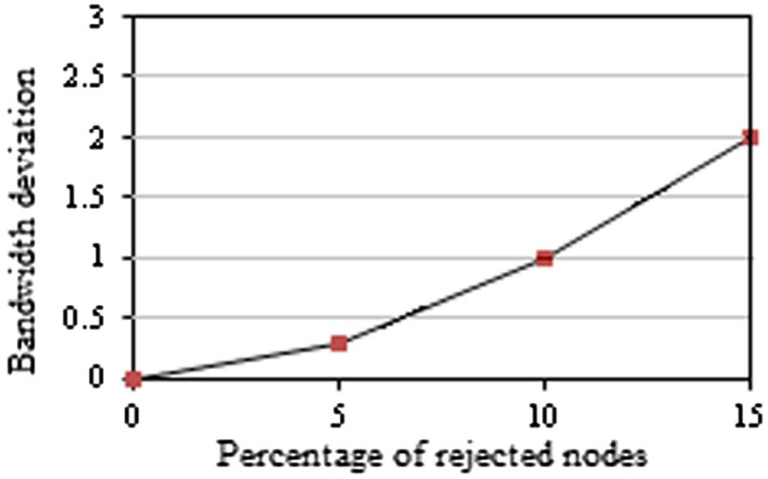

Figure 11 Bandwidth deviation over percentage of rejected nodes.

percentage of spoofing attack detected by the system while running the bandwidth allocation process and the timer monitor protocol. The attack detection ratio also calculates the percentage of flooding attacks detected by the system while running the bandwidth monitor protocol. The attack detection ratio for spoofing attack and flooding attack is reasonably good which is the goal of the *ColShield* System.

7) *Bandwidth deviation:* It is the fraction of deviated bandwidth from the allotted bandwidth. Figure 11 shows the percentage of bandwidth deviation with respect to the percentage of rejected nodes. The percentage of rejected nodes increases when they cross the threshold value ϖ. It is strictly followed that nodes that have a bandwidth deviation beyond the threshold value are rejected.
8) *Registration overhead:* The number of communications and the number of computations required by a client node during the registration process determines the registration overhead (Figure 12). A single client node communicates four messages to complete the registration process. But each node requires X-OR computations for a single timer value to complete the registration process, which is reasonable. The

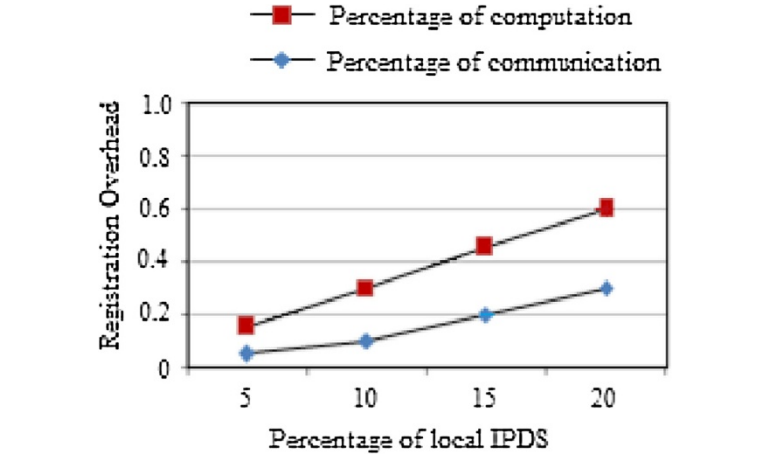

Figure 12 Registration overhead over percentage of local IPDS.

computations are required in all phases to continually monitor the registration of client nodes, to allocate bandwidth as well as to analyze traffic and to effectively mitigate spoof-based collaborative flooding attacks. The computation overhead can be balanced by placing reasonable number of local IPDs in the network. As the number of local IPDS increases in the network, the computation overhead is tolerated to 60%.

ColShield effectiveness relies on the collaboration between different IPDS. The *ColShield* cannot be enabled on all routers. The IPDS are routers that perform detection and forward messages to the neighboring routers and the global IPDS. An IPDS communicates with neighboring IPDS only for signaling collaborative information. Thus only 20% of communication overhead is caused in the network which does not affect the performance of the network.

Conclusion

In this paper, we propose an effective flood detection and prevention architecture, *ColShield* to detect flooding attacks and also report the specific victim client node and port being attacked. *ColShield* does not give any chance for an attacker to evade the detection. The time taken to detect the start of an attack is less than one detection interval and the time taken to detect the end of an attack is less than two detection intervals. Through simulations, it is demonstrated that *ColShield* is the fastest and most accurate detection method compared with *FireCol*.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

I.Diana Jeba Jingle carried out the studies on wireless mesh networks and distributed denial-of-service attacks, designed the architecture and algorithms, carried out the simulation experiments and drafted the manuscript. Elijah Blessing Rajsingh provided full guidance and support to prepare the manuscript. All authors read and approved the final manuscript.

Authors' information

I. Diana Jeba Jingle is the Assistant Professor of Loyola Institute of Technology and Sciences, India. She reveived her Bachelor of Engineering degree in Information Technology from Sun College of Engineering and Technology, Anna University, India in 2006. She received her Master of Engineering degree in Computer Science from Francis Xavier Engineering College, Anna University, India in 2008. Currently she is pursuing her Ph.D in Karunya University, Coimbatore, India. Her research interests lie in the area of Wireless Networks, Mobile Ad-hoc Networks and network security and specifically focus on denial-of-service characterization, detection and defense, IP spoofing defense. She is a member of CSI.
Elijah Blessing Rajsingh is the Professor and Director for the Department of Computer Science and Engineering of Karunya University, India. He received his Master of Engineering degree with Distinction from the College of Engineering, Anna University, India. He received the Ph. D degree in Information and Communication Engineering from the College of Engineering, Anna University, India in 2005, focusing on Security in Wired and Wireless Networks. He is the member of IEEE. He has very strong research background in the areas of Network Security, Mobile Computing, Wireless & Ad hoc Networks, Parallel and Distributed Computing. He is an Associate Editor for International Journal of Computers & Applications, Acta Press, Canada and member of the editorial review board for International Journal of Cases in E Commerce as well as for Information Resources Management Journal, Idea Group Publishers, USA. He is the recognized guide for Ph.D students of Karunya University and is guiding students in their doctoral programme. He has published a number of papers in well-referred international journals and conferences.

Acknowledgement

We would like to thank the reviewers for their valuable comments.

Received: 10 December 2013 Accepted: 15 April 2014
Published online: 06 September 2014

References

1. François J, Aib I, Boutaba R (2012) Firecol: a collaborative protection network for the detection of flooding Ddos attacks. IEEE/ACM Trans Networking 20(6):1828-1841
2. Sun C, Hu C, Liu B (2012) SACK2: effective SYN flood detection against skillful spoofs. IET Inf Secur 6(3):149–156

3. Shevtekar A, Anantharam K, Ansari N (2005) Low rate TCP denial-of-service attack detection at edge routers. IEEE Commun Lett 9(4):363-365
4. Xiaoa B, Chenb W, Hec Y (2008) An autonomous defense against SYN flooding attacks: detect and throttle attacks at the victim side independently. J Parallel Distr Comput 68:456–470
5. Cert Advisory Ca-1996-21 (1996) TCP SYN Flooding and IP Spoofing Attacks". CERT CC. https://www.cert.org/historical/advisories/CA-1996-21.cfm?
6. Geneiatakis D, Lambrinoudakis C (2007) A lightweight protection mechanism against signaling attacks in a SIP-based VoIP environment". Telecommun Syst 36(4):153–159
7. Geneiatakis D, Vrakas N, Lambrinoudakis C (2009) Utilizing bloom filters for detecting flooding attacks against SIP based services. J Comput Secur 28(7):578–591
8. Zhang G, Fischer-Hübner S, Ehlert S (2010) Blocking attacks on SIP VoIP proxies caused by external processing". Telecommun Syst 45(1):61–76
9. Safa H, Chouman M, Artail H, Karam M (2008) A collaborative defense mechanism against SYN flooding attacks in IP networks. J Netw Comput Appl 31(4):509–534
10. Mopari IB, Pukale SG, Dhore ML (2009) Detection of DDoS attack and defense against ip spoofing. In: Proceedings of the International Conference on Advances in Computing, Communication and Control, ICAC3'09, Mumbai, Maharashtra, India, PP, pp 489–493
11. Khalil I, Bagchi S, Shroff NB (2008) Mobiworp: mitigation of the wormhole attack in mobile multihop wireless networks. J Ad Hoc Netw 6:344–362
12. Ioannidis J, Bellovin S (2002) Implementing Pushback: Router-Based Defense Against Dos Attacks. In: Proc. NDSS
13. Mirkovic J, Reiher P (2005) D-WARD: A Source-End Defense Against Flooding Denial-Of-Service Attacks. IEEE Trans Dependable Secure Comput 2(3):216-232
14. Chee J, Teo M, Tan CH, Ng JM (2007) Denial-of-service attack resilience dynamic group key agreement for heterogeneous networks". Telecommun Syst 35(3–4):141–160
15. Saxena N, Denko M, Banerji D (2010) A hierarchical architecture for detecting selfish behaviour in community wireless mesh networks. J Comp Commun 548–555
16. Ferguson P, Senie D (2000) Network Ingress Filtering: Defeating Denial Of Service Attacks That Employ IP Source Address Spoofing. Internet RFC 2827
17. Lee PPC, Bu T, Woo T (2009) On the detection of signaling Dos attacks On 3G/Wimax wireless networks. J Comput Netw 53(15):2601–2616
18. Yi S, Deng X, Kesidis G, Das CR (2008) A dynamic quarantine scheme for controlling unresponsive TCP sessions. Telecommun Syst 37(4):169–189
19. Misra S, Dhurandher SK, Rayankula A, Agrawal D (2010) Using honeynodes for defense against jamming attacks in wireless infrastructure-based networks. J Comput Electr Eng 36(2):367–382
20. Misra S, Krishna PV, Abraham KI, Sasikumar N, Fredun S (2010) An adaptive learning routing protocol for the prevention of distributed denial of service attacks in wireless mesh networks. Comput Math Appl 60(2):294–306
21. Jana S, Kasera SK (2010) On fast and accurate detection of unauthorized wireless access points using clock skews. IEEE Trans Mob Comput 9(3):449–462
22. Ranjan, Swaminathan R, Uysal M, Nucci A, Knightly E (2009) DDoS-Shield: DDoS-Resilient scheduling to counter application layer attacks. IEEE/ACM Trans Networking 17(1):26–39
23. Chen W, Yeung D-Y (2006) Throttling spoofed SYN flooding traffic at the source". Telecommun Syst 33(1–3):47–65
24. Yang X, Wetherall D, Anderson T (2008) TVA: A DoS-limiting network architecture. IEEE/ACM Trans Networking 16(6):1267-1280
25. Jana S, Kasera SK (2010) On fast and accurate detection of unauthorized wireless access points using clock skews. IEEE Trans Mob Comput 9(3):449-462
26. Wang H, Zhang D, Shin KG (2002) Detecting SYN flooding attacks". In: Proceedings of IEEE INFOCOM, vol 23, pp 1530–1539
27. Granelli F, Doron E, Wool A (2009) IEEE 802.11s Wireless Mesh Networks: Challenges and Perspectives. In: Proceedings of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), vol 13, pp 263–271
28. Fallah MS (2010) A puzzle-based defense strategy against flooding attacks using game theory. IEEE Trans Dependable Secure Comput 7(1):5–19
29. Mirkovic J, Hussain A, Fahmy S, Reiher P, Thomas RK (2009) Accurately measuring denial of service in simulation and testbed experiments. IEEE Trans Dependable Secure Comput 6(2):81–95
30. Kuo S-P, Kuo H-J, Tseng Y-C (2009) The beacon movement detection problem in wireless sensor networks for localization applications. IEEE Trans Mob Comput 8(10):1326–1338
31. Gao D, Reiter MK, Song D (2009) Beyond output voting: detecting compromised replicas using HMM-based behavioral distance. IEEE Trans Dependable Secure Comput 6(2):96–110
32. Fabio Martignon A, Stefano Paris B, Antonio Capone B (2009) Design and implementation of mobisec: a complete securityarchitecture for wireless mesh networks. J Comput Netw 53:2192–2207
33. Huang D-W, Lin P, Gan C-H (2008) Design and performance study for a mobility management mechanism (WMN) using location cache for wireless mesh networks. IEEE Trans Mob Comput 7(5):546-556
34. Hwang K, Cai M, Chen Y, Qin M (2007) Hybrid Intrusion detection with weighted signature generation over anomalous internet episodes. IEEE Trans Dependable Secure Comput 4(1):41-55
35. Chen S, Song Q (2005) Perimeter-based defense against high bandwidth DDOS attacks. IEEE Trans Parallel Distrib Syst 16(6):526-537
36. Ehud D, Avishai W (2010) WDA: a web farm distributed denial of service attack attenuator. J Comput Netw 1037–1051
37. Muogilim OE, Loo K-K, Comley R (2011) Wireless mesh network security: a traffic engineering management approach. J Netw Comput Appl 34(2):478–491
38. Dong J, Ackermann K, Nita-Rotaru C (2009) Secure group communication in wireless mesh networks. J Ad Hoc Netw 7:1563–1576

39. Noh S, Jung G, Choi K, Lee C (2008) Compiling network traffic into rules using soft computing methods for the detection of flooding attacks. J Appl Soft Comput 8:1200–1210
40. Li L, Su-Bin S (2008) Packet track and traceback mechanism against denial of service attacks. J China Univer Posts Telecommun 15(3):51–58
41. Li Q, Trappe W (2006) Reducing delay and enhancing dos resistance in multicast authentication through multigrade security. IEEE Trans Inf Forensics Sec 1(2):190–204
42. Liu Q, Yin J, Leung VCM, Cai Z (2013) FADE: forwarding assessment based detection of collaborative grey hole attacks in WMNs. IEEE Trans Wirel Commun 12(10):5124–5137

doi:10.1186/s13673-014-0008-8
Cite this article as: Jingle and Rajsingh: *ColShield*: an effective and collaborative protection shield for the detection and prevention of collaborative flooding of DDoS attacks in wireless mesh networks. *Human-centric Computing and Information Sciences* 2014 **4**:8.