Future Generation Computer Systems 89 (2018) 685–697

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics

Kshira Sagar Sahoo [a], Deepak Puthal [b,*], Mayank Tiwary [c], Joel J.P.C. Rodrigues [d,e,f,g], Bibhudatta Sahoo [a], Ratnakar Dash [a]

[a] Department of CSE, National Institute of Technology, Rourkela, India
[b] Faculty of Engineering and IT, University of Technology Sydney, NSW, Australia
[c] SAP Labs, Bangalore, India
[d] National Institute of Telecommunications (Inatel), Brazil
[e] Instituto de Telecomunicaes, Portugal
[f] ITMO University, St. Petersburg, Russia
[g] University of Fortaleza (UNIFOR), Brazil

HIGHLIGHTS

- Investigated different control plane security issues of SDN.
- Highlighted the importance of GE metric and implement on incoming packets.
- The results are compared with Shannon metric for performance comparison.
- Our result also compared with other information distance metrics.

ARTICLE INFO

Article history:
Received 21 April 2018
Received in revised form 25 June 2018
Accepted 10 July 2018
Available online 21 July 2018

Keywords:
SDN
Controller
DDoS attack
Information distance
Generalized entropy

ABSTRACT

The primary innovations behind Software Defined Networks (SDN) are the decoupling of the control plane from the data plane and centralizing the network management through a specialized application running on the controller. In spite of many advantages, SDN based data centers' security issues is still a matter of concern among the research communities. Although SDN becomes a valuable tool to defeat attackers, at the same time SDN itself becomes a victim of Distributed Denial-of-Service (DDoS) attacks due to the potential vulnerabilities exist across various SDN layer. The logically centralized controller is always an attractive target for DDoS attack. Hence, it is important to have a fast as well as accurate detection model to detect the control layer attack traffic at an early stage. We have employed information distance (ID) as a metric to detect the attack traffic at the controller. The ID metric can quantify the deviations of network traffic with different probability distributions. In this paper, taking the advantages of flow based nature of SDN, we proposed a Generalized Entropy (GE) based metric to detect the low rate DDoS attack to the control layer. The experimental results show that our detection mechanism improves the detection accuracy as compared to Shannon entropy and other statistical information distance metrics.

© 2018 Published by Elsevier B.V.

1. Introduction

Safeguarding the network security is a rat race process between attackers and victims for many years. Advancement of the technology, open up new attack tools to launch various attacks, consequently, the defenders require sophisticated and up-to-date defense mechanism to countermeasure the attack. As contrasting to other attacks, DDoS attack, can cause a massive interruption in any kind of network infrastructure. With the recent advancement of virtualization-based cloud computing, SDN paradigm is being adopting as an security solutions by the modern data centers [1–3]. In SDN the entire control decisions are made by a separate entity called controller [4]. This decoupling framework brings many benefits to the network management and provides an easy solution to improve the overall network efficiency [5]. As the control plane separate from the data plane, it is believed that a flexible and scalable network can be designed to meet the requirement of the ever-changing business need. On one hand, the programmability and

* Corresponding author.
E-mail addresses: kshirasagar12@gmail.com (K.S. Sahoo), deepak.puthal@gmail.com (D. Puthal), mayank.tiwary@sap.com (M. Tiwary), joeljr@ieee.org (J.J.P.C. Rodrigues), bdsahu@nitrkl.ac.in (B. Sahoo), ratnakar.dash@gmail.com (R. Dash).

https://doi.org/10.1016/j.future.2018.07.017
0167-739X/© 2018 Published by Elsevier B.V.

logically centralized architecture help controller to easily detect the attack, on the other hand the control layer of SDN paradigm itself is likely to targeted by DDoS attacker [6].

All the forwarding decision are taken by the controller and managed by flow tables of OpenFlow switches. There is a search in the flow table for every new packet to the switch. For a successful match, the flow action will carry out. Otherwise, the packet will be sent to the controller for further instructions. The Fig. 1 shows the header fields that are used during matching period. During DDoS attack, spoofing the source IP address is a common practice. In SDN based data center scenario, for spoofed address there is a mismatch each time on the flow table. Therefore, for each unmatched flow, a packet_in is sent to the controller [7]. If the arrival rate of packet_in is very high in case of DDoS attack, the controller resources will start to deplete soon. A high rate IP spoofed may overwhelm the controller and as a result, it disconnects from the data plane. In a centralized SDN controller architecture single point of failure will defunct the entire network [8]. Hence, it is necessary to identify the DDoS traffic from the benign traffic (see Fig. 2).

In anomaly detection usually the threshold is fixed, and any abnormal deviation of some statistical features of the incoming traffic, can help to identify attack traffic [9]. Therefore, the choice of statistical techniques is so vital in case of DDoS detection. The information theory based metrics can identify more accurately the variations of the traffic behavior of such events. In this work, we have extended the idea of [10], and explore the Generalized Entropy (GE) and Generalized Information Divergence (GID) metrics to detect low rate DDoS attacks. The main contribution of the paper are:

- Investigate different control plane security issues of SDN.
- Highlight the importance of GE metric and implement it on the incoming packets coming to the controller for identifying the low rate attack traffic.
- Compare the result with Shannon metric. In addition to this we compare our result with other information distance metrics.
- We simulate the above scenario on Mininet emulator with POX controller.

The rest of the paper is organized as follows. Section 2 describes the motivation behind this work and related work. Section 3 discussed the information metrics used in the work. The detection procedure explains in Section 4. The experimental setup is mentioned in Section 5. The performance of the algorithms and results are well discussed in Section 6. Finally, Section 7 ends with concluding remarks.

2. Motivation and related work

In the last decades, there is significant research carried out on the security of the traditional network [11–13]. Specifically, a number of approaches have been suggested to identify and mitigate the DDoS attack traffic in traditional network [14–18]. Keeping in the view of the future traffic demand, the traditional network was computationally expensive and requiring innovation for the variety of security issues. In this regard, OpenFlow enabled SDN architecture has proven to be successful in various security challenges [19–21]. Despite the innovative solutions provided by the SDN architecture, the security of control layer is not widely explored. The programmability model and the logical centralization introduce new threat vectors which cause the control layer of SDN to become an attractive target for attackers [22]. Especially, the DDoS attacks against the control plane could render unavailable a large part of the network [23].

It is very difficult to identify the DDoS traffic at data plane level because OpenFlow enabled switches have no intelligence to segregate the different types of traffic flows. In addition to this, to make controller as a victim, attackers can cause the attack by using easily available tools and commodity hardwares [35]. In this regard, it is important to detect the DDoS attack against control plane in an earlier stage. Although authors have proposed many solutions, but the methods have some limitations. In this section, we make a review of the state-of-art DDoS detection techniques to address the control plane attack which are listed in Table 1.

Some other well-known approaches are proposed to detect DDoS traffic by utilizing the SDN architecture. A Self-Organizing Maps (SOM) based machine learning (ML) model was used for detecting DDoS attack traffic by Braga et al. [24]. In their work, they have used six different features to identify the attack traffic such as Average Byte per flow (ABf), Average Packet per flow(APf), Average Duration per flow (ADf), etc. Likewise, in [32,36], authors have used different ML techniques to differentiate the traffic flows. An adaptive flow collection method was proposed in [37] for DDoS detection in SDN. A traffic measurement tool called OpenSketch, uses a hash table for measuring the traffic. Instead of flow sampling, this tool uses a three stage pipeline process to collect traffic and identify the malicious traffic from them. Most of the DDoS detection solutions for SDN have used machine learning and knowledge-based techniques to identify the attack traffic.

In general, ML technique differentiates the attack flows based on certain features pertain to traffic characteristics. ML-based anomaly detection techniques are preferred to use in a small network. However, in a large-scale network, the flow collection and statistics process may overload the controller [26]. Additionally, the response time of controller is an important factor during ongoing attack scenario. Moreover, the performance of ML techniques is typically dependent upon the dataset that has been used for training.

On the other hand, some statistical techniques are used for traffic analysis by few authors [8,38,39]. Mousavi et al. proposed an entropy-based technique for early detection of malicious traffic targeting to the POX controller [30]. The limitation fo this work is that, when the number of hosts increases the false positive report. To lower the computation burden of the controller during the detection of attack traffic David et al. have proposed a fast entropy method in a flow-based network [39]. A scheduling based approach has proposed by Lim et al. [29]. In their method, a single processing queue has been subdivided into k number of logical queues and each are belongs to the network switch. During heavy request scenario, the controller utilizes these logical queues to the serves the request with a scheduling manner.

Mehdi et al. uses maximum entropy estimation technique to determine the normal traffic distribution to fix network security problems in home and office networks using SDN [25]. Experimental works were carried out using OpenFlow switches with NOX controller. In this work, authors have used the low rate network traffic to do the experiments as their primary focused was to identify attack traffic in a home environment. In another work, authors have used entropy method to detect the worm propagation, and portscan attacks [26]. Further, an entropy-based anomaly detection scheme was proposed by Wang et al. [8]. In their work, to reduce the burden of the control plane the detection module runs in the edge switches.

A packet_in filtering technique has proposed by Kotani et al. to protect the controller [40]. This technique lists the contents of the packet header fields before the forwarding the packet_in event. On the contrary, if the attacker generates new flows in which the packets have a distinct value from the listed one the technique will be inefficient. Dong et al., designed a detection method to locate the compromised interfaces where the attackers are connected [31]. In anomaly detection usually the threshold is fixed, and any abnormal deviation of some statistical features of the incoming traffic, can

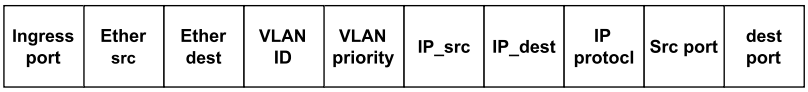


Fig. 1. Packet header list for OpenFlow 1.3v.

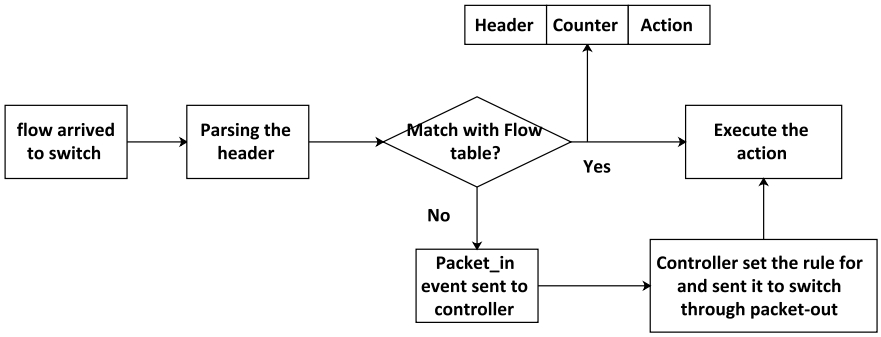


Fig. 2. OpenFlow packet precessing steps.

Table 1
Existing works on DDoS attack detection against SDN control layer.

Works	Year	short analysis	Technique used
[24]	2010	First of its kind, this work presents SDN based DDoS attack techniques and used six different traffic flow features.	SOM, neural network model.
[25]	2011	Proposed various ways to collect traffic and implement different techniques to detect DDoS traffic.	TRW-CB, Rate-limiting, maximum entropy.
[19]	2013	The proposed technique reduces the interaction between control plane and data plane	Connection migration
[26]	2014	Flow-related traffic features used to detect DDoS, portscan attack and worm propagation.	Entropy-based algorithm.
[27]	2014	Experiment was carried on low-rate traffic to detect the DDoS traffic using fuzzy rule inference.	Rate limiting and TRW.
[28]	2014	To prevent control layer attack SVM classifier is used.	Machine learning.
[29]	2015	A scheduling based approach has proposed to handle the heavy request scenario	Redirecting suspected traffic
[8]	2015	DDoS attack detection module runs on edge switch in order to reduce the flow collection made by the controller.	Entropy technique.
[30]	2016	Shannon entropy has used for early detection of attack	Multiple source of attack could be bypassed
[31]	2016	A Sequential Probability Ratio Test(SPRT) method has used to detect for high rate DDoS attack	SPRT
[32]	2016	Leveraging on SDN's flow monitoring capability and uses ML approaches to detect and mitigate the malicious traffic	SOM ML technique
[33]	2017	Technique has been proposed to mitigate TCP flood attack	Monitor the ongoing TCP connection requests and block malicious hosts.
[34]	2018	For low false positive, and high precision against DDoS attack the extreme gradient boosting classifier has used.	ML techniques.

help to identify attack traffic [41–44]. Recently, Mohammadi et al. propose a detection technique to counter the TCP SYN flooding attacks to SDN. For detection purpose, they have utilized the programmability property of the SDN [33]. However, the controller is vulnerable to other possible protocols also. As contrasting to other attacks, DDoS attack, can cause a massive interruption in any kind of network infrastructure [45–47].

From the above literature survey, we found that very few works have considered low rate DDoS attack for SDN control plane and no work has adopted the information distance metric for the attack traffic detection purpose. The information metrics have some advantages over the existing detection techniques. Primarily, these techniques use a minimum number of features during the detection period. Additionally, these techniques are computationally low and can be used in varying scale, in terms of instances taken per window.

Motivated by this, we have carried out the above work and use information distance metric to detect control plane DDoS attack. As DDoS attack exhaust the computing resources of the controller, which is the key entity of SDN, our detection mechanism, extract the traffic statistics from the flow table and impose the GE metric to make an early alert for the attack. Safeguarding the network security is a rat race process between attackers and victims for many years [48–52].

Table 2 lists all the notations and symbols used in this work.

3. General entropy and information distance

Entropy is also termed as Shannon–Wiener index is an essential concept in information theory [53]. It was introduced to measure the uncertainty of an event associated with a given probability distribution $\mathcal{X}$. A higher value of entropy is expected when the

Table 2
Notations used in the works.

Symbols	Description
$\mathcal{H}$	Hash table
Δ T	Sampling interval
δ_1	Threshold for entropy variation
δ_2	Threshold for information distance variation
S	$\{S_1, S_2, \ldots, S_n\}$ OpenFlow switch set
C	$\{C_1, C_2, \ldots, C_k\}$ Controller set
dest_ip	Destination address of the IP
ID	Information distance
GE	Generalized entropy
α	Order of the general entropy
H	Entropy metric
IF_i	ith incoming flow
P	Total probability
k	Total number of *dest_ip* lies within the window
n	Total number of considered destinations ip
σ	Standard deviation
μ	Mean
E_{shn}	Shannon entropy
$H_\alpha(IF)$	General entropy of a input flow (with α order)

probability distribution event is more random in nature. On the other hand, the least value of entropy is expected when the amount of uncertainty in the event is relatively small. To quantify the randomness of the event, Renyi had introduced entropy metric of order α as a generalization of Shannon entropy.

The formal definition of GE in terms of a discrete variable $\mathcal{X}$, with possible outcomes $x_1, x_2, \ldots, x_n$ i.e. $\sum_{i=1}^{n} x_i = 1, x_i \geq 0$, then Renyi's entropy of order α, can be defined as:

$$H_\alpha(\mathcal{X}) = \frac{1}{1-\alpha} log_2 \left(\sum_{i=1}^{N} p_i(\alpha) \right) \tag{1}$$

where, $\alpha \geq 0$ and $x_i \geq 0$. By varying the α order, different types of entropy values can be obtained. When $\alpha = 0$, it indicates the maximum value of the generated information. It can be derived as: $H_0(\mathcal{X}) = log_2 n$. But, when $\alpha = 1$, the GE can be expressed as:

$$\begin{aligned} H_1(\mathcal{X}) &= \sum_{i=1}^{n} p(x_i) log_2 \frac{1}{p(x_i)} \\ &= -\sum_{i=1}^{n} p(x_i) log_2 p(x_i) \end{aligned} \tag{2}$$

where $p(x_i) = Prob(\mathcal{X} = x_i)$ is the probability of the ith outcome of $\mathcal{X}$.

The Eq. (2) is termed as Shannon Entropy (E_{shn}). Similarly, when $\alpha \to 2$ the GE can be written as:

$$H_2((\mathcal{X})) = -log_2 \sum_{i=1}^{n} p_i^2 \tag{3}$$

The above Eq. (3) is termed as Renyi's quadratic entropy. The relationship between E_{shn} and general entropy discussed in [54]. The GE value is always depends on α value. More specifically, when $\alpha > 1$, GE can increase the margin of different probability distribution compared to E_{shn} [55,56]. Based on this analysis in this work, we consider the legitimate traffic and attack traffic as two different probability distribution and combine the property of GE to detect the attack traffic. The fact is that the high uncertainty events contribute more information in GE than E_{shn} [57]. Hence, according to the requirement we can get different entropy values by adjusting the α order. In this regard, we can obtain much better attack detection results by adjusting the α value of GE.

Usually ID applied to measures the similarity amongst the different events and traditionally it is defined as follows.

Let us consider two different probability distribution events such as, $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$, where

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1, \quad p_i, q_i \in [0-1] \tag{4}$$

The information distance (ID) can be written as follows:

$$D_\alpha(P, Q) = \frac{1}{1-\alpha} log_2 \left(\sum_{i=1}^{n} p_i^\alpha q_i^{1-\alpha} \right) \tag{5}$$

The ID is always non-negative, if $\alpha \geq 0$. When both the probability distribution events are same in nature then the $D_\alpha(P, Q) = 0$. Similar to GE, we can obtain following equation by changing the α order.

$$D_0(P, Q) = -log_2 \left(\sum_{i=1}^{n} q_i \right), \quad when \ \alpha = 1 \tag{6}$$

$$D_1(P, Q) = -\sum_{i=1}^{n} p_i log_2 \left(\frac{p_i}{q_i} \right) \ when \ \alpha = 2 \tag{7}$$

The Eq. (7) is referred to as the Kullback–Leibler divergence (KL distance). The KL divergence is a suitable metric for measuring the information distance because it satisfies the three properties given in Eqs. (8)–(9).

1. Identity Property:

$$D_\alpha = 0, D_1(P, Q) = 0; \quad when \ P = Q \tag{8}$$

2. Triangular inequities::

$$\begin{aligned} D_\alpha(P, Q) &\leq D_\alpha(P, S) + D_\alpha(S, Q) \\ D_1(P, Q) &\leq D_1(P, S) + D_1(S, Q) \end{aligned} \tag{9}$$

3. Symmetry Property:

$$D_\alpha(P, Q) = D_\alpha(Q, P) \ and \ D_1(P, Q) = 0, D_1(Q, P) \tag{10}$$

Since, both GE and GID follow all the above properties, hence it can be used as a detection metric.

There are few other metrics which are used to measure the statistical information distance among two probability distributions such as Jeffrey distance, Sibson distance, Hellinger distance. The statistical formulas of these distance metrics are given in Section 6. For comparison purpose we compute these popular distance measures with our proposed ID metric defined in Definition 2. We formulate our SDN based DDoS attack detection on the above analysis.

To further enhance the proposed scheme we introduce some lemmas and definition in this section.

Definition 1 (*DDoS Attack*)**.** In DDoS scenario, the packet rate of the flow is significantly higher than the normal traffic. Let, the traffic sample(s) collected during ΔT period, then a DDOS attack traffic is a subsample *s.t.* the difference of GE values between the normal and attack traffic at least the minimum allowable δ_2 value and individual H_α of any IF_i must be less than δ_1 for consecutive five windows.

Definition 2 (*Information Distance (ID)*)**.** In this proposed method, the ID indicates, the spacing of the entropy (H_α) between two probability distributions. Here, H_α is based on the *dest_ip* of the flows. In other words ID is defined as: ID $= |H_\alpha(IF_i) - H_\alpha(IF_j)|$. Where $\alpha = \{1, 2, \ldots, m\}$

Lemma 1. *The entropy value is maximum when the relative frequency of each dest_ip$_i$ is equally distributed among all the hosts.*

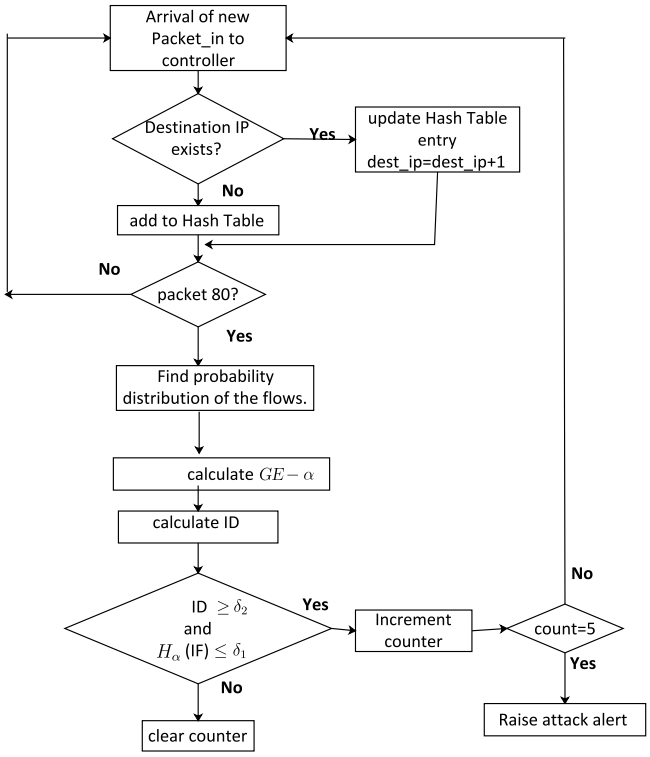


Fig. 3. Flowchart of Proposed DDoS detection.

Proof. Let us consider 50 packets and each packet sending to 50 distinct destination, then probability of each destination address will be $P_i = \frac{1}{k} = \frac{1}{50}$. And the entropy value H_1 as per Eq. (2) will be: $-\sum_1^{50} 0.02 \times log_2 0.02 = 5.644$. Instead of this, out of 50, if 10 packets heading to a single destination then entropy value reduced to 5.214. So, the entropy value is maximum when all incoming packets move to all hosts uniformly. □

Lemma 2. *When $\alpha = 1$, then $H_1(\mathcal{X}) = -\sum_{i=1}^{n} p_i log_2 p_i$*

Proof. At $\alpha = 1$, the value of this quantity is likely undefined as it generates the form 0/0. To find the limit of the entropy value, we apply l'Hospital's theorem i.e.

$$\lim_{\alpha \to 1} \frac{f(\alpha)}{g(\alpha)} = \lim_{\alpha \to 1} \frac{f'(\alpha)}{g'(\alpha)}$$

In this case, $f(\alpha) = log_2 \sum_{i=1}^{n} p_i(\alpha)$ and $g(\alpha) = 1 - \alpha$. So,

$$\begin{aligned}
\lim_{\alpha \to 1} H_\alpha &= \lim_{\alpha \to 1} \frac{1}{1-\alpha} log_2 \sum_{i=1}^{n} p_i(\alpha) \\
&= \lim_{\alpha \to 1} \frac{\frac{d}{d\alpha}(log_2 \sum_{i=1}^{n} p_i(\alpha))}{\frac{d}{d\alpha}(1-\alpha)} \\
&= -\lim_{\alpha \to 1} \frac{\sum_{i=1}^{n} \frac{d}{d\alpha} p_i(\alpha)}{\sum_{i=1}^{n} p_i(\alpha)} \\
&= -\lim_{\alpha \to 1} \frac{\sum_{i=1}^{n} p_i(\alpha)}{\sum_{i=1}^{n} p_i(\alpha) ln 2} \\
&= -\sum_{i=1}^{n} p_i log_2(p_i)
\end{aligned}$$

which is the Shannon entropy. □

Lemma 3. *The GE value is a non-increasing function of α.*

Proof. When $\alpha \geq 0$, the partial derivate of $\frac{\partial H_\alpha(x)}{\partial \alpha} \leq 0$ [10]. It implies that the GE value is a non-increasing function of α. So, $H_{\alpha_1(x)} \geq H_{\alpha_2(x)}$ for $\alpha_1 < \alpha_2$. □

4. Proposed attack detection scheme

Usually, a DDoS attack model can comprise of three-tuples i.e. <Attacker, Victim, Type>. In SDN environment, it can be identified as: $Attacker = \{host, switch, botnet\}$, $victim = \{controller, switch, host, controller - switch\ link\}$, $Type = \{UDPflood, TCP\_Synflood, HTTPflood\}$. Any attack can fall into the permutation of this tuple set [58]. A target of an attack may affect multiple types of victim. For instance, in our case, host, controller, UDP flood, which implies attacker is the host, target is controller, and the type of attack is UDP flooding. As above said, targeting to a host indirectly affect controller instance present in the same domain. With this model we have formulated out attack detection scheme. The proposed detection scheme is depicted in Fig. 3. This scheme incorporate two modules in the controller. One module is meant for statistic collection for the unmatched incoming flows and creates a hash table which has given in Algorithm 1. The second module is meant for anomaly section. The steps of the detection scheme is given in Algorithm 2.

Algorithm 1 Dest_ip_Statistics_Collections

Input: *sampling interval T,r=0*
Output: *Hash table $\mathcal{H}$(dest_ip, occurrence)*

```
for all incoming packet_ins received by controller do
    collecting IP statistics of corresponding packet_in
end for
for ∀dest_ip ∈ window
    if dest_ip ∉ H(dest_ip, r) then do
        H(dest_ip, r) ← 1
    else
        H(dest_ip, r) ← r + 1
    end if
end for
```

Output: Return $\mathcal{H}(dest\_ip, occurrence)$

For detection of DDoS attack using entropy there are two important components need to follow (i) the window size (ii) a threshold value. Again, the window size can be decided by two ways, one is in terms of number of packets received and in terms elapsed time. We have considered the number of received packets as the parameter of window size. The entropy is calculated based on the occurrence of the incoming packets within the window. As indicated in Lemma 1, a sudden decrease in the entropy value in a network is an alarm for the possible attack. For SDN, detection of possible DDoS attack in an early stage is vital. Hence, for every incoming packet_in, the module first unwraps it and get the flow information from the flow table. For the window size 80, the Algorithm 1 extracts the *dest_ip* and stored into the hash table. The hash table contains the *dest_ip* and occurrence of it. This information helps for computing the entropy as well as ID among the flows. To determine an attack, we are setting two threshold value. One for entropy (δ_1) and another for ID (δ_2).

For this we calculate the GE of both the flows (normal and attack) and measure the ID of order α as defined in Eq. (5). If the $ID \geq \delta_2$ and $H_\alpha(IF_i) \leq \delta_1$ continue for an array of consecutive windows then it is confirmed that there is an attack in progress.

Since IP spoofing is a common practice, in a complex scenario it is hard to identify the source IP address. Hence, to mitigate the low rate attack, a mitigation module has deployed in the controller. This module generates the flow rule to block the specific type of

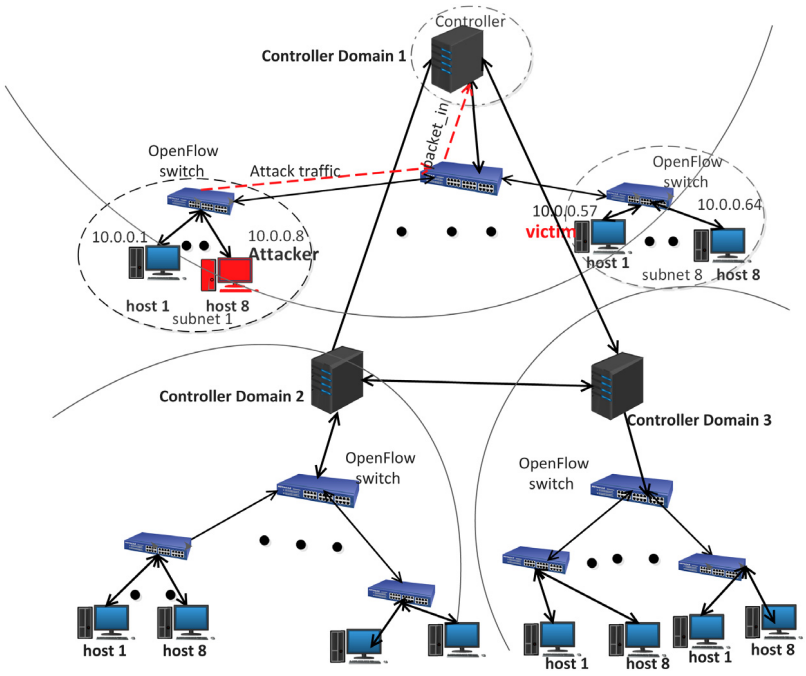


Fig. 4. Considered attack scenario for the work.

Algorithm 2 Low-rate_DDoS Attack_Detection using GE

Input: *Hash table* $\mathcal{H}(dest\_ip, occurrence), \delta_1, \delta_2$
Output: *Alarm for low-rate DDoS attack*

for all $IF \in \mathcal{H}$ **do**
  for identify IF_i and IF_j **do**
    Find the probability distribution of both flows
    calculate $H_\alpha(IF_i)$ and $H_\alpha(IF_j)$ using Equation (1)
    if $ID \geq \delta_2$ and ($H_\alpha(IF_i)$ or $H_\alpha(IF_j)$)$\leq \delta_1$
      for 5 successive windows **then**
      Alarm for low-rate DDoS attack traffic
    else
      Normal traffic
    end if
  end for
end for

Algorithm 3 Low-rate_DDoS Attack_Mitigation

Input: S_i, *dest_ip*, *protocol*
Output: *Insert rule to drop the flows*

Identify the victim's IP from Probability Distribution of the flows.
for $S_i \in C$ **do**
  Generate the flow rules *FR*(*dest_ip*, *protocol*, *drop*)
  Set the flow rule, $S_i \leftarrow C$
end for
Output: Insert rule to drop the flows

flows which heading to a particular destination. As soon as the attack alarm is raised, the victim and protocols are identified. The victim of the attack is nothing but the most frequent destination IP address within the window. As explained in Algorithm 3, a flow rule is generated by the controller to drop the flows destined to the specific victim employing the specific protocol. After getting the flow rule, corresponding switch $S_i \in C$, through which flows destined to the victim, drop the flows and rest will regularly communicate within the network.

5. Experimental setup

We run our experiment on a PC with Intel Core i7-4770 processor, 3.4 GHz clock speed with 4 GB RAM. The operating system is Linux Ubuntu 14.04 LTS and Mininet V 2.2.26 which supports the OF version 1.3. To illustrate the functionality of the proposed approach, we have used the following scenario which has illustrated in Fig. 4. The network consists of three controller domain and each domain consists of one POX controller, 64 hosts and 8 OpenFlow switches. We connect 8 hosts to each OF switch to create a network subnet. As POX is a fast and lightweight controller and can work on all most all available platforms, most of the authors have used this in their experiment. For simulation purpose, we create this network using MiniEdit tool. The kernel name-space feature of Mininet helps to prototype a network scenario in a single PC. The individual process have the own routing table, network interface etc. Mininet takes the help of this features and utilizes the process based virtualization concept to run network elements in the kernel. For network switch we have used Open Virtual Switch(OVS) in which network interfaces and various protocols have already implemented on it. The Scapy tool is considered as a powerful tool to generate real flooding attacks. In our experiments, this tool has been used to generate UDP packets and spoof the source IP address of the packet. The legitimate traffic holds the payload whereas the attack traffic does not contain any payload. We have used the "randrange" function of Python to generate random source IP addresses in between [1-255]. For validation of this work, we only consider a single controller domain i.e. controller domain-1 for carry out the attack scenario. For which the legitimate traffics' destination IP is starting from 10.0.0.1 to 10.0.0.64. The two different python scripts have written for generating traffic: one for attack traffic and other is for normal traffic. For single host attack, we randomly choose one host as attack node and another as a victim, and the rest run the normal traffic. For example 10.0.0.8 is the attacker in Fig. 4.

6. Performance evaluation

In order to validate the information distance metric and carry out the simulation, we make the following assumptions.

Table 3
Threshold value comparison w.r.t. different attack rate.

Parameters	Normal traffic	10% attack traffic	20% attack traffic	30% attack traffic	50% attack traffic	80% attack traffic
Mean	0.8189	0.80705	0.7751	0.72994	0.6274	0.3327
Standard deviation	0.0152	0.0193	0.0256	0.03007	0.0374	0.03711
Confidence-Max	0.8204	0.80988	0.77760348	0.733955	0.63103	0.33627
Confidence-Min	0.8174	0.80421	0.77260259	0.723955	0.623748	0.32908
Confidence-Interval	0.00301	0.00566	0.004957	0.0080157	0.0072816	0.007219
Threshold (δ_1)		0.8058	0.7894	0.75174762	0.6346	0.3439

- The normal traffic follows the Gaussian distribution and the attack traffic follows the Poisson distribution.
- The attack flows coming from OpenFlow devices and merge at controller.
- During the attack, the used attack traffic generating tool generates UDP packets and spoof the source IP address of the packet.
- During an attack, the whole network system is stable.

6.1. Test cases

In order to examine the effectiveness of this metric in detecting low rate DDoS attacks on the controller, in this experiment we are covering two different scenarios. In one case, three different intensity of attack rates are running on a single host and in the second case with different intensity of attack run on a subnet belong to a single switch. For a single victim attack, we randomly choose a host to generate attack traffic and rest are executing the normal traffic. Since, packet_in events generally contain only headers of the packets so, in the simulation we have not added any payload to the generated packets. In the first scenario, we are considering 10%, 20%, and 30% attack rate. For this two separate Scapy programs written in Python are running separately. To control the intensity of attack rate $Attack_{Rate}$, we are using the following Eq. (11).

$$Attack_{Rate} = \frac{P_{attack}}{P_{total}} \times 100\% \tag{11}$$

where, P_{attack}, represents the number of attack packets and P_{total} represent the number total packets generate during the simulation. The attack script runs manually one after another. In this section we will use the term False Positive (FP) and False Negative (FN). The FN represents attack traffic classified as legitimate traffic. Similarly FP, denotes the legitimate flow classified as attack traffic.

6.2. Threshold setting

For setting the threshold value we ran the code for 10 times in each case. After running the simulation 10 times, the False Positive (FP) has significantly reduced, but the False Negative Report has remained within a specific range. Further, we have not found any changes in the false report. So, we stop run the code and set the threshold value. The given entropy value is the highest entropy among the 10 cases. Table 3, represents the threshold value selection in different attack scenario. For setting the threshold value we carried out the following steps.

Step I: At first we calculate the possible minimum value of a normal traffic entropy. This can be achieved by difference between normal traffic mean entropy and confidence interval.

Step II: Then, we calculate the possible maximum value of the attack traffic. It can be achieved by adding the attack traffic mean entropy and confidence interval.

Step III: After that difference between these values decide the δ_1. The mean and standard deviation formulas are given below:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} H_i \tag{12}$$

Table 4
Different parameter setting used in the test scenarios.

Parameter	Type/Value
Packet type	UDP
Window size	80
Destination port id	80
Source port id	2
Normal traffic interval	0.1 (s)
Attack traffic interval	0.025 (s)
Total packet sent	Up to 6500
Payload	None
Topology type	Tree

where, N is the total number of windows and H_i represents entropy of the individual window.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H_i - \mu)^2} \tag{13}$$

To find the optimal threshold value, we run both the codes for ten times. Table 3 represent the optimal threshold value in different attack scenario.

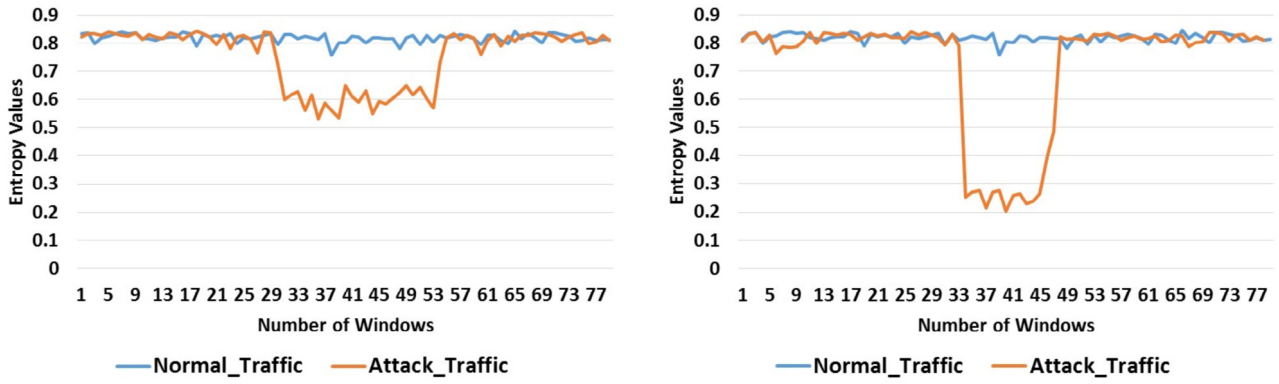
6.3. Results

As the main objective of this work is to improve the detection accuracy during DDoS attack, for which we have compared our work with the related works done by Mousavi et al. [30]. The author has considered E_{shn} as the detection metric.

The Table 4 represents the different parameter setting used in the experiment. In order to evaluate the proposed method, we have conducted various experiments on the above-designed topology. Five different attack rates are tested under two attacking types: single victim and multiple victim attacks. The attack traffic rate has decided by Eq. (11). An entropy value is maximum when all IP addresses are different. For all different IPs, there would be a provision for one slot within the specified window size. Hence, we have considered the window size is 80 because there are 64 hosts in the topology.

Each point on the horizontal line of Figs. 5 and 7 denote a window and the vertical line represent the E_{shn} value of each window. The blue curve and red curve represent the normal traffic and attack traffic respectively. To make the network stabilize we run the normal traffic code for 10 min. This period can be treated as the primary learning stage of the algorithm and then we run the attack code. Then we run the attack code and total 500 number of packets are sent during the attack period.

In order to test a low-rate DDoS attack we have considered three different attack scenarios. These three different attack scenarios are tested under two attacking types such as: single victim and multiple victim attack. In single victim attack one single host being targeted and in multiple victim attack a group of hosts of a subnet being targeted. During simulation to quantify an early detection, we check the first 500 packets of the traffic because it has assumed that for a consecutive five windows if the entropy falls below the threshold, it is an alarm for the attack.



(a) 50% attack traffic (b) 80% attack traffic

Fig. 5. Entropy variation with high rate attack. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

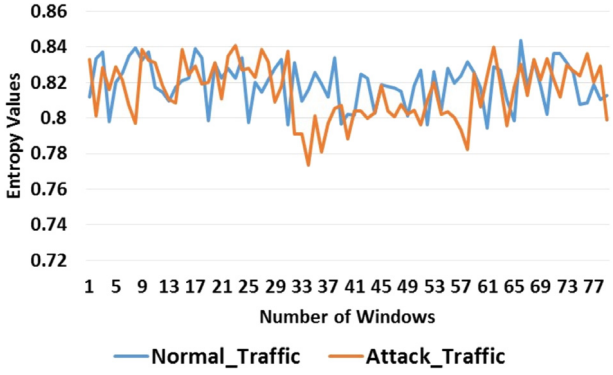


Fig. 6. Entropy variation with 10% attack traffic.

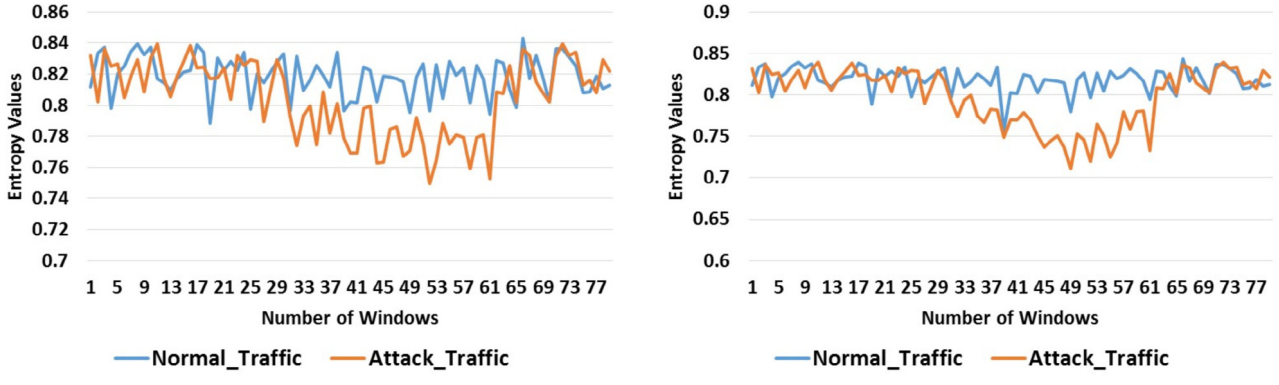
In the initial part of this simulation, we use the E_{shn} to detect the DDoS traffic. Later we are utilizing the proposed metric and compare with E_{shn}.

During simulation if an attack is suspected due to entropy variation it can reflect to the log file. If the entropy variation falls below the δ_1 for 5 consecutive windows an attack report will raised. The rate of attack which are written in the log file could be significant information for the network administrator so that he or she can make suitable flow rule to mitigate the attack in much earlier.

6.3.1. Single victim attack

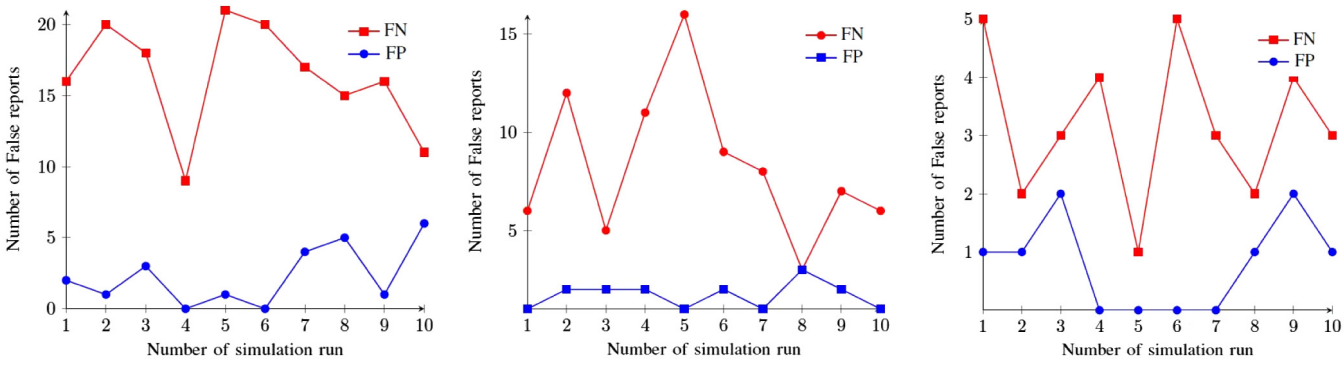
In order to evaluate the performance of E_{shn} metric we have conducted an experiment with 50% and 80% attack rate which has shown in Fig. 5a and b respectively. From the graph it has been observed that, a higher rate DDoS traffic's entropy value drops substantially. But, the drop of entropy value is insignificant in case of 10% and 20% attack rate which are shown in Fig. 6 and Fig. 7a respectively. The Fig. 6 is the result of a single instance among 10 runs. From the figure it can be noticed that at the initial phase of the attack (window number 30 onwards), the entropy value is unstable. At 32nd and 33rd the entropy is 0.802 and 0.791 which are below the threshold value. Then after the entropy value is 0.8113 which is above the threshold. So, the attack detection is a challenging task because the attack traffic is not well below the threshold for a consecutive windows and in low rate attack, choosing the threshold is difficult.

In another experiment we find the FP and FN reports which are shown in Fig. 8a, b, and c. The results are obtained after 10 different runs. For example in 10% attack rate the average number of FN report is 16.3 and it is 3.2% of the first 500 attack traffic. It FN, signifies that attack was in progress but not detected by the controller. In 30% attack rate, the average number of FN report is 3. In each run we record the mean false reports and calculate the percentage value of FN and FP error which is listed in Table 5. The results indicate that false report reaches at peak in the lowest attack rate compared to other attacking rate. As we go on increasing the attack rate, the false report reduces. Here, we conclude that at



(a) 20% attack traffic (b) 30% attack traffic

Fig. 7. Entropy variation with different low rate attack. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) 10% attack traffic (b) 20% attack traffic (c) 30% attack traffic

Fig. 8. False report in different attack rate.

Table 5
FN an FP report using E_{shn} (single victim attack).

False negative	10% attack traffic	20% attack traffic	30% attack traffic
Avg. no of FN report	16.3	8.3	3.2
% of FN report	3.2	1.6	0.66
False positive	10% attack traffic	20% attack traffic	30% attack traffic
Avg. no of FP report	2.2	1.7	0.8
% of FP report	0.44	0.34	0.16

Table 6
ID with different α value for 10% attack traffic.

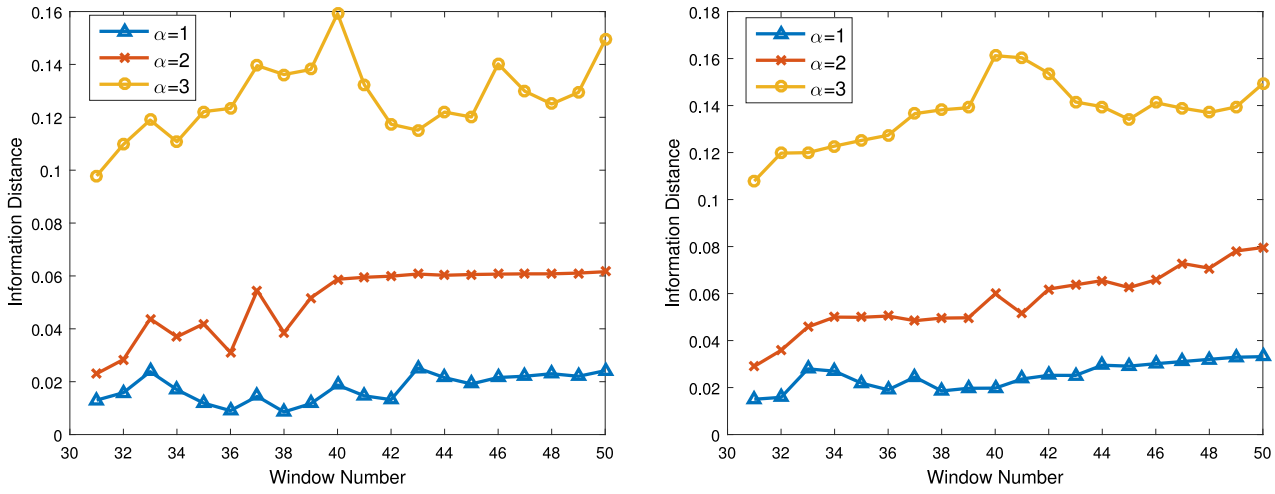
$\alpha = 1$ $\delta_1 = 0.805$ $\delta_2 = 0.03$			α=2 $\delta_1 = 1.214$ $\delta_2 = 0.12$			α=3 $\delta_1 = 1.121$ $\delta_2 = 0.22$		
Normal	Attack	ID	Normal	Attack	ID	Normal	Attack	ID
0.832777902	0.817785254	0.014992648	1.36444161	1.296592701	0.067848909	1.355916081	1.387809731	0.03189365
0.831226078	0.79096505	0.040261028	01.372720808	1.252971405	0.119749402	1.355916081	1.387809731	0.03189365
0.80946049	0.791238737	0.018221753	1.418391634	1.219423485	0.198968149	1.355916081	1.387809731	0.03189365
0.815842503	0.773703081	0.042139421	1.353557062	1.232805782	0.120751279	1.37724733	1.368881059	0.008366271
0.825600837	0.801379555	0.024221282	1.384007607	1.261853911	0.122153696	1.365425404	1.339665862	0.025759542
0.819184878	0.781118528	0.038066351	1.407601154	1.294234849	0.113366304	1.397592712	1.353086791	0.044505921
0.811558763	0.796932078	0.014626685	1.362590332	1.283033398	0.079556935	1.37724733	1.1384334	0.23881393
0.833866484	0.805200927	0.028665557	1.391932744	1.245757491	0.146175253	1.369368756	1.129188	0.240180756
0.796468824	0.807347741	0.010878917	1.362434986	1.29164368	0.070791306	1.36444161	1.124597	0.23984461
0.802319867	0.78809448	0.014225387	1.389851072	1.230622674	0.159228398	1.372720808	1.13526716	0.237453648
0.801770363	0.803801076	0.002030713	1.383755611	1.251441428	0.132314183	1.418391634	1.121204246	0.297187388
0.824721931	0.804079078	0.020642853	1.337393254	1.240020423	0.09737283	1.353557062	1.130807	0.222750062
0.822469131	0.799996679	0.022472452	1.331987028	1.28077203	0.051214998	1.384007607	1.1291531	0.254854507
0.802355093	0.802809126	0.000454033	1.369368756	1.267388912	0.101979844	1.407601154	1.13123404	0.276367114
0.818890356	0.81802066	0.000869696	1.345169139	1.243942322	0.101226817	1.362590332	1.1260237	0.236566632
0.817898179	0.803923557	0.013974622	1.351210786	1.311010985	0.040199802	1.391932744	1.124206	0.267726744
0.81683468	0.800689998	0.016144682	1.382434986	1.250572901	0.131862085	1.362434986	1.127168	0.235266986
0.815014063	0.807744702	0.007269362	1.363071012	1.308034897	0.055036115	1.389851072	1.12302239	0.266828682
0.801098303	0.801997553	0.00089925	1.399157739	1.305079065	0.094078674	1.383755611	1.1250143	0.258741311
0.818078783	0.80456629	0.013512492	1.380578246	1.287809731	0.092768514	1.337393254	1.1242042	0.213189054

Table 7
FN report with increasing α value using ID (single victim).

α		10% attack traffic	20% attack traffic	30% attack traffic
$\alpha = 2$	Avg no. FN report	8	2	1
	% of FN report	0.16	0.40	0.20
$\alpha = 3$	Avg no. FN report	0	0	0
	% of FN report	0.0	0.0	0.0

low rate attack traffic, the detection is challenging and the Shannon entropy is lacking to provide the information of attack happened.

To solve the above limitation of E_{shn}, we have used information distance metric for which the GE has been considered. For comparison purpose, we kept the number of windows same as the previous simulation. The GE values of order α and spacing between the normal and attack traffic during simulation for window number starting from 30 to 50 are listed in Table 6. In this work we use the GE metric upto the $\alpha = 3$. The spacing between the two flows (cells colored with yellow) indicates the effectiveness of the ID measure to detect the attack traffic. For instance, the ID is very small in case of E_{shn} which makes the distinction of attack flow more challenging. When $\alpha = 1$, although the entropy value below the δ_1 i.e. 0.8058, but the ID is not above the δ_2 (0.03) for a consecutive five windows. Hence, it is challenging to correctly identify the DDoS traffic using E_{shn} metric at an early stage. When $\alpha = 2$ the ID is more than the threshold (δ_2) i.e. 0.12 for 5 consecutive windows and the GE of the DDoS flow is below threshold (δ_1) as well. But, few



(a) 20% attack traffic (b) 30% attack traffic

Fig. 9. Spacing between normal and low-rate DDoS traffic with different α value.

Table 8
Threshold setting for δ_1 and δ_2 and corresponding false report (multiple victim attacks).

10% attack	$\alpha_1 = 1$			$\alpha_2 = 2$		
	δ_1	δ_2	FN report	δ_1	δ_2	FN report
4 hosts	1.191	0.20	4	1.211	0.26	0
8 hosts	1.342	0.18	1	1.371	0.22	0

attack traffics are not identified by the controller. At $\alpha = 3$, the ID is greater than 0.22 for consecutive 5 windows which is even more than previous α order. In each case we compute the false error rate. However, it is found that at $\alpha = 3$, FP report has completely avoided and the FN report reduced drastically which is near to zero as reported in Table 7. The Fig. 9a and b provide the experimental results when using the GE metric for detecting 20% and 30% attack rate. We observed that (for window 30 to 50), as the α order increases the spacing between normal and attack traffic also increases substantially. For the suitability, the α order and δ_2 can be modulated to raise an attack alert at an early stage.

6.3.2. Multiple victim attack

For multiple victim attacks, during simulation, we target four hosts belong to the same subnet. For measuring the effectiveness of the Shannon entropy, by keeping the threshold value same, we start the simulation on the hosts at 10% attack rate. As the number of targeted hosts is more, it is difficult to analyze the entropy value within the 80 window size. So, we increase the window size to 100.

During simulation, it has been observed that the number of attack packets that are sent to the controller is varying from 30 to 40 within the window. On an average, each host receives around 8 packets per window in an attack. Although it seems very few, as compared to single host the incoming attack packet is five times faster than the normal traffic. The chosen threshold (δ_1) and ID (δ_2) values for multiple victim attack are listed in Table 8. In 10% attack, for $\alpha = 2$ the ID value is higher for both the cases and we get almost zero FN report.

Another important parameter i.e. average detection time has been measured in each attack scenarios. The Fig. 10 shows that, for $\alpha = 1$ the detection time is higher than other α order. For single victim attack the average delay for 30% attack is 2.1 s ($\alpha = 3$).

To evaluate the GE metrics in a more complex scenario, we increase the number of hosts gradually in control domain-1.

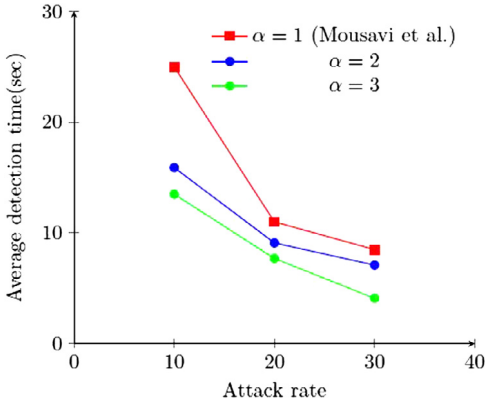


Fig. 10. Detection time compared to different attack rate.

Accordingly, we increase the window size, traffic volume, and duration of the simulation. Next, we consider the highest difference between the ID values between two flows using GE metric. The experimental result as shown in Fig. 11 illustrate that as the attack rate increases the spacing between two flows increases with α value accordingly.

6.4. Performance of distance metric

We have seen in the experiment that it is more FN report than FP report in all the scenarios. For the purpose of this discussion, we assume that the FN rate is due to Shannon metric. Therefore, the FNR (θ) is defined as the proportion of attack events that are mistakenly treated as being legitimate events in the total simulation. The FNR is defined as:

$$\theta = \frac{\theta_{E_{shn}} - \theta_{GE}}{\theta_{E_{shn}}} \tag{14}$$

The θ value specifies how much effectively the generalized entropy metric outperforms the E_{shn} metric. The GE metric reduces the FNR for detecting low-rate DDoS attack which has shown in Table 9.

Next, we investigate the effectiveness of our considered metric with other distance metrics. As we have discussed in Section 3, every information theory metric must hold these three properties to be used as a detection metric, such as symmetric, identity,

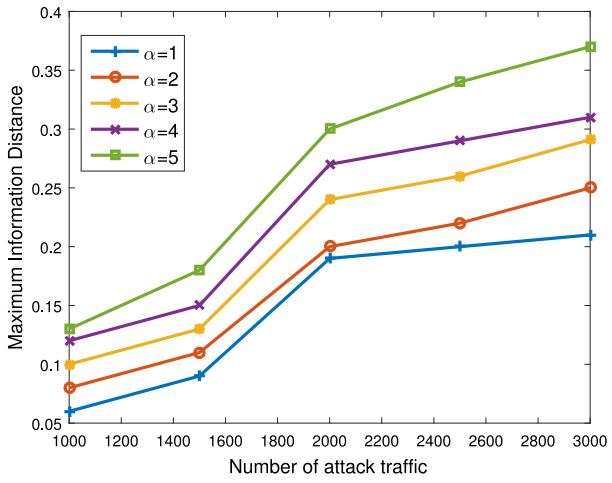


Fig. 11. Variation of information distance in increasing attack traffic intensity.

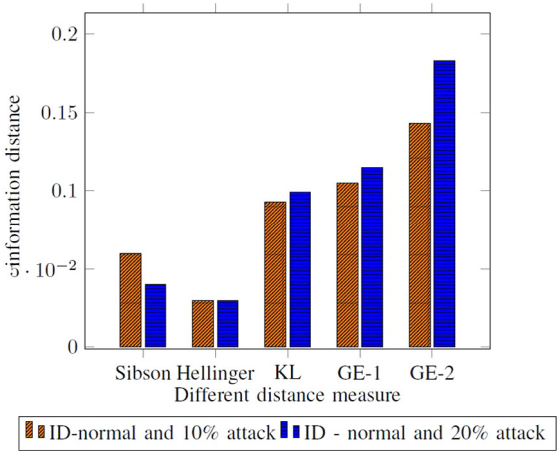


Fig. 12. Comparison of different ID metrics (in single victim scenario).

Table 9
Percentage of reduction in FNR (θ) due to GE metric over E_{shn}.

GE metric	10% attack rate	20% attack rate	30% attack rate
$\alpha = 2$	50.92%	75.9%	68.1%
$\alpha = 3$	100%	100%	100%

and triangular inequalities. In this respect, both GE and GID holds follow all these properties. Therefore, we can compare GE with another distance metric such as Sibson, Hellinger, Kullback–Leibler (KL) divergence. The statistical formulas of these metrics are listed above. The KL information distance metric has already discussed in Section 3. From Fig. 12, it is observed that the GE provides a greater information distance and hence the detection accuracy is better as compared to other metrics.

From the above experiment, we have observed some important facts, which are listed below.

- For detection of low rate DDoS attack to SDN control and data plane, the ID metrics show better result than the normal Shannon entropy utilized by earlier works.
- While we increase the α order of GE, the spacing between attack and normal traffic also increases as compared KL divergence which helps to improve the accuracy.
- It is a powerful statistical method through which we can quickly detect the possible attack threats only after observation of a fewer number of windows' ID values. This method can raise attack alert for the first 250–500 malicious packets reaching to the controller.
- Network administrator can make more accurate decision by modulating the α order of GE.

7. Conclusion

Low rate DDoS attack is a serious threat to the control layer of SDN based data center. It is very much important to identify the attack much before it happens. One of the ways the controller layer can be attacked is due to increasing the number of packet_in control events. When packet_in event increases, it becomes a bottleneck for the controller, and the resources start depleting. In such situation, usual Shanon entropy is a less efficient method to detect the false alarm. Hence, we have employed generalized entropy (GE) as the information distance metric to discriminates low rate DDoS attack and normal traffic. Also, we have compared GE with other ID metrics like KLD, Hellinger, and Sibson distance. We have observed that this metric can able to identify attack traffic from legitimate traffic to a greater extent with an improved false negative rate. In the future work, we can employ this technique for high rate DDoS attack and try to set the threshold more dynamic way in a real traffic scenario such that the detection can be done as early as possible.

References

[1] F. Hu, Q. Hao, K. Bao, A survey on software-defined network and openflow: From concept to implementation, IEEE Commun. Surv. Tutor. 16 (4) (2014) 2181–2206.
[2] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, M. Conti, A survey on the security of stateful sdn data planes, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1701–1725.
[3] I. Alsmadi, D. Xu, Security of software defined networks: A survey, Comput. Secur. 53 (2015) 79–108.
[4] K.S. Sahoo, D. Puthal, M.S. Obaidat, A. Sarkar, S.K. Mishra, B. Sahoo, On the placement of controllers in software-defined-wan using meta-heuristic approach, J. Syst. Softw. (2018).
[5] H. Farhady, H. Lee, A. Nakao, Software-defined networking: A survey, Comput. Netw. 81 (2015) 79–95.
[6] A. Akhunzada, E. Ahmed, A. Gani, M.K. Khan, M. Imran, S. Guizani, Securing software defined networks: taxonomy, requirements, and open issues, IEEE Commun. Mag. 53 (4) (2015) 36–44.
[7] S. Denazis, E. Haleplidis, J.H. Salim, O. Koufopavlou, D. Meyer, K. Pentikousis, Software-Defined Networking (SDN): Layers and Architecture Terminology, RFC–7426, 2015.
[8] R. Wang, Z. Jia, L. Ju, An entropy-based distributed DDoS detection mechanism in software-defined networking, in: Trustcom/BigDataSE/ISPA, 2015 IEEE, Vol. 1, IEEE, 2015, pp. 310–317.
[9] D. Puthal, M.S. Obaidat, P. Nanda, M. Prasad, S.P. Mohanty, A.Y. Zomaya, Secure and sustainable load balancing of edge data centers in fog computing, IEEE Commun. Mag. 56 (5) (2018) 60–65.
[10] Y. Xiang, K. Li, W. Zhou, Low-rate ddos attacks detection and traceback by using new information metrics, IEEE Trans. Inf. Forensics Secur. 6 (2) (2011) 426–437.
[11] L. Zhou, M. Liao, C. Yuan, H. Zhang, Low-rate ddos attack detection using expectation of packet size, Secur. Commun. Netw. 2017 (2017).
[12] R.R. Kompella, S. Singh, G. Varghese, On scalable attack detection in the network, in: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, ACM, 2004, pp. 187–200.
[13] B.R. Swain, B. Sahoo, Mitigating ddos attack and saving computational time using a probabilistic approach and hcf method, in: Advance Computing Conference, 2009. IACC 2009. IEEE International, IEEE, 2009, pp. 1170–1172.
[14] H. Wang, C. Jin, K.G. Shin, Defense against spoofed ip traffic using hop-count filtering, IEEE/ACM Trans. Netw. (ToN) 15 (1) (2007) 40–53.
[15] Y. Chen, K. Hwang, Collaborative detection and filtering of shrew ddos attacks using spectral analysis, J. Parallel Distrib. Comput. 66 (9) (2006) 1137–1151.
[16] T. Vissers, T.S. Somasundaram, L. Pieters, K. Govindarajan, P. Hellinckx, Ddos defense system for web services in a cloud environment, Future Gener. Comput. Syst. 37 (2014) 37–45.
[17] G. Somani, M.S. Gaur, D. Sanghi, M. Conti, Ddos attacks in cloud computing: collateral damage to non-targets, Comput. Netw. 109 (2016) 157–171.

[18] S. Alanazi, J. Al-Muhtadi, A. Derhab, K. Saleem, A.N. AlRomi, H.S. Alholaibah, J.J. Rodrigues, On resilience of wireless mesh routing protocol against dos attacks in iot-based ambient assisted living applications, in: E-Health Networking, Application & Services (HealthCom), 2015 17th International Conference on, IEEE, 2015, pp. 205–210.
[19] S. Shin, V. Yegneswaran, P. Porras, G. Gu, Avant-guard: Scalable and vigilant switch flow management in software-defined networks, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, 2013, pp. 413–424.
[20] S. Shin, P.A. Porras, V. Yegneswaran, M.W. Fong, G. Gu, M. Tyson, FRESCO: Modular composable security services for software-defined networks, in: NDSS, 2013.
[21] J. Noh, S. Lee, J. Park, S. Shin, B.B. Kang, Vulnerabilities of network os and mitigation with state-based permission system, Secur. Commun. Netw. 9 (13) (2016) 1971–1982.
[22] D. Kreutz, F. Ramos, P. Verissimo, Towards secure and dependable software-defined networks, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ACM, 2013, pp. 55–60.
[23] Q. Yan, F.R. Yu, Q. Gong, J. Li, Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 602–622.
[24] R. Braga, E. Mota, A. Passito, Lightweight ddos flooding attack detection using nox/openflow, in: Local Computer Networks (LCN), 2010 IEEE 35th Conference on, IEEE, 2010, pp. 408–415.
[25] S.A. Mehdi, J. Khalid, S.A. Khayam, Revisiting traffic anomaly detection using software defined networking, in: International Workshop on Recent Advances in Intrusion Detection, Springer, 2011, pp. 161–180.
[26] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, , Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Comput. Netw. 62 (2014) 122–136.
[27] S. Dotcenko, A. Vladyko, I. Letenko, A fuzzy logic-based information security management for software-defined networks, in: Advanced Communication Technology (ICACT), 2014 16th International Conference on, IEEE, 2014, pp. 167–171.
[28] R. Kokila, S.T. Selvi, K. Govindarajan, Ddos detection and analysis in sdn-based environment using support vector machine classifier, in: Advanced Computing (ICoAC), 2014 Sixth International Conference on, IEEE, 2014, pp. 205–210.
[29] S. Lim, S. Yang, Y. Kim, S. Yang, H. Kim, Controller scheduling for continued sdn operation under DDoS attacks, Electron. Lett. 51 (16) (2015) 1259–1261.
[30] S.M. Mousavi, M. St-Hilaire, Early detection of ddos attacks against sdn controllers, in: Computing, Networking and Communications (ICNC), 2015 International Conference on, IEEE, 2015, pp. 77–81.
[31] P. Dong, X. Du, H. Zhang, T. Xu, A detection method for a novel ddos attack against sdn controllers by vast new low-traffic flows, in: Communications (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.
[32] Y. Xu, Y. Liu, Ddos attack detection under sdn context, in: INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications, IEEE, IEEE, 2016, pp. 1–9.
[33] R. Mohammadi, R. Javidan, M. Conti, Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks, IEEE Trans. Netw. Serv. Manag. (2017).
[34] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, J. Peng, Xgboost classifier for ddos attack detection and analysis in sdn-based cloud, in: Big Data and Smart Computing (BigComp), 2018 IEEE International Conference on, IEEE, 2018, pp. 251–256.
[35] R. Kandoi, M. Antikainen, Denial-of-service attacks in openflow sdn networks, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, IEEE, 2015, pp. 1322–1326.
[36] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, B. Yang, Predicting network attack patterns in sdn using machine learning approach, in: Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on, IEEE, 2016, pp. 167–172.
[37] M. Yu, L. Jose, R. Miao, Software defined traffic measurement with opensketch, in: NSDI, Vol. 13, 2013, pp. 29–42.
[38] K.S. Sahoo, M. Tiwary, B. Sahoo, Detection of high rate ddos attack from flash events using information metrics in software defined networks, in: Communication Systems & Networks (COMSNETS), 2018 10th International Conference on, IEEE, 2018, pp. 421–424.
[39] J. David, C. Thomas, DDoS attack detection using fast entropy approach on flow-based network traffic, Procedia Comput. Sci. 50 (2015) 30–36.
[40] D. Kotani, Y. Okabe, A packet-in message filtering mechanism for protection of control plane in openflow switches, IEICE Trans. Inf. Syst. 99 (3) (2016) 695–707.
[41] V. Prokhorenko, K.-K.R. Choo, H. Ashman, Web application protection techniques: A taxonomy, J. Netw. Comput. Appl. 60 (2016) 95–112.
[42] V. Prokhorenko, K.-K.R. Choo, H. Ashman, Context-oriented web application protection model, Appl. Math. Comput. 285 (2016) 59–78.
[43] J. Peng, K.-K.R. Choo, H. Ashman, User profiling in intrusion detection: A review, J. Netw. Comput. Appl. 72 (2016) 14–27.
[44] A. Nanda, P. Nanda, X. He, A. Jamdagni, D. Puthal, A hybrid encryption technique for secure-glor: The adaptive secure routing protocol for dynamic wireless mesh networks, Future Gener. Comput. Syst. (2018).
[45] O. Osanaiye, H. Cai, K.-K.R. Choo, A. Dehghantanha, Z. Xu, M. Dlodlo, Ensemble-based multi-filter feature selection method for ddos detection in cloud computing, EURASIP J. Wirel. Comm. Netw. 2016 (1) (2016) 130.
[46] O. Osanaiye, K.-K.R. Choo, M. Dlodlo, Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework, J. Netw. Comput. Appl. 67 (2016) 147–165.
[47] A.K. Mishra, A.K. Tripathy, D. Puthal, L.T. Yang, Analytical model for sybil attack phases in internet of things, IEEE Internet Things J. (2018).
[48] C. Lin, D. He, N. Kumar, K.-K.R. Choo, A. Vinel, X. Huang, Security and privacy for the internet of drones: Challenges and solutions, IEEE Commun. Mag. 56 (1) (2018) 64–69.
[49] C.J. DOrazio, K.-K.R. Choo, Circumventing ios security mechanisms for apt forensic investigations: A security taxonomy for cloud apps, Future Gener. Comput. Syst. 79 (2018) 247–261.
[50] Q. Do, B. Martini, K.-K.R. Choo, Cyber-physical systems information gathering: A smart home case study, Comput. Netw. 138 (2018) 1–12.
[51] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, G. Das, Everything you wanted to know about the blockchain: Its promise, components, processes, and problems, IEEE Consum Electron. Mag. 7 (4) (2018) 6–14.
[52] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, C. Yang, The blockchain as a decentralized security framework, IEEE Consum. Electron. Mag. 7 (2) (2018) 18–21.
[53] S. Oshima, T. Nakashima, T. Sueyoshi, Early dos/ddos detection method using short-term statistics, in: Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, IEEE, 2010, pp. 168–173.
[54] K. Życzkowski, Rényi extrapolation of shannon entropy, Open Syst. Inf. Dyn. 10 (03) (2003) 297–310.
[55] K. Kumar, R. Joshi, K. Singh, A distributed approach using entropy to detect ddos attacks in isp domain, in: Signal Processing, Communications and Networking, 2007. ICSCN'07. International Conference on, IEEE, 2007, pp. 331–337.
[56] S. Behal, K. Kumar, Detection of ddos attacks and flash events using novel information theory metrics, Comput. Netw. 116 (2017) 96–110.
[57] A.R. Barron, L. Gyorfi, E.C. van der Meulen, Distribution estimation consistent in total variation and in two types of information divergence, IEEE Trans. Inform. Theory 38 (5) (1992) 1437–1454.
[58] N. Dayal, S. Srivastava, Analyzing behavior of ddos attacks to identify ddos detection features in SDN, in: Communication Systems and Networks (COMSNETS), 2017 9th International Conference on, IEEE, 2017, pp. 274–281.

Kshira Sagar Sahoo is a Ph.D. scholar in Computer Science and Engineering at National Institute of Technology, Rourkela, India. He received his M.Tech degree in Information and Communication Technology from IIT, Kharagpur, India in 2014. His research interests include Software Defined Network, network virtualization, Big Data Analytics. He is a student member of IEEE computer society.

Deepak Puthal received the Ph.D. degree in computer from University of Technology Sydney (UTS), Australia. He is currently a Lecturer (Assistant Professor) with the Faculty of Engineering and IT, University of Technology Sydney, Australia. He has authored in several international conferences and journals, including IEEE and ACM transactions. His research interests include cyber security, Internet of Things, distributed computing, and big data analytics. He is an Associate Editor of the IEEE Consumer Electronics Magazine and the KSII Transactions on Internet and Information Systems.

Mayank Tiwary is working as a core developer in SAP Cloud Platform at SAP Labs Bangalore, India. He has received his graduation degree from Biju Patnaik University of Technology, Odisha, India. He has numbers of publications in the domain of distributed and cloud computing.

Joel J.P.C. Rodrigues is a professor and senior researcher at the National Institute of Telecommunications (Inatel), Brazil and senior researcher at the Instituto de Telecomunicações, Portugal. He has been professor at the University of Beira Interior (UBI), Portugal and visiting professor at the University of Fortaleza (UNIFOR), Brazil. He received the Academic Title of Aggregated Professor in informatics engineering from UBI, the Habilitation in computer science and engineering from the University of Haute Alsace, France, a Ph.D. degree in informatics engineering and an M.Sc. degree from the UBI, and a five-year B.Sc. degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. His main research interests include e-health, sensor networks and IoT, vehicular communications, and mobile and ubiquitous computing. Prof. Joel is the leader of NetGNA Research Group (http://netgna.it.ubi.pt), the President of the scientific council at ParkUrbis—Covilhã Science and Technology Park, the Past-Chair of the IEEE ComSoc Technical Committee on eHealth, the Past-chair of the IEEE ComSoc Technical Committee on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community and Publications co-Chair, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Advances on Communications and Networking Technology, the editor-in-chief of the Journal of Multimedia Information Systems, and editorial board member of several high-reputed journals. He has been general chair and TPC Chair of many international conferences, including IEEE ICC, GLOBECOM, and HEALTHCOM. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 500 papers in refereed international journals and conferences, 3 books, and 2 patents. He had been awarded several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, an IARIA fellow, and a senior member ACM and IEEE.

Bibhudatta Sahoo obtained his M. Tech. and Ph.D. degree in Computer Science & Engineering from NIT, Rourkela. He has 24years of Teaching Experience in undergraduate and graduate level in the field of computer Science & Engineering. He is presently Assistant Professor in the Department of Computer Science & Engineering, NIT Rourkela, INDIA. His technical interests include Data Structures & Algorithm Design, Parallel & Distributed Systems, Networks, Computational Machines, Algorithms for VLSI Design, Performance evaluation methods and modeling techniques Distributed computing system, Networking algorithms, and Web engineering. He is a member of IEEE & ACM.

Ratnakar Dash received his Ph.D. degree from National Institute of Technology, Rourkela, India, in 2013. He is presently working as Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. His area of interests includes signal processing, image processing, intrusion detection system and steganography. He has published more than 10 papers in journals of international repute.