Scalable Computing:
Practice and Experience

Scientific International Journal
for Parallel and Distributed Computing

ISSN: 1895-1767

Universitatea de Vest din Timişoara

Volume 20(1) March 2019

Editor-in-Chief

Dana Petcu
Computer Science Department
West University of Timisoara
and Institute e-Austria Timisoara
B-dul Vasile Parvan 4, 300223 Timisoara, Romania
Dana.Petcu@e-uvt.ro

Managinig and TEXnical Editor

Silviu Panica
Computer Science Department
West University of Timisoara
and Institute e-Austria Timisoara
B-dul Vasile Parvan 4, 300223 Timisoara, Romania
Silviu.Panica@e-uvt.ro

Book Review Editor

Shahram Rahimi
Department of Computer Science
Southern Illinois University
Mailcode 4511, Carbondale
Illinois 62901-4511
rahimi@cs.siu.edu

Software Review Editor

Hong Shen
School of Computer Science
The University of Adelaide
Adelaide, SA 5005
Australia
hong@cs.adelaide.edu.au

Domenico Talia
DEIS
University of Calabria
Via P. Bucci 41c
87036 Rende, Italy
talia@deis.unical.it

Editorial Board

Peter Arbenz, Swiss Federal Institute of Technology, Zürich, arbenz@inf.ethz.ch
Dorothy Bollman, University of Puerto Rico, bollman@cs.uprm.edu
Luigi Brugnano, Università di Firenze, brugnano@math.unifi.it
Giacomo Cabri, University of Modena and Reggio Emilia, giacomo.cabri@unimore.it
Bogdan Czejdo, Fayetteville State University, bczejdo@uncfsu.edu
Frederic Desprez, LIP ENS Lyon, frederic.desprez@inria.fr
Yakov Fet, Novosibirsk Computing Center, fet@ssd.sscc.ru
Giancarlo Fortino, University of Calabria, g.fortino@unical.it
Andrzej Goscinski, Deakin University, ang@deakin.edu.au
Frederic Loulergue, Northern Arizona University, Frederic.Loulergue@nau.edu
Thomas Ludwig, German Climate Computing Center and University of Hamburg, t.ludwig@computer.org
Svetozar Margenov, Institute for Parallel Processing and Bulgarian Academy of Science, margenov@parallel.bas.bg
Viorel Negru, West University of Timisoara, Viorel.Negru@e-uvt.ro
Moussa Ouedraogo, CRP Henri Tudor Luxembourg, moussa.ouedraogo@tudor.lu
Marcin Paprzycki, Systems Research Institute of the Polish Academy of Sciences, marcin.paprzycki@ibspan.waw.pl
Roman Trobec, Jozef Stefan Institute, roman.trobec@ijs.si
Marian Vajtersic, University of Salzburg, marian@cosy.sbg.ac.at
Lonnie R. Welch, Ohio University, welch@ohio.edu
Janusz Zalewski, Florida Gulf Coast University, zalewski@fgcu.edu

SUBSCRIPTION INFORMATION: please visit http://www.scpe.org

Scalable Computing: Practice and Experience

Volume 20, Number 1, March 2019

TABLE OF CONTENTS

Special Issue on Opportunistic Network and its Security Challenges:

© SCPE, Timişoara 2019

DOI 10.12694/scpe.v20i1.1519
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. iii–iv. http://www.scpe.org © 2019 SCPE

INTRODUCTION TO THE SPECIAL ISSUE ON OPPORTUNISTIC NETWORK AND ITS SECURITY CHALLENGES

It is our great privilege to present before you Volume 20, Issue 1 of the Scalable Computing: Practice and Experience. We had received 41 paper submissions from Belgium, Malaysia, Indonesia, Bangladesh, Yemen and India, and selected 12 papers for publication. The acceptance rate of this issue is 29 percent. The aim of this special issue is to give the solutions of opportunistic networks and security challenges as well as to collect other research problems in opportunistic networks for further research. This special issue gives the new dimensions for opportunistic networks in the perspective of research.

Ritu Nigam et al. had tried to investigate the contact information and social pattern of the node and propose a message forwarding technique in the social opportunistic network. To obtain the contact information and social pattern of nodes, a bonding metric is constructed to show the direct and indirect bonding of neighboring nodes in the system. As the detachment period contains both the encountered frequency and duration of their connections in the vicinity of its neighbors, it is precise to use it to illustrate the direct bonding in the neighboring relationship. It also incorporates the indirect bonding of nodes by identifying weakest direct bonded nodes to replace it with active indirect bonded nodes of the network. The results depict that the proposed protocol can significantly raise the forwarding efficiency concerning the number of messages delivered, overhead ratio, message dropping, and average latency.

Md. Sharif Hossen et al. presented the paper on Analysis of delay tolerant networks routing protocols using the impact of mobility models. In an intermittently connected mobile network, the communication among the mobile nodes can be established easily using the store-and-forward strategy. This network is also called delay-tolerant which can allow the long latency, irregular data rates, and disruption in mobility. Under such a network scenario, the authors had analysed the impact of mobility models using the performance of several delay-tolerant routing protocols. Hence, they simulate the scenario using the opportunistic network environment simulator. Considering three performance metrics namely, deliver probability, latency, and overhead ratio it was seen that spray-and-focus showed good performance while epidemic gave poor results among the routing techniques considered. Furthermore, shortest path map based movement mobility model deserves good performance compared to random walk and random direction.

Deepak Kumar Sharma et al. presented the paper on Establishing Reliability for Efficient Routing in Opportunistic Networks. The Reliability in Oppnet (RIO) protocol is a reliable protocol that improves the routing in Oppnet and works in combination with the existing routing protocols. RIO makes the source node aware about the status of the message so that if an error occurred in routing then the source node can take suitable action like resending of the message and error reporting. In this work, solutions are provided for five errors namely redirection error, buffer overflow error, parameter problem, Time Limit Exceeded i.e. TTL expiration error and destination unreachable. Through simulation using ONE simulator it has been found that RIO enhances the Spray and Wait routing protocol while working in parallel with it.

Pawan Singh Mehra et al. propound E-CAFL which is an enhancement over CAFL protocol. It takes remnant energy, node density and distance from the sink as input to Fuzzy Inference System for calculating the rank of each sensor node for cluster head candidature. Also, member nodes intelligently select their cluster head on the basis of Cluster Head-Chance which takes into account the rank of the cluster head and distance to that cluster head nodes during cluster formation. Simulation experiments have been performed for the designed protocol. It has been observed from the results that E-CAFL has better stability period and protracted lifetime as compared to LEACH and CAFL protocol.

Rajan Sharma et al. proposed Zone-based Energy Efficient Routing Protocols for Wireless Sensor Networks, is a zone-based framework that focused on minimizing the energy consumption in the re-selection process of zone-head (ZH). In this novel ZH re-selection process, the number of control messages exchanged during the selection process of ZH significantly reduces and it helps to achieve a prolonged lifetime. In addition to that the stability version of the above-said algorithm is proposed, that improves the stability period of the network by selecting ZH on the basis of residual energy.

Arvinda Kushwaha et al. presented an overview of the integration of the wireless sensor network and cloud computing. Particle swarm optimization (PSO) is used to optimize resources. The optimization is done through the load scheduling algorithm. This paper proposed load scheduling algorithm that is based on the particle

swarm optimization technique which is used to optimize total transfer time and cost. Simulation result shows that the proposed method is better than the conventional method.

Musaeed Abouaroek et al. proposed the NTRU algorithm for node authentication in opportunistic networks. NTRU algorithm comes in the category of post-quantum cryptography. It is unbreakable and fast than the RSA algorithm and Elliptic Curve Cryptography.

Suman Bala et al. presented the paper on the security of authenticated group key agreement protocols. The authors analyzed the two AGKA protocols against attacks and found both protocols are insecure. In addition, they fixed the vulnerabilities of Tans protocol.

Pankhuri Sai et al. proposed a user authentication scheme which is one of the most important components of a secure system. Even after the development of advanced authentication mechanisms such as biometrics, the traditional concept of passwords still continues to be the most widely adopted means for user authentication. Owing to the limitations of text-based passwords such as smaller password space, susceptibility to brute force attacks; and that of graphical passwords like shoulder surfing attacks, this paper proposed a novel pattern-based multi-factor authentication scheme that involves the use of a combination of textual and graphical passwords. The proposed system has a larger password space and is secure against shoulder surfing and dictionary attacks since it involves additional mouse input along with the keyboard input. Moreover, a brute force attack is also infeasible for it.

Asif Iqbal Hajamydeen et al. presented about the Intrusion detection systems (IDSs) in the paper. The growing number of the Internet users has paved the way in improving the Internet availability infrastructure to make the facilities accessible through various devices but failed to address the arising security issues. The ubiquitous nature of todays Internet agrees with devices or applications that have adapted the technology to link with the Internet which has brought the challenge in providing network security. In the current scenario with interactions between untrusted clients and networks, the network infrastructures are very defenceless. To defend such situations efficiently, intrusion detection systems (IDSs) were used to deliver a self-defensive power for a system or a network. Therefore, this paper provides an overview and review of data mining-based intrusion detection approaches and also those utilizing heterogeneous logs for intrusion detection. Conclusively, it proposes the characteristics to be contained in future intrusion detection model/framework, ways by which the classification/clustering algorithms to be utilized in the model and the considerations on choosing the data to be tested, in order to detect intrusions effectively.

Chanchal Kumar et al. presented the need for designing a security system for a network system arises due to the greater complex structure. An extended protocol for hiding pertinent information based on a fast Diffie-Hellman using Kummer Surface is described in the paper. The extended protocol is devised by the inclusion of an additional point in the Kummer surface for higher security need. Further, the use of a machine learning method is illustrated, which is employed for the selection of a specific surface from a set of available Kummer surfaces. The use of NSGA-2 algorithm is next described for selection of a specific surface. The newer version of key expansion of the AES-128 algorithm is described and illustrated in the paper. This version is based on a newly devised content-matrix. The scheme adopted for the construction of content-matrix is fully illustrated and an optimization algorithm used in the scheme is given along with the outputs obtained with sample data. The outputs of the new version of key expansion are given. The LIM index that is commonly used for cryptanalysis purpose is described next. Hence, the cryptosystem based on extended Kummer surface and new scheme adopted for key expansion of AES-128 could provide useful techniques for hiding the information in a network system. The use of machine learning and NSGA-2 algorithm could enhance automatic selection for a specified extended Kummer surface. These tools can be quite useful for a cryptosystem designer.

Hamza Mutaher et al. proposed the Zero-Knowledge Proof Based Identification Scheme for Securing Software Defined Network. Due to the leak of security in SDN network, many types of attack like host impersonation attack, Man-in-the-middle attack and Denial of service attack can penetrate into the controller to control the whole network or to shut the network totally down. In this paper, a Zero-knowledge proof based identification scheme has been proposed to secure the SDN controller while the data and control planes establish communication. In this security scheme, the user does not need to send his password to the controller in every login attempt. Instead, the authentication server will share the same secret between both user and controller; the user has to prove to the controller that he knows the secret without revealing the exact secret. Communication

and competition costs along with storage overhead analysis are discussed as well in order to validate this work.

We would like to express our sincere thanks go to the chief editor, editorial board members and reviewers who have reviewed the manuscripts within the time to publish this special issue on the scheduled date.

Rosilah Hassan, Universiti Kebangsaan Malaysia, Malaysia
Khairol Amali Bin Ahmad, National Defence University of Malaysia, Malaysia
Khaleel Ahmad, Maulana Azad National Urdu University, India

DOI 10.12694/scpe.v20i1.1469
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 1–16. http://www.scpe.org © 2019 SCPE

BONDING BASED TECHNIQUE FOR MESSAGE FORWARDING IN SOCIAL OPPORTUNISTIC NETWORK

RITU NIGAM[*], DEEPAK KUMAR SHARMA[†], SATBIR JAIN[‡], SARTHAK GUPTA[§], AND SHILPA GHOSH[¶]

Abstract. Integrating social networks properties such as centrality, tie strength, etc. into message forwarding protocols in opportunistic networks has grown into a vital major benchmark. The opportunistic network is a demanding network with no set route to travel a message from the source to the destination. During these networks, nodes use possibilities gained based on store-carry-forward patterns to forward communications. Every node that obtains a message when it encounters another node makes selection concerning the forwarding or not necessarily transmitting the message came across. Most of these message forwarding protocols use the benefit of social properties information like contact information and social relationship enclosed by the nodes in the Social Opportunistic Network. In this paper, a Bonding based forwarding technique proposes which is finding direct and indirect bonding among nodes by exploiting contact information and social pattern. In the proposed protocol, we also focus on indirect bonding by finding weakest direct bonded nodes and then replace it with strong indirect bonded nodes of the network. In this work, the balance between transmission delay and network traffic consider by using shortest path map based mobility model. ONE simulator use for simulation and performance of the proposed protocol compare contrary popular approaches for instance Epidemic, PRoPHET, BubbleRap, and Interaction based when using the shortest path map based mobility model. The Bonding based forwarding technique performs adequately well concerning the number of messages delivered, overhead ratio, message dropping and average latency.

Key words: Direct and Indirect Bonding based forwarding, Social opportunistic network, ONE Simulator.

AMS subject classifications. 68M12

1. Introduction.

In this digital era, many wireless devices like mobile devices, laptops are available which are using wireless technology for communication. It includes ad-hoc scenario; no infrastructure is present, no topology is used for communication between mobile devices. The situation as mentioned above leads to a random network called opportunistic network where a message is traveled from source to destination by using store-carry-forward pattern. In the opportunistic network environment; no fixed route is available, as the nodes come into the contact of other neighbor nodes, it establishes a random path dynamically to forward a message. Many routing techniques are present for communication in the opportunistic network. Different approaches to map the word in the opportunistic environment are available like diffusion based and context based. Routing protocols which follow diffusion technique, circulate multiple replicas of the message to one-hop neighbor nodes. By distributing, numerous reproductions of the text to nodes increment the possibility of reaching the word to the target node in a speedy manner but raises network traffic. Routing based on context uses information that is the frequency of the node to meet other nodes, distance, speed, location history, etc. to minimize the network traffic but increases transmission delay. As of now, universal knowledge of network topology is the base for very efficient routing and forwarding decisions. Aggregating and swapping the information of network topology in the opportunistic network is inconvenient because of their infrequent connectivity and uncertain mobility. Routing strategies for this type of systems generally build on limited knowledge and on forecasting of future contacts which produces deteriorated routing performance. As the nodes in the opportunistic network encounter periodically, there is a requirement of estimating the quality of links between all combinations of nodes to find out the feasible forwarding option between nodes. Later, a distinct weight assigns to each connection of the nodes by condensing the appropriate encounter information. Then, a neighboring graph of nodes forms for forwarding the messages.

[*]Division of Computer Engineering, University of Delhi, Netaji Subhas Institute of Technology Delhi, India (ritu.nigam2106@gmail.com).

[†]Division of Information Technology, University of Delhi, Netaji Subhas Institute of Technology Delhi, India (dk.sharma1982@yahoo.com), corresponding author.

[‡]Division of Computer Engineering, University of Delhi, Netaji Subhas Institute of Technology Delhi, India (jain_satbir@yahoo.com).

[§]Division of Information Technology, University of Delhi, Netaji Subhas Institute of Technology Delhi, India (sarthakgupta259@gmail.com).

[¶]Division of Information Technology, University of Delhi, Netaji Subhas Institute of Technology Delhi, India (ghoshshilpa01@gmail.com).

With the transformation of Online Social Network applications and platforms like Facebook, Twitter, or Linkedin, information about the social intercommunication of nodes has to turn into readily available [1]. Moreover, although the intercontact information in the opportunistic network is varying dynamically, the nodes and the links of a social network stay comparatively stable. Earlier, several approaches, including interaction based [2], communities, similarity, centrality, encounter frequency, total or average contact period and average separation period [3] were adopted to quantify the quality of connections between couples of nodes. Despite, all the approaches as mentioned above have a few shortcomings in the exact depiction of the sending option between nodes.

The leading carrier of mobile devices are human beings, and humans are social by character. These social characteristics play an essential role in forwarding mechanism of the opportunistic social network. Community and centrality of nodes are social characteristics of nodes in the opportunistic social network. With the help of these social characteristics mobility pattern, primary contact nodes, etc. can be identified to predict the path towards the destination. Therefore, it seems that cultural properties are the performance booster elements in the system.

This paper exploits the social network attributes of the opportunistic social network and presents a Bonding based forwarding technique (BBFT) for information propagation. Social relations between nodes evaluate by defining bonding concerning their behavior. The bonding property of the node calculates as a bonding metric determines their concerning behavior. Later, a distinct weight assigns to each connection of the nodes by condensing the appropriate encounter information. Then, a neighboring graph of nodes forms for forwarding the messages behavior. A new bonding metric is specified to measure different forms of bonding behavior regarding contact information and social pattern two types of bonding defined in this paper, i.e., direct bonding and indirect bonding.

The highlights of this work outline as follows:

1. This work inspects the social meeting patterns from the temporal prospects. The explanations of average detachment time and variance introduce the nodes average detachment period as well as the deviation a certain period both by considering it.
2. It inspects that, nodes temporal, social meeting patterns mirror a strong interrelationship. To include this information, two methods design which is named as direct bonding and indirect bonding to forward messages.
3. Based on the direct bonding and indirect bonding methods, an efficient message forwarding technique is proposed to enhance the capability of message forwarding in the opportunistic social network.
4. Through extensive simulations, it is shown that the proposed algorithm is significantly superior to current message forwarding techniques concerning message delivery ratio, dropped messages, overhead, and average latency.

The rest of the paper systematize as follows. Segment 2 presents an overview of previously done research work related to routing protocols. Segment 3 gives the motivation for building the forwarding technique. Segment 4 illustrates the detailed design of the BBFT routing protocol. Segment 5 shows the BBFT algorithm. Segment 6 describes the simulation model using ONE simulator and results. Segment 7 concludes the presented algorithm BBFT and highlights the future work.

2. Literature Review. Research related to forwarding techniques in the opportunistic network done from past decades. Here an overview of these routing techniques is discussed.

Y. Li and R. Bartos, proposed Interaction based Routing Algorithm [2] which utilizes interaction as a two-way effect among nodes like meetings and talking in the social network. It contains two type of components which is direct effects and indirect effects. The immediate effort is considered as a direct impact of a node on another when not disseminated via a third node. The indirect implications regard as an indirect impact of a node on another that propagates through a third node. This algorithm exploits these effects for calculating most popular nodes in the network. F. Li and J. Wu proposed LocalCom [3] protocol which exploits the separation time of nodes to create similarity metrics between every node pair. This local information is then utilized to form communities. The message is forwarded to the destination by gateway nodes in the communities. To develop community structure in this scheme an extended clique clustering used which utilizes virtual links of the network these built communities equipped with specific desirable properties like robust community connection

and controllable diameter, which can provide intra-community communication based on the single-copy source routing. A diffusion-based routing [4], proposed by Vahdat and Becker is epidemic routing. The message is diffused like a virus in the network to deliver to the destination randomly. This routing strategy works on each host that keeps up a buffer containing texts that it has produced as well as words that it is accumulating for some other hosts. A summary vector is maintained by each node which includes saved messages in its buffer space. This protocol suffers from high network traffic.

A probability-based routing [5], proposed by Lindgren et al. is called PRoPHET. In this routing technique probability of a node to send a message to the destination is calculated by its encounter history with other nodes. This protocol also applies transitivity to get predicted next hop. Every node calculates a matrix called delivery predictability before sending the message to the next node or destination. Based on goggles page rank algorithm, A. Mtibaa et al. proposed people rank [6] protocol. In this protocol, the social properties of a social network are used to classify the nodes. A node with higher ranking uses as a forwarder between two nodes. Ranking of nodes is updated whenever nodes make contact. Daly, E. H., et al. proposed SimBet [7] protocol which uses the concept of ego network. A utility function derived from betweenness centrality and the similarity of the node is used to choose the forwarding nodes. Hui, P. et al. proposed BubbleRap [8] forwarding protocol which applies a community-based approach. Each node has assigned itself global rank and local rank for forwarding purpose. Both positions calculated by using betweenness centrality. By using universal ranking, the message transmitted with the help of hierarchical ranking tree mechanism. Later when the message reached to destination node community than the local ranking of nodes of that community used for forwarding. K-clique used as a community detection strategy in this algorithm. Spyropoulos, T. et al. proposed Spray and Wait protocol [9]. This protocol limits the random dissemination of the message. A fixed amount of message copies are spread to other nodes to reach the destination. Jiho Park et al. proposed ABCON [10] message forwarding protocol by using Artificial bee colony approach. This nature-inspired algorithm mostly prefers for multidimensional and multimodal optimization problems. But the author slightly alters the behavior of the bees by defining their responses as follows. If a bee is carrying no message and searching for a message is known as Scout bee. If a bee is taking a single message with it is called Employed bee. An employed bee always forward that carried a message to Onlooker Bee. Percolation centrality of bee nodes is calculated to forward the message to the destination node in the Social Opportunistic Network. Q. Li et al. proposed selfish node routing [11] in delay tolerant network. A selfish node plays maliciously in the forwarding process by not sending the message to next hop. Sanjay K. Dhurandher et al. proposed HBPR [12] which keep track on movement history of the network nodes to predict next hop of the node. The space of the network partition into identical size cells and a cell allocate to each node as its home location which is familiar to other nodes in the system. The parameters used to transmit the data packets to their destination are the speed of the node with which it is moving and the direction of its movement in the network space by using these parameters of the node, a utility metric measure which shows the appropriateness of a node to deliver the message to the destination. Nodes carrying higher utility value than a defined threshold will become the next forwarder of the word. This protocol also includes a method which acknowledges the received messages to help in the buffer management of the intermediate nodes. Research in OppNets has even started focusing on tackling security issues [13, 14] energy efficiency [15, 16] and nature-inspired routing techniques [17, 18].

This paper is presenting a Bonding based forwarding technique, for routing, which effortlessly detects the important nodes using bonding information and exploits the indirect connections to raise the forwarding efficiency. To operate the bonding relationship of the Social Opportunistic Network (SON), a metric called bonding metric is illustrated to represent the neighboring relationship of nodes in the beginning. In the Social Opportunistic Network, nodes mostly have the insight of their connections, also called history of frequency of encounter, which encloses both geographical and temporal information. In Bonding based forwarding technique, we choose the states of the average detachment period to abbreviate nodes insight into a metric that is needed. The average detachment duration calculates for a node based on the encounter frequency and duration of their connections in the vicinity of its neighbors. Indeed, a shorter detachment duration emulates a proximate relationship. Direct bonding metric and indirect bonding metric together depicts a unique way to forward a message using a bonding based graph structure.

3. Motivation. The domain of Social Opportunistic Network (SON) is rising as a distinct paradigm to utilize the social properties of network nodes for designing networking solutions. In recent years, several analysis efforts associated with mobile devices are created to explore the potential of mobile devices. In opportunistic networks, communication is established among mobile devices through routing protocols by using frequent disconnections and uncertain links. According to the Researchers, mobile devices are sharing a close relationship with human society because human beings and vehicles generally carry these devices. Humans always maintain the various types of relationships among themselves called social properties. For example, human beings are usually friendly, and humans with identical behavior mostly devote a long time with each other, being more inclined to share data and resources. Moreover, social properties are crucial to find the nodes mobility pattern and forecast contact opportunities more precisely. Therefore, the SON is capable of determining more positive nodes by using the advantage of the social properties of network nodes.

In literature, many Community-Based and Community-Independent routing protocols have suggested for SON. The Community-Based routing approaches forms communities in the network by taking nodes of common interest. Although, detection and formation of the distributed community are still a difficult task by cause of the dynamic topology changes and complications in data exchange and computation. Due to the difficulty in community detection of these protocols jolts us to adopt Community-Independent routing to select next hop. Besides, the Community-Independent schemes have proposed previously like PROPHET, HiBOP, SimBet, SimBetTS, and People Rank exploit the nodes context information and social properties to send each message to the destination. It reduces the network overhead. As a result, routing protocols following community-independent strategy sidestep the uncovering of community and take on the information of context, social properties (like centrality, tie strength, and similarity), and social meeting pattern to forecast the best forwarder. The implementation of these routing protocols is understandable and manageable. The functioning of Bonding based forwarding technique has been defined to take advantage of Community-Independent routing, to gain a high delivery probability in association with reducing the network overhead, dropped messages and average latency.

4. Proposed Approach. This paper applies the idea to use social information that is more stable and then augment partial intercontact details to implement the efficient message routing protocol in the Social Opportunistic Network. The inspiration behind this proposed idea is that human interaction is a social activity and this social connectedness of common nodes provides a better way to forward a message to any specified destination. Adopting social interaction among people for sending messages in a network has previously been showed to cut down many message replicas at the same time growing the possibility of reaching a letter to its destination.

4.1. Dynamic Weighted Graph. The bonding based forwarding technique, which reveals and employs the essential social bonding to speed up message forwarding in the Social Opportunistic Network, has three steps: Direct bonding, indirect bonding, and message forwarding. A Social Opportunistic Network is considered as a dynamic weighted graph which can characterize as a time progression of network graph denoted by (G_{s1}, G_{s2}, G_{s3}, ...,G_{st}, ...) where $G_{st} = (V_{st}, E_{st}, F_{st}, W_{st})$ represents the state of the SON at time t. In this social graph, V_{st} represents nodes, E_{st} represents links between nodes,$W_{st} = w_{ab}^{t}$, $\in$ [0,1)|a,b $\in V_{st}$ and (a, b) $\in E_{st}$ depicts the bunch of weights on edges at time t, and $F_{st} : E_{st} \rightarrow W_{st}$ is a function that credits weights to links. The pair of node group and link group modify over time. A link (a, b) $\in E_{st}$ if, and only if, these nodes (a,b) are socially bond to each other. In the proposed approach, the social relation is established based on bonding between two nodes; bonding can be (i) direct bonding or (ii) indirect bonding. When a node has direct or indirect bonding with other nodes, then it is considered as an active node to forward the message.

4.2. Direct Bonding. To explore an efficient link metric that utilizes the nodes bonding relations more accurately to transmit a message to the destination node this proposed approach has examined the following human behavioral features of close bonding by adapting average detachment period. An analysis says that the social meeting pattern in the temporal dynamic graph is undoubtedly distinct from that in the aggregated static chart [19, 20]. For that reason, this section concentrates on formulating the social meeting patterns from the temporal aspect. In SON, nodes in the network ordinarily have the insight of their prior contact knowledge with other nodes, also termed contact history, e.g., the inter-contact time, contact duration, and detachment period. The average detachment period encloses both the frequency and duration period reflects a strong

direct bonding relationship with neighbor nodes [3]. In the meantime, the variance of the average detachment period is also listed to mirror the inconsistency in the direct bonding relationship. The deviation in the average detachment period should not neglect. If two conditions have an equal average detachment period, the one along more significant differences would be less preferred since the node would be more uncertain concerning the estimated future detachment period. Thus, this is also necessary to measure the variance of the detachment period dispersion to mirror the deviation using inconsistency metric $VA_{(a,b)}$. The use of average detachment period is more precise to characterize the neighboring relationship rather than only utilizing the inter-contact time or contact duration. Distinctly, a shorter average detachment reflects the consistency in the relationship. A single metric called direct bonding can then deduce from the average detachment period and the variance of the average detachment period. This metric illustrates the relationship among each couple of nodes in SONs and captures the crux of temporal and spatial contact information. So with that, a node-link carrying smaller detachment period and variance show a strong direct bonding to send a message from the source to destination. We denote $D_{(a,b)}$ as the direct bonding between any node a and b with the average detachment period $DP_{(a,b)}$ between two nodes a and b in a particular time window, $VA_{(a,b)}$ as the variance of the average detachment period between nodes a and b in a specific window of time.

$$DP_{(a,b)} = \frac{\sum_i DP_{i(a,b)}}{n_{ab}} \tag{4.1}$$

$$VA_{(a,b)} = VA(DP_{i(a,b)}) = (DP_{(a,b)} - DP_{i(a,b)})^2 \tag{4.2}$$

$$D_{(a,b)} = \frac{1}{DP_{(a,b)} + VA_{(a,b)}} \tag{4.3}$$

where $DP_{i(a,b)}$ denotes the detachment period between any two nodes a and b in the i_{th} encounter, n_{ab} serves as the number of times that a and b are aside from each other. The smaller value of $DP_{(a,b)}$ expresses shorter communication suspension between a and b. The variance of each detachment period between two nodes is measured by subtracting each time nodes encounter in the given time window and the average, and taking the square.

4.3. Indirect Bonding. Every node can calculate its links close bonding with other neighbor nodes against bonding metric. Though, nodes those are not sharing direct bonding can also share indirect bonding. It has recognized that the power of indirectly associated nodes extremely builds upon the number of different direct or indirect paths connecting them. Accordingly, we examine the influence of numerous implicit n hop paths in our interpretation of the indirect bonding. Typically, social ties between individuals are asymmetrically reciprocal. Accordingly, to measure the indirect bonding with the help of the social power of various implicit n hop paths between user a and b, this bonding approach considers two hop of social relationship ($n = 2$), where n shows the length of all possible implicit two-hop paths between a and b. Here a social graph is considered which connects nodes with links weighted based on the power of their direct social bonding communications. For this scenario we are assuming $P_{a,b}$ as the group of divergent implicit paths carrying length n and accompanying indirect bonding between nodes a and b, $D_{(a,c)}$ and $D_{(c,b)}$ is the direct bonding weight of node a, c and b, c. N_a is the neighbor set of node a. Notation c $\in N_a$ is defined as the social power between node a and node b from node a viewpoint over the implicit path of n-hop as:

$$I_{(a,b)} = 1 - \prod_{c \in N_a} 1 - \frac{D_{(a,c)} \times D_{(c,b)}}{2} \tag{4.4}$$

4.4. Forwarding Strategy. This section illustrates the Bonding based forwarding technique summarized in the first algorithm. This forwarding technique imitates the message exchange between nodes a and b. Beginning with the encounter of a message x node a calculates the direct bonding with as neighbor nodes by measuring nodes average detachment period and variance. After figuring the direct bonding of neighboring nodes, the direct bonding weight assigns to node a neighbors links. Subsequently, those nodes are selected for

forwarding among n neighboring nodes as a contender for next hop whose are carrying direct bonding weight values more significant than the bonding weight of current node with destination b (e.g., node as neighbor nodes N_i). Select those nodes from n neighboring nodes as a candidate for next hop whose value of bonding weights ($D_{(a,b)} < D_{(N_i,b)}$). Let this set of nodes be L.

For indirect forwarding, first, indirect bonding is calculated by using implicit n hop paths knowledge between node a and b. The extension algorithm observes two steps:

1. Now each node a, find bonding node k carrying weakest direct bonding weight such that $D_{(a,b)} = min[D_{(k,b)}]$ where $k \in N_i$. This weakest normalized direct bonding weight refer to as ρ_a.
2. For each m of as n-hop implicit nodes, if $I_{(m,b)} \geq \rho_a$, the node m is infused in the candidate peer set of node a. Instinctively, this guarantees that the social power between a and b, positioned at length n in the social opportunistic network graph, is at least as strong as node as the weakest direct bonded node. Let this set of nodes be M.

Now take the nodes belonging to the intersection of sets L and M and then forwards the message copy to these nodes.

5. Algorithms and Sub-Routines of Bonding based Forwarding Approach. In this section, the algorithm and sub-routines used in BBFT algorithm for sending the message from the source to the destination are discussed in detail. Algorithm 1, Algorithm 2 and Algorithm 3 are mentioned below.

Algorithm 1 BBFT Forwarding Algorithm

Q[] direct node
R[] indirect node
L[] is the number of selected direct nodes for message forwarding
M[] is the number of selected indirect nodes for message forwarding

Begin
The source node (a) creates a new message x in its message buffer
Calculate $D_{(a,b)}$ and $I_{(a,b)}$ for source node (a) and destination node (b)
for Each Neighboring Node N_i of (a) **do**
    Calculate $D_{(N_i,b)}$
    Q[]← assign $D_{(N_i,b)}$ to direct links
end for
for Each Neighboring Node N_i of (a) **do**
    Calculate $I_{(N_i,b)}$
    R[]← assign $I_{(N_i,b)}$ to indirect links
end for
for Each direct connected Node Q[] **do**
    if $D_{(N_i,b)} > D_{(a,b)}$ **then**
        L[i] ← $D_{(N_i,b)}$
    end if
end for
for Each L[i] **do**
    find $\rho_a \leftarrow \min[D_{(k,b)}]$ where k ∈ N_i
end for
for Each R[i] **do**
    if $I_{(R[i],b)} \geq \rho_a$ **then**
        M[i]← R[i]
    end if
end for
Forward the message copy to the intersection of both the sets L and M
End

Algorithm 2 Direct sub_bonding routine

Begin
Calculate $DP_{(a,b)}$ between nodes a and b
$DP_{(a,b)} = \frac{\sum_i DP_{i(a,b)}}{n_{ab}}$
Calculate $VA_{(a,b)}$
$VA_{(a,b)} = VA(DP_{i(a,b)}) = (DP_{(a,b)} - DP_{i(a,b)})^2$
Calculate Direct bonding $D_{(a,b)}$ between nodes a and b
$D_{(a,b)} = \frac{1}{DP_{(a,b)} + VA_{(a,b)}}$
return $D_{(a,b)}$
End

Algorithm 3 Indirect bonding sub_routine

Begin
Calculate $D_{(a,c)}$
if $D_{(a,c)}$ calculated **then**
    Calculate $D_{(c,b)}$
    if $D_{(c,b)}$ calculated **then**
        Calculate $I_{(a,b)} = 1 - \prod_{c \in N_a} 1 - \frac{D_{(a,c)} \times D_{(c,b)}}{2}$
    end if
end if
return $I_{(a,b)}$
End

6. Simulation Setup and Results.

6.1. Data Forwarding Experiment.

Algorithm Comparison. In this segment, the performance of the Bonding based forwarding technique has compared against few encounter based routing protocols (Epidemic [4], PROPHET [5], BUBBLE RAP [8], Interaction based routing protocol [2]). Distinctly, the last two have social-aware properties further.

Simulation Setup. The ONE simulator [21] has chosen as a performance evaluation tool for proposed Bonding based forwarding technique. The default framework for the simulation arranged as described in Table 6.1 and Table 6.2. These establish the standard configuration for all the four forwarding protocols Epidemic, Prophet, BubbleRap, and Interaction Based.

The mobile node is partitioned into six groups and out of which two groups serve as pedestrians. The pedestrians belong to first and third group nodes. This pedestrian group form of 40 host nodes each one have carried a buffer size of 5 MB and motion according to the movement model named as Shortest path map based movement model. The nodes of the pedestrian groups follow a walking speed of 0.5-1.5 m/s. The second group also have 40 nodes, but these nodes depict car. The speed of the car nodes are between 2.7-13.9 m/s, buffer size and movement model followed by car group same as pedestrian groups.

The rest three groups form of two nodes each depicting tram nodes. The tram group nodes motion according to the Route based map movement model, have a buffer space of 50 MB and a speed of 710 m/s. A word size of $4500 \times 3400\ m^2$ has defined to the movement models. The mobile nodes of all six groups have a transmission range of 10 m and the transmission speed of 2 Mbps. The communication is initiated between nodes by two distinct interfaces that have detailed in Table 6.2. The pedestrian and car groups adapt Bluetooth interface for communication, at the same time tram groups adopt the high-speed interface. Each message produces in the Social opportunistic network has credited value of 300 sec as a Time-to-live (Message TTL). In the system, messages generated at regular intervals of 25-35 sec and these messages have a message size between 500 KB to 1 MB. The time of simulation is assigned each of the five protocols (Bonding based, Epidemic, Prophet, BubbleRap, and Interaction Based) is 43200 sec.

Table 6.1: Common default settings of simulation specification.

Specification	Value
Simulation area	$4500m * 3400m$
Simulation time	43200
Total no. of nodes	126
Total No. of node groups	6
Speed range	$0.5 - 1.5m/s$
Total no. of movement models	2
Message TTL	$300s$
Buffer size	$5M$
Message size	$500k, 1M$
Transmission speed	$250k$
Transmission range	10

Table 6.2: Group Specific default settings of simulation specification.

Specification	Pedestrian	Car	Tram
No. of groups	2	1	3
Nodes in each group	80	40	6
Node buffer capacity	$5M$	$5M$	$50M$
Speed range	$0.5 - 1.5m/s$	$2.7 - 13.9m/s$	$7 - 10m/s$
Movement Model	Shortest path	Map Route Based	Map Route Based
Interface	Bluetooth	Bluetooth	Bluetooth, High Speed

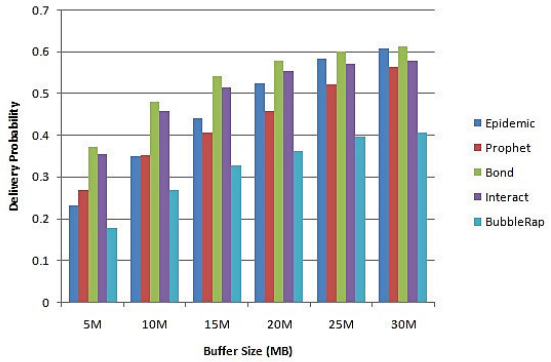
The specified parameters of the network have been fluctuated to produce results for comparison:

1. fluctuating the buffer size (MB): The buffer size in the simulation are assorted as 5, 10, 15, 20, 25 and 30 to observe the performance of Bonding based forwarding technique protocol.
2. fluctuating the Time-to-live(Sec): The TTL in the simulation are assorted as 100, 200, 300, 400, 500.
3. fluctuating the message size: The message size in the simulation assort as (100KB to 200KB), (200KB to 300KB), (300KB to 400KB), (500KB to 1MB) and (1MB to 2MB).
4. fluctuating the number of nodes in the simulation: The total no of nodes in the simulation are assorted as 20, 40, 80, 100 and 120.
5. fluctuating the number of nodes in the simulation: The total no of nodes in the simulation are assorted as 20, 40, 80, 100 and 120.

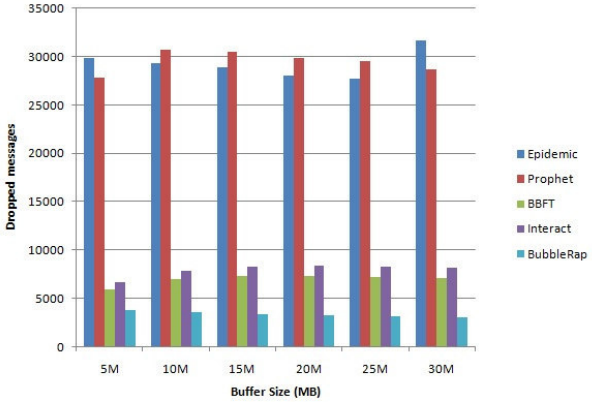
The performance metrics used are:

1. Message delivery probability: the Delivery probability of the nodes implies the estimate of successful acceptance of messages by their destination. Delivery probability is supposed to be high in the course of the simulation.
2. Overhead ratio: Overhead ratio is the average number of forwarded replicas per message. Network resource utilization and bandwidth efficiency evaluate by overhead ratio. An efficient routing protocol causes a small overhead ratio.
3. Dropped messages: Dropped messages show the total number of words descended from the node buffers. Every node contains a limited buffer space in the network, and an excellent routing protocol attempts to lower its utilization.
4. Average latency: It is the average of the difference between the message delivery time and message creation time.

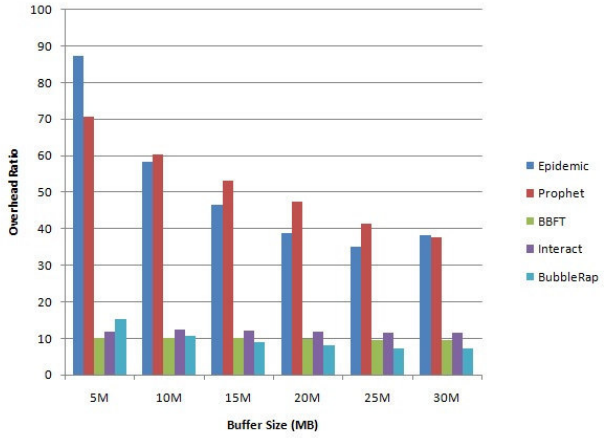
6.2. Comparing the performance of three protocols at varying the buffer size. Figure 6.1 a-d understands the outcome of the increase in buffer size by performance metrics. The performance of BBFT compare to Interaction based, BubbleRap, Prophet, and Epidemic and it can recognize from the Fig.6.1a that BBFT has a greater no. of message delivery as compare to protocols as mentioned earlier. It has interpreted from Fig.6.1a that fluctuating the buffer size of the nodes in the increasing order results in the increase of



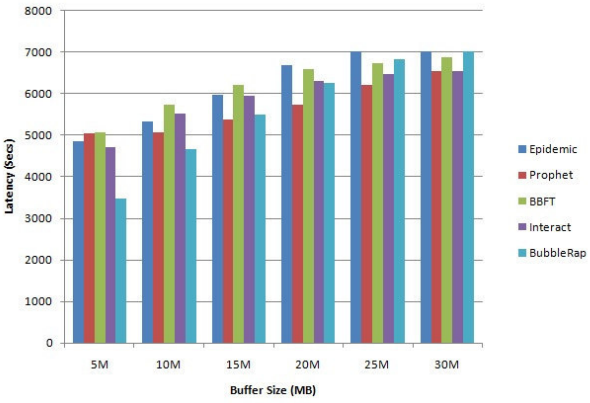
(a) Buffer size vs. delivery probability



(b) Buffer size vs. Dropped messages



(c) Buffer size vs. Overhead ratio

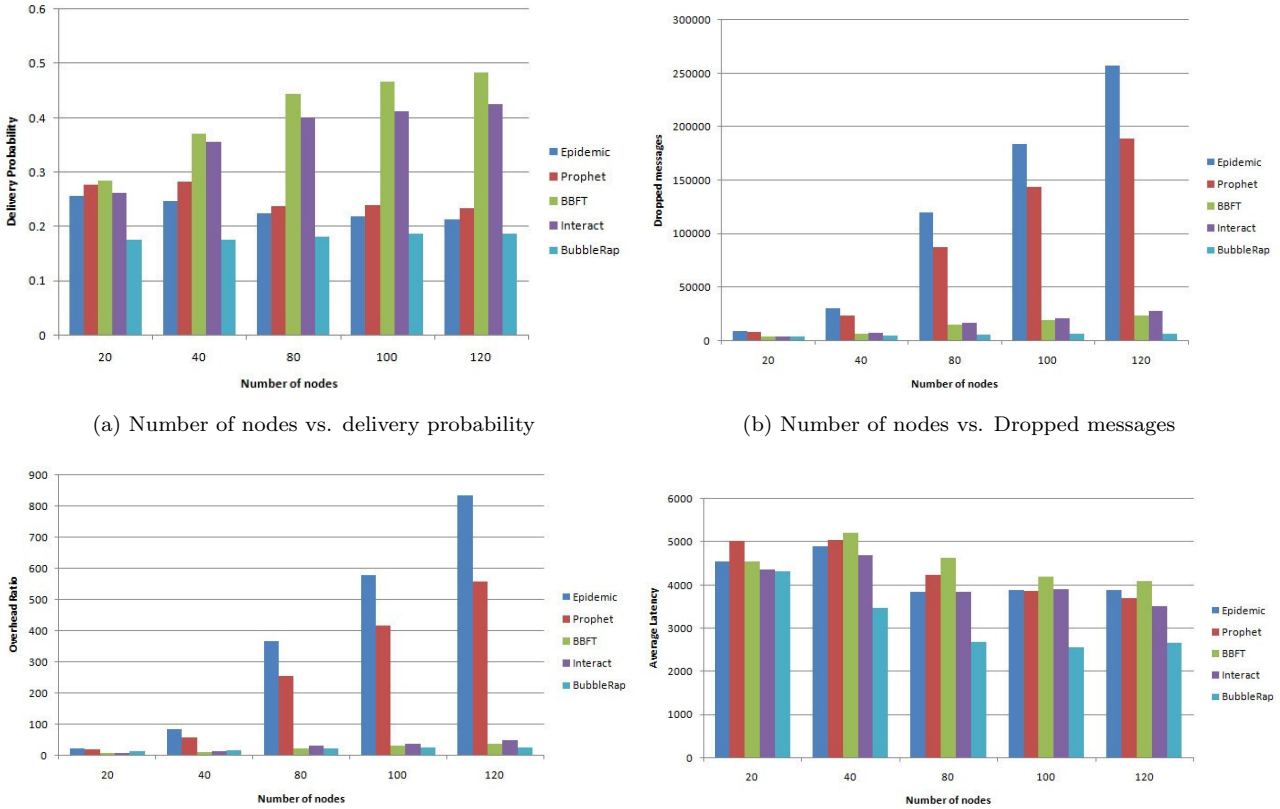


(d) Buffer size vs. Average latency

Fig. 6.1: Performance Metrics versus Buffer size

delivery probability. As the buffer size increase from 5 Mb to 30 Mb, the capacity to hold the messages enhance by the nodes that increase messages forwarded to the destination. Figure 6.1b shows that as the buffer size of nodes increases the number of dropped messages also mostly increases. This above happens because, with the increase of buffer size, no significant improvement observe as the buffer has reached the saturation. That is, messages have already been delivered to the destination before the buffer gets filled and they need to drop. The number of dropped messages of BBFT is far less than Interaction based, Prophet and Epidemic and a little bit more than BubbleRap. Figure 6.1c interprets that the overhead ratio drops with the rise in the buffer size and it is lesser than Epidemic, Prophet, and Interaction based routing protocols. In the process of sending messages, the increase in buffer size results decreases in overhead ratio when a source node has more competent neighbor nodes. The reason behind this overhead ratio drop in BBFT is the appropriate selection of next hop hosts by exploiting the bonding metric. It can be interpreted from Fig.6.1d that the average latency increments with the gain in buffer size. As the buffer size increases the message drop decreases due to that more message will grab the opportunity to be conveyed to the destination. As a result of that average latency is increased. BBFT has increased average latency as compare to Epidemic, Prophet, BubbleRap, and Interaction based routing protocols.

The BBFT is 16.36% better Epidemic, 24.12% better than Prophet, 66.51% better than BubbleRap and



(a) Number of nodes vs. delivery probability

(b) Number of nodes vs. Dropped messages

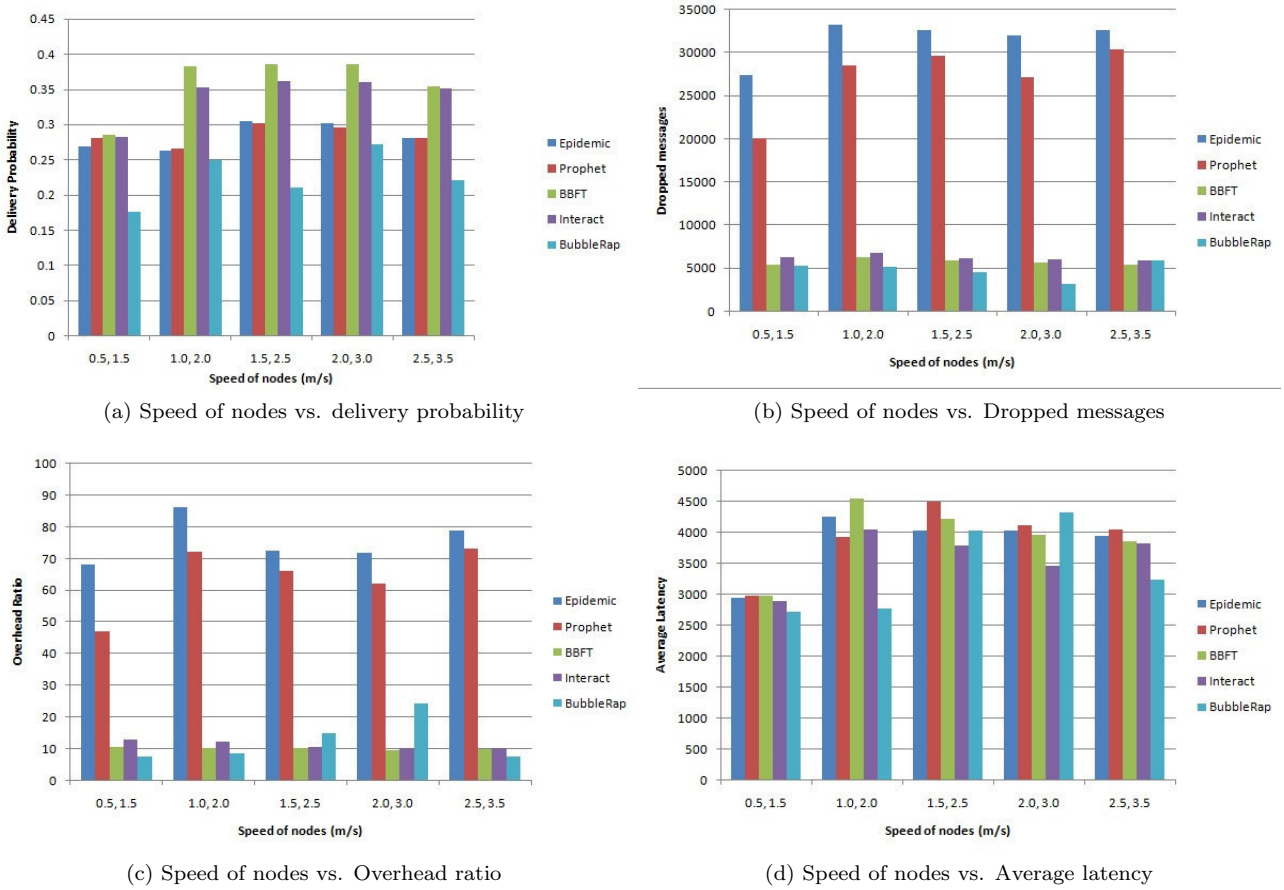
(c) Number of nodes vs. Overhead ratio

(d) Number of nodes vs. Average latency

Fig. 6.2: Performance Metrics versus Total number of nodes

5.06% better than Interaction based regarding the number of messages delivered. The number of messages drops in BBFT is decreased which is 8437 whereas Epidemic, Prophet, and Interaction based has 26667, 35438, 9588. The BubbleRap has 4110 drop which is slightly better. The overhead ratio of BBFT is 80.59% decreased than Epidemic, 80.96% declined than Prophet and 16.84% decreased than Interaction based. The Bubblerap has 3.78 % improved overhead ration as compare to BBFT. The average latency of BBFT is 0.85% more than Epidemic, 9.45% more than Prophet, 4.78% more than Interaction based and 10.25% more than Bubblerap.

6.2.1. Comparing the performance of three protocols at different number of nodes. Figure 6.2 a-d interprets the aftereffect of increase in the number of nodes in the network by performance metrics. The performance of BBFT compares to Interaction based, BubbleRap, Prophet, and Epidemic. It can visualize from Fig.6.2a, by varying the number of nodes from 100 to 500, the average delivery probably for BBFT increases as compared to Epidemic, Prophet, BubbleRap and Interaction based protocols. The message delivery probability for BBFT hikes as more hosts is joined, which can associate with the corresponding gain in the number of message transmission in the network. It can be inspected in Fig.6.2b, as the number of hosts increases in the network, the number of dropped messages are decline. This reaction is due to the hike in the number of message transmission and is in interrelation with huge delivery probability examined for message delivery. The no. of dropped messages for BBFT decrease as compare to Epidemic, Prophet, and Interaction based protocols. By scanning Fig.6.2c, it observes that by changing the number of nodes, the overhead ratio of messages for BBFT decrease as compared to Epidemic, Prophet, and Interaction based protocols. The minimum overhead ratio of BBFT exploits more convoluted information from the network parameters and then process it to compose better



(a) Speed of nodes vs. delivery probability

(b) Speed of nodes vs. Dropped messages

(c) Speed of nodes vs. Overhead ratio

(d) Speed of nodes vs. Average latency

Fig. 6.3: Performance Metrics versus Speed of nodes

decisions about next hop choice which lowers the number of message replicas in the network. It reduces network resource utilization and thus cutting down the overhead ratio. It can interpret from Fig.6.2d that in BBFT the average latency hikes with increment in the number of messages. The reason is that as the nodes raise, the communications a sender node can have with its neighbor nodes also rises. Due to that BBFT uses some time in deciding to choose the best possible node as next hop for conveying the message. BBFT has increased average latency as compare to Epidemic, Prophet, BubbleRap, and Interaction based routing protocols.

The BBFT is 77.3% better Epidemic, 61.71% better than Prophet, 126.5% better than BubbleRap and 10.47% better than Interaction based regarding the number of messages delivered. The number of messages dropped in BBFT decrease remarkable which is 13091 whereas Epidemic, Prophet, and Interaction based have 119961, 90176, 14981. The BubbleRap has 4798 drop which is better. The overhead ratio of BBFT is 94.69% decreased than Epidemic, 92.29% declined than Prophet and 21.58% decreased than Interaction based. The BubbleRap has 4.8% improved overhead ration as compare to BBFT. The average latency of BBFT is 7.61% more than Epidemic, 3.82% more than Prophet, 11.7% more than Interaction based and 44.38% more than BubbleRap.

6.2.2. Comparing the performance of three protocols at different speeds. The performance of the distinct protocols observe as a parameter of mobility, the range of speed of the 80 nodes in the two pedestrian groups (as per the baseline settings) is varied from 0.51.5 m/s to 2.53.5 m/s. The outcome of the delivery probability, the number of dropped messages, overhead ratio, and average latency, has been illustrated in Fig.6.3

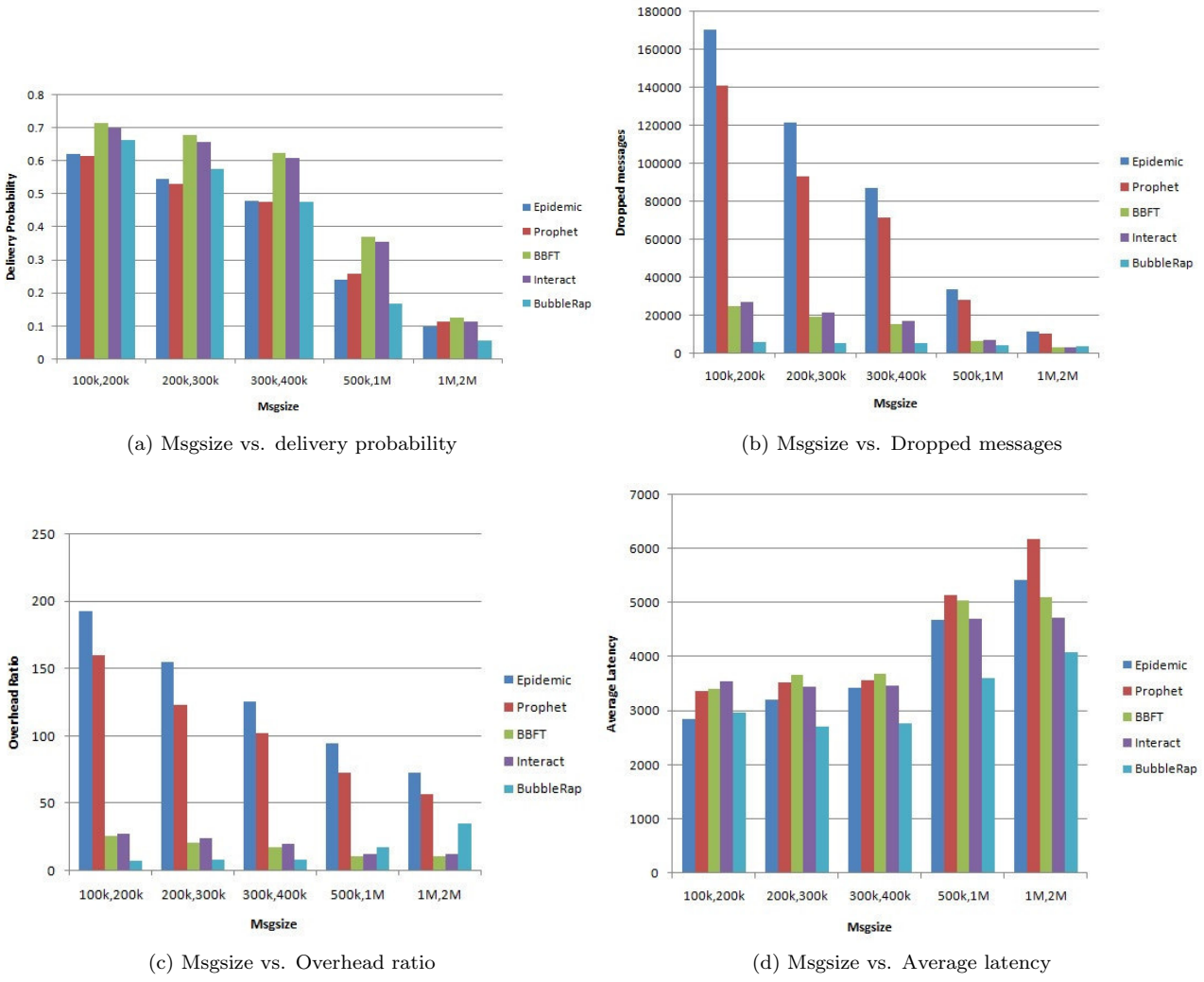
ad accordingly. By varying the speed of the nodes in the increasing order results gain in delivery probability of the messages also. As this scene from Fig.6.3a that that BBFT has a leading no. of message delivery in contrast to the Interaction based, BubbleRap, Prophet and Epidemic protocols. Figure 6.3b interprets that when messages are transmitted further into the network along with an expansion in mobility of the hosts, this results to a higher opportunity of the message reaching nearer to its destination host in the network. The number of dropped messages for BBFT increments randomly with the increase in speed of the nodes, as message propagation raises with the escalation in the node mobility which results in more number of dropped messages. Changing the speed of the nodes leads in a small decrease in the overhead ratio for BBFT which validate as the message transmissions in the network increase Fig.6.3c. It can interpret from Fig.6.3d that in BBFT the average latency hikes randomly with the increase in speed of nodes. BBFT has increased average latency as compare to Epidemic, Prophet, BubbleRap, and Interaction based routing protocols.

The BBFT is 26.42% better than Epidemic, 25.59% better than Prophet, 59.15% better than BubbleRap and 4.95% better than Interaction based regarding the number of messages delivered. The number of messages dropped in BBFT is decrease remarkable which is 5678 whereas Epidemic, Prophet, and Interaction based has 31648, 27145, and 6162. The BubbleRap has 4753 drop which is slightly better. The overhead ratio of BBFT is 86.82% decreased than Epidemic, 84.49% declined than Prophet, and 10.11% decreased than Interaction based and 19.9% decreased than the BubbleRap. The average latency of BBFT is 1.9% more than Epidemic, 0.04% more than Prophet, 8.57% more than Interaction based and 14.70% more than BubbleRap.

6.2.3. Comparing performance of three protocols at Varying the Message Size. Figure 6.4 a-d interprets the outcome of the increase in message size by performance metrics. The performance of BBFT compare to Interaction based, BubbleRap, Prophet, and Epidemic and it can be recognized from Fig.6.4a that BBFT has larger no. of messages delivered as compared to protocols as mentioned above. It interprets from Fig.6.4a that fluctuating the message size of the nodes in the increasing order results in a decrease of delivery probability. The message sizes increase from 100k-200k to 1M-2M, the node buffer capacity to hold the messages decrease which results in the reduction of messages forwarded to the destination. Figure 6.4b shows that as the message size of nodes increases the number of dropped messages also mostly decreases. The BBFT has lesser number of lost messages than Interaction based, Prophet and Epidemic and a little bit more than BubbleRap. Figure 6.4c shows that the overhead ratio of BBFT decreases with increase in the message size and is less than PRoPHET. Figure6.4d Shows that the average latency increases with the increase in message size. This interpretation is because, with increased message size, number of messages are dropped from it reduces. Thus, more words get the chance to be delivered to the destination which increases the average latency.

The BBFT is 21.06% better than Epidemic, 20.76% better than PRoPHET, 3.02% better than INTERACT and 22.98% better than BubbleRap regarding Delivery Probability. The number of messages dropped in BBFT is decrease remarkable which is 13514 whereas Epidemic, Prophet, and Interaction based have 84736, 68648, and 14784. The BubbleRap has 4611 drop which is better. The overhead ratio of BBFT is 87.02% less than Epidemic, 83.91% less than PRoPHET, 12.94% less than INTERACT and 11.48% more than BubbleRap. The latency is 6.40% more than Epidemic, 4.11% less than PRoPHET, 4.95% more than INTERACT and 22.86% more than BubbleRap.

6.2.4. Comparing performance of three protocols at Varying the Time to Live(TTL). Figure 6.5 a-d interprets the outcome of the increase in TTL by performance metrics. The performance of BBFT compare to Interaction based, BubbleRap, Prophet, and Epidemic and it can recognize from the Fig.6.5a that BBFT has a high delivery ratio as compare to protocols as mentioned earlier. It has interpreted from Fig.6.5a that changing the TTL of the message in the increasing order results in the increase of delivery probability. As the TTL increase from 100 to 500, messages are getting enough time to live in the network to reach the destination. Figure 6.5b shows that as the TTL of messages increases the dropping varies a little only. It happens because, with the increase in TTL, no significant improvement observe as we are following the Drop Oldest strategy. The number of dropped messages of BBFT is far less than Interaction based, Prophet and Epidemic and a little bit more than BubbleRap. Figure 6.5c shows that the overhead ratio decreases slightly with the increase in the TTL. However, the overhead ratio of BBFT remains much less than the Interaction based, BubbleRap, Prophet and Epidemic protocols because of its efficient selection of next hop nodes using the Bonding Metric. It can observe from Fig.6.5d that the average latency increases with increase in TTL. This



(a) Msgsize vs. delivery probability

(b) Msgsize vs. Dropped messages

(c) Msgsize vs. Overhead ratio

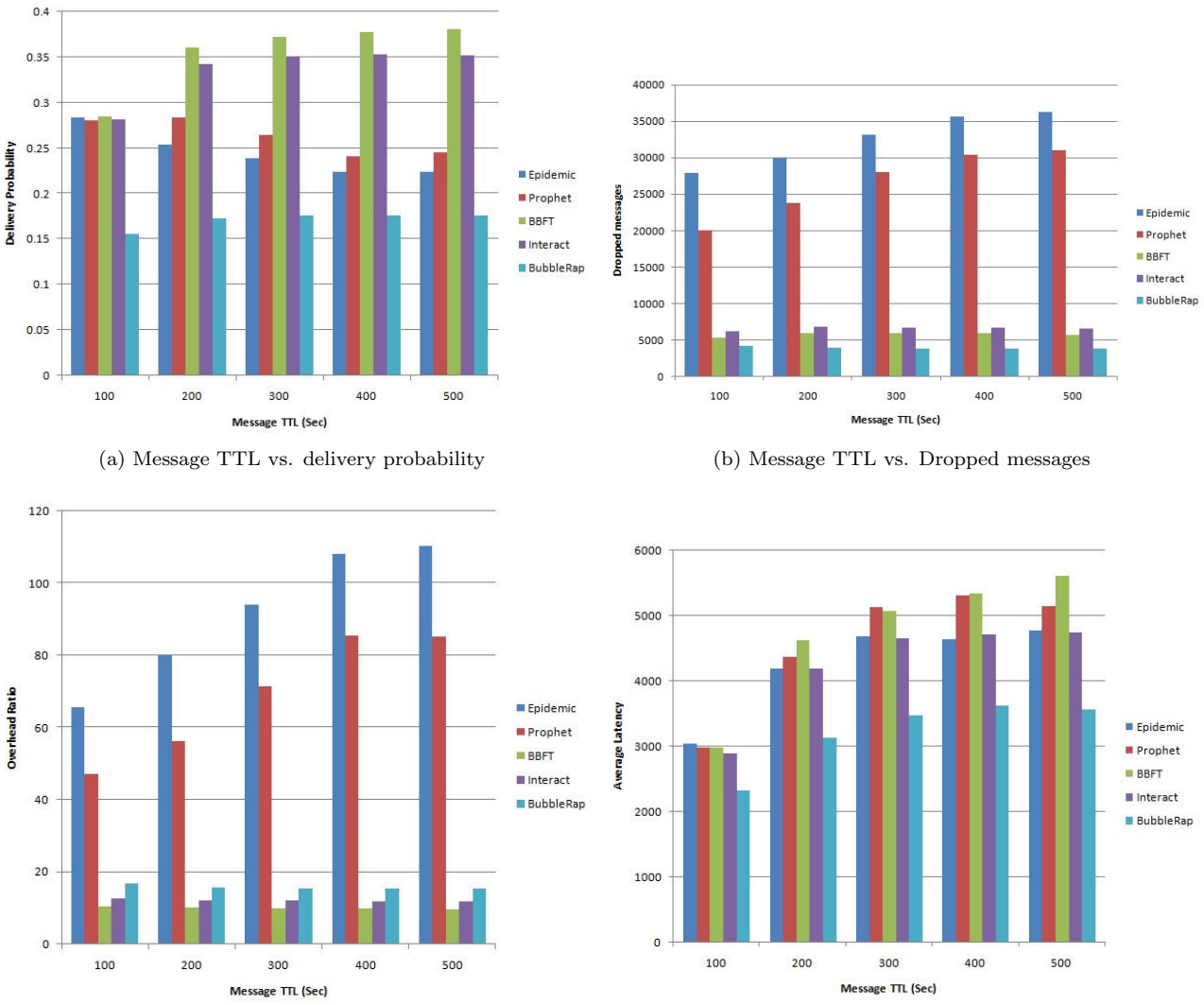
(d) Msgsize vs. Average latency

Fig. 6.4: Performance Metrics versus Msgsize

observation is because with increased TTL lesser number of messages are dropped from it. Thus, more words will get the chance to be delivered to the destination which results in increased average latency.

The BBFT is 31.03% better than Epidemic, 25.91% better than PRoPHET, 5.26% better than INTERACT and 51.72% better than BubbleRap regarding the number of messages delivered. The number of messages dropped in BBFT decrease remarkable which is 5821 whereas Epidemic, Prophet, and Interaction based have 32617, 26651, and 6616. The BubbleRap has 3952 drop which is better. The overhead ratio of BBFT is 817.39% less than Epidemic, 591.59% less than PRoPHET, 21.74% less than INTERACT and 56.93% less than BubbleRap. The average latency of BBFT is 9.81% more than Epidemic, 3.02% more than PRoPHET, 10.30% more than INTERACT and 31.90% more than BubbleRap.

7. Conclusion. In this attempt, a novel routing algorithm BBFT propose that evaluates the strength of direct and indirect bonding and avail it to extend the preferred set of nodes in the Social Opportunistic Network while preserving the social patterns that are the most desirable in SON. Specifically, by taking into account both the intensity of the number of encounters and the number of associated paths results in an average



(a) Message TTL vs. delivery probability

(b) Message TTL vs. Dropped messages

(c) Message TTL vs. Overhead ratio

(d) Message TTL vs. Average latency

Fig. 6.5: Performance Metrics versus Message TTL

detachment period and variance, the preferred set of nodes extend. The enlarged node set incorporate not only the directly connected nodes in the SONs but also companion nodes located two hops distant with whom node has an indirect bonding higher than that of nodes weakest directly bonded node. The BBFT interpret against Epidemic, PRoPHET, BubbleRap, and Interaction based routing protocols. It functions significantly fine concerning the delivery probability of the nodes, as a result of selecting improved and dependable next hop nodes. The overhead ratio and dropped messages for BBFT are also lower than the other protocols. In future, the BBFT can interpret with more routing protocols like HiBOp and CAR. Average latency is one parameter in which other protocols still behave better than BBFT and hence enhancing the average message latency while maintaining the substantial delivery probability of BBFT is another field of future work. Future efforts will be executed to accomplish the BBFT protocol secure, and the issue of energy consumption also taken into account.

REFERENCES

[1] D. J. Brass, K. D. Butterfield, and B. C. Skaggs, "Relationships and unethical behavior: A social network perspective," *Academy of management review*, vol. 23, no. 1, pp. 14–31, 1998.

[2] Y. Li and R. Bartos, "Interaction based routing algorithm for opportunistic mobile social networks," in *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*. IEEE, 2017, pp. 492–497.

[3] F. Li and J. Wu, "Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks." in *SECON*, vol. 9, 2009, pp. 574–582.

[4] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," 2000.

[5] A. Lindgren, "D0ria a, schelen o," *Probabilistic routing in intermittently connected networks*, vol. 7, no. 3, p. l, 2003.

[6] A. Mtibaa, M. May, C. Diot, and M. Ammar, "Peoplerank: Social opportunistic forwarding," in *Infocom, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.

[7] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 32–40.

[8] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[10] J. Park, J. Lee, S.-K. Kim, K. Jang, and S.-B. Yang, "A forwarding scheme based on swarm intelligence and percolation centrality in opportunistic networks," *Wireless Networks*, vol. 22, no. 8, pp. 2511–2521, 2016.

[11] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[12] S. K. Dhurandher, D. K. Sharma, I. Woungang, and S. Bhati, "Hbpr: history based prediction for routing in infrastructure-less opportunistic networks," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*. IEEE, 2013, pp. 931–936.

[13] A. Chhabra, V. Vashishth, and D. K. Sharma, "A fuzzy logic and game theory based adaptive approach for securing opportunistic networks against black hole attacks," *International Journal of Communication Systems*, vol. 31, no. 4, p. e3487, 2018.

[14] ——, "A game theory based secure model against black hole attacks in opportunistic networks," in *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*. IEEE, 2017, pp. 1–6.

[15] ——, "Seir: A stackelberg game based approach for energy-aware and incentivized routing in selfish opportunistic networks," in *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*. IEEE, 2017, pp. 1–6.

[16] D. K. Sharma, S. K. Dhurandher, M. S. Obaidat, A. Bansal, and A. Gupta, "Genetic algorithm and probability based routing protocol for opportunistic networks," in *Computer, Information and Telecommunication Systems (CITS), 2017 International Conference on*. IEEE, 2017, pp. 58–62.

[17] D. K. Sharma, S. K. Dhurandher, I. Woungang, A. Bansal, and A. Gupta, "Gd-car: A genetic algorithm based dynamic context aware routing protocol for opportunistic networks," in *International Conference on Network-Based Information Systems*. Springer, 2017, pp. 611–622.

[18] D. K. Sharma, S. K. Dhurandher, D. Agarwal, and K. Arora, "krop: k-means clustering based routing protocol for opportunistic networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2018.

[19] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 299–308.

[20] H. Zhou, J. Chen, H. Zhao, W. Gao, and P. Cheng, "On exploiting contact patterns for data forwarding in duty-cycle opportunistic mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4629–4642, 2013.

[21] A. Keranen, "Opportunistic network environment simulator," *Special Assignment report, Helsinki University of Technology, Department of Communications and Networking*, 2008.

Edited by: Khaleel Ahmad
Received: Nov 11, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1450
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 17–26. http://www.scpe.org © 2019 SCPE

ANALYSIS OF DELAY-TOLERANT ROUTING PROTOCOLS USING THE IMPACT OF MOBILITY MODELS

MD. SHARIF HOSSEN* AND MUHAMMAD SAJJADUR RAHIM†

Abstract. Intermittently connected mobile networks are sparsely connected wireless ad-hoc networks where there is no end-to-end path from a source device to a destination. Generally, these paths do not exist. Hence, these devices use intermittent paths using the concept of store-and-forward mechanism to successfully accomplish the communication. These networks are featured by long delay, dissimilar data rates, and larger error rates. Hence, we look into the analysis of several delay-tolerant routing protocols, e.g., epidemic, spray-and-wait, prophet, maxprop, rapid, and spray-and-focus using opportunistic network environment simulator. At first, the investigations of the above considered routing protocols are done across three mobility models namely random direction, random walk, and shortest path map based (SPMB) movement model for node impact only. Then, we evaluate these routing protocols against the impacts of message copy, buffer, and time-to-live using SPMB movement model considering the results of node impact. We use three metrics, and the results show that spray-and-focus yields good performance for showing higher delivery, lower latency, and lower overhead among all routing techniques, while epidemic is poor.

Key words: Delay-tolerant network; ad hoc networks; communication; routing; replication; simulation

AMS subject classifications. 68M12, 68U20

1. Introduction. With the increase of portable devices (e.g. smartphones, laptops), a class of ad-hoc networks have become popular nowadays [1, 2, 3, 4, 5, 6, 7, 8] which are known as delay-tolerant networks (DTNs) [9]. DTNs [10] are also called intermittently connected (IC) [11]. These are wireless mobile adhoc networks where devices can not build an end-to-end route. Examples of such networks are satellite communication [12], wildlife tracking,[13], military, and vehicular [14, 15].

However, these networks use a technique named *store-carry-and-forward* to successfully communicate among mobile devices [16, 17, 18, 19, 20, 21] by passing the information to intermediate devices, where messages are stored in respective buffer and are forwarded to other relays in the network. Message copy passing can be in two ways, firstly only using a single copy called forwarding based and secondly using two or more copies called replication based [22, 23, 24].

Only replication based routing techniques mentioned in Sec. 2 are used in this research. Three mobility models are used to analyse their impact on the performance of routing protocols in the considered IC mobile network scenario. The performances are analysed for changing message copies, node impact, TTL, and buffer on the impact of mobility models. Using ONE simulator we see that spray-and-focus shows good performance compared to all and epidemic poor.

Remaining part of this research is written as follows. In Sec. 2, we discuss the classification of routing strategies. Section 3 involves different parameter setting with simulator description. Different mobility models are summarized in Sec. 4. Then, the explanation of different arguments is included in Sec. 5. Finally, the summary with future endeavors is discussed in Sec. 6.

2. Routing Protocols. This section summarizes the general classification and description of routing techniques.

2.1. Classification of Routing. Generally, two categories of routing protocols are single-copy and multi-copy. In the first case, only one message transmission through the nodes is possible [11, 30]. In the second case, two or more message transmissions at a time are possible in the network [25]. The first case exhibits lower message transmission with the higher delay due to network partition [26, 27]. While, the second case can provide better delivery due to the message replication [28, 29].

*Lecturer, Department of Information & Communication Technology, Comilla University, Comilla-3506, Bangladesh (sharif5613@gmail.com).

†Associate Professor, Department of Information & Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh, (sajid_ice@ru.ac.bd).

2.2. Forwarding Based. Here, we discuss forwarding based routing techniques. As an example, we consider the source device as X, the destination as Z, and intermediate devices can be any of the following symbols: Y, P, Q, R, S.

First Contact. Using this routing, a device can send a copy to the nearest first device found. As an example, the source device X can forward a single copy to the first available device (say Y). After successfully sending a message to Y, the device X does not store any copy. Y can also forward a message similar to X, and this process continues until found Z [26].

Direct Delivery. Using this routing, a device can send a message directly to a device without sharing its copy to anyone. That is, no intermediate nodes are used here. As an example, source device X directly sends the message to Z without taking any help of intermediate devices: Y, P, Q, etc. So, this technique [31] faces high delay.

2.3. Replication Based. Here, replication based routing techniques are summarized. we consider the source device as X, the destination device as Z and the intermediate devices as Y, P, Q, R, S, etc.

Epidemic. This routing is the first among the replication based concepts where total replicas are very high. Hence is named as flooding. As an example, X can generate and forward several message copies to the intermediate devices, say, P, Q, R, S, which can store a message copy if they do not have that copy. These devices then replicate and continue this process until found Z [32, 33]. This routing exhibits much more overhead for its replication process [22].

Prophet. Using this routing, if X wishes to forward a message copy to its intermediate devices, say, P, Q, then depending on the past record of delivery, X forwards the message to either P or Q, i.e., if the probability of meeting Z from P is greater than the probability of meeting Z from Q, then X forwards the message to P, and P then easily contact to Z [33].

Prophet version-2. Prophet suffers from the delivery problem if the transitivity among nodes is not zero. Prophet version 2 solves this issue by calculating the maximum probability among two frequently encountered nodes having transitivity[34].

Rapid. Using this routing, devices can use one of the three optimization metrics, i.e., reduction of average delay, missed deadline, and maximum delay. Hence, the devices can forward a message to a relay calculating the utility value, which is related to the bandwidth, and the buffer capacity in the network [35].

Maxprop. Using this routing, a device forwards a message copy to an intermediate device having lower hop counts using Dijkstra shortest path algorithm [36].

Spray-and-wait (SNW). This routing has two variations: vanilla, and binary. In the first variation, X can send N copies to the first N devices where all devices including X are in the waiting phase to contact Z. While, in the second variation, X having N copies sends N/2 copies to P, Q, R, and S where only one device, say, P receives N/2 copies. If a device has a copy, then it can not store another copy. P again sends N/4 copies to the next first devices available and continues it until having a single copy (say S carrying it). This indicates that S is in the waiting phase and directly contacts Z [37]. In this work, binary version (B-SNW) has been used.

Spray-and-focus (SNF). A device having only a single copy is in the waiting phase as the description of SNW, but SNF routing [38] forwards this single copy to another relay instead of destination which is the focus phase having a criteria as following: P can forward a single copy instead of Z to Q if the utility of Q to meet Z is greater than the utility of P to contact Z plus a fixed value.

3. ONE Simulator with Simulation Setting. We discuss here ONE simulator and the parameters used in the simulation.

3.1. ONE Simulator. ONE is a java unified mapping platform to trace the visual contact of mobile nodes, message passing, event generation of real scenario which is packaged as a single java project in [31, 39, 40]. A simple simulation scenario is shown in Fig. 3.1.

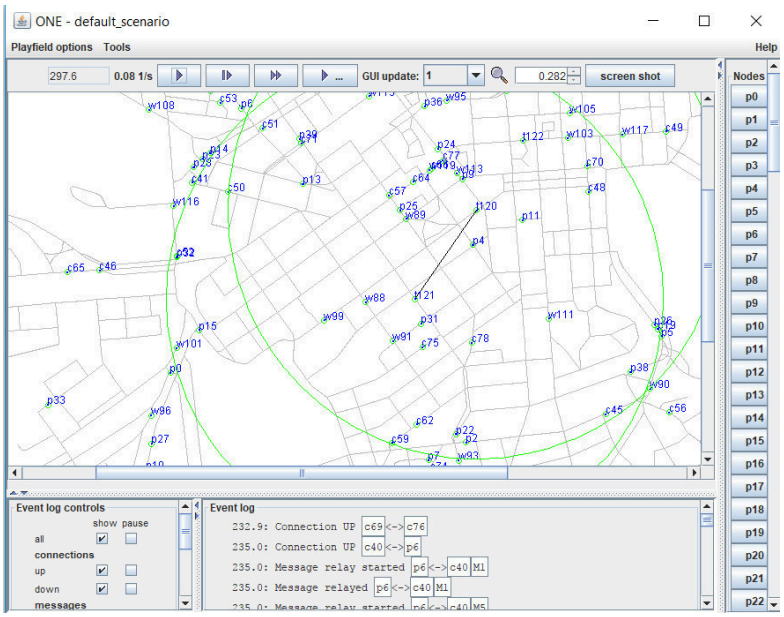


FIG. 3.1. *Simulation scenario.*

3.2. Simulation Setting. During the evaluation, we have used a laptop: HP Pavilion, processor: Intel(R) Core(TM) i7-6500U CPU @ 2.50 GHz, installed memory (RAM): 8.00 GB, and 64-bit OS. In the simulation, we have used various parameters which are summarized as follows. We have used 24 hours as the simulation time, Bluetooth as an interface with transmit velocity 250 kbps and range 10 m. Message size has been in the range 500 KB to 1 MB. We have considered simulation area size as 4500m x 3400 m (width x height). The fixed values for message copies, number of hosts, buffer, and time-to-live are 2 (msg/min), 100, 5 MB, 300 (minutes) respectively. When one parameter is changed, other values are kept at fixed. Hence, we have changed the message copies as 2, 3, 4, 5, 6, and 10 (msg/min). Node's buffer size has been changed as the values of 5, 10, 15, 20, 25 (MB). We have changed time-to-live as 50, 100, 150, 200, 250, and 300 (minutes). In this case, during the change of TTL (50, 100, 150, 200, 250, and 300 minutes), the fixed values for message copies, number of hosts, and buffer size are 2 (msg/min), 100, and 5 MB.

4. Mobility Models. We have considered three mobility models which are summarized below.

4.1. Random Walk (RW). Using this mobility [41, 42, 43], devices randomly move from one location to another until found the destination with a fixed speed within a predefined range.

4.2. Random Direction (RD). There are three categories of this mobility [44, 45, 46]. In the first case (we use here) [44], a device with a predefined speed starts to move in a particular direction randomly until hits the boundary of the simulation area. After touching the boundary wall, it takes a pause and moves to another direction to meet the destination.

4.3. Shortest Path Map Based (SPMB) Movement. This mobility uses the Dijkstra algorithm to find out the shortest path between two devices [47].

5. Results and Discussion. Firstly, we see the analysis of three mobility models discussed in Sec. 4 using node impact. Then, taking decisions from the results we investigate the DTN routing protocols' performance for changing various impact of parameters.

5.1. Mobility Based Node Impact on DTN Routing. Here, we analyze the node impact on the mobility models.

5.1.1. Delivery. Both RW and RD follow the random nature but different strategy to trace a path from one to another position, whereas SPMB movement model follows Dijkstra strategy to determine the minimum distance of reaching a node from other. Hence, we see from Fig. 5.1 that all routing techniques provide higher delivery using SPMB mobility while lower delivery for RD in the considered scenario.

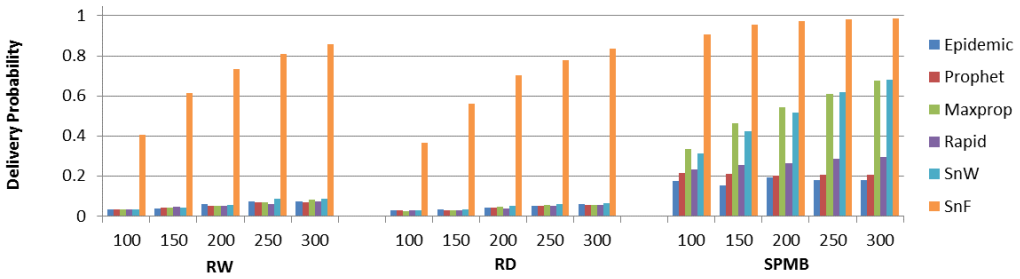


FIG. 5.1. *Delivery probability with varying number of nodes.*

5.1.2. Latency. From Fig. 5.2, we see that with the increase of devices in the network, latency increases for three cases i.e., RW, RD, and SPMB. However, we see lower latency using SPMB mobility while higher latency in other two mobility models. Therefore, we can conclude that SPMB will show the desired performance in our scenario.

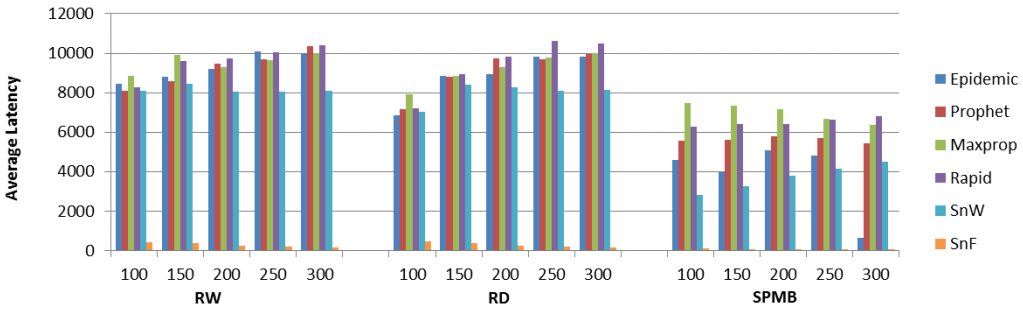


FIG. 5.2. *Average latency with varying number of nodes.*

5.1.3. Overhead. Although the overhead is higher for all routing strategies except for SNF and SNW using SPMB mobility while lower in other two mobility models, we see that SNF and SNW have lower overhead in SPMB mobility rather than using RW and RD as shown in Fig. 5.3.

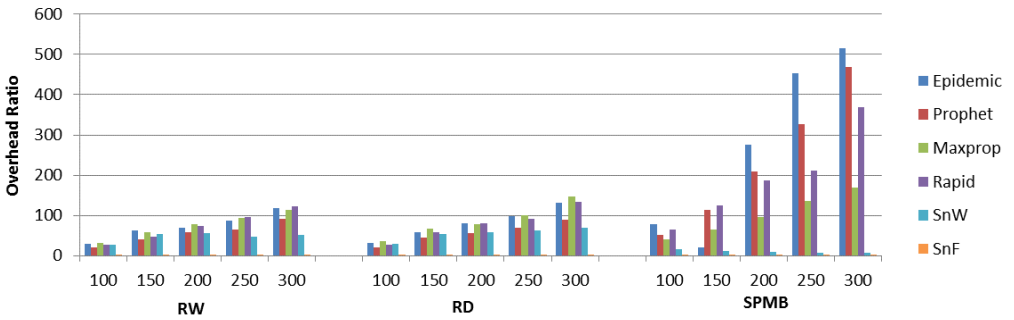
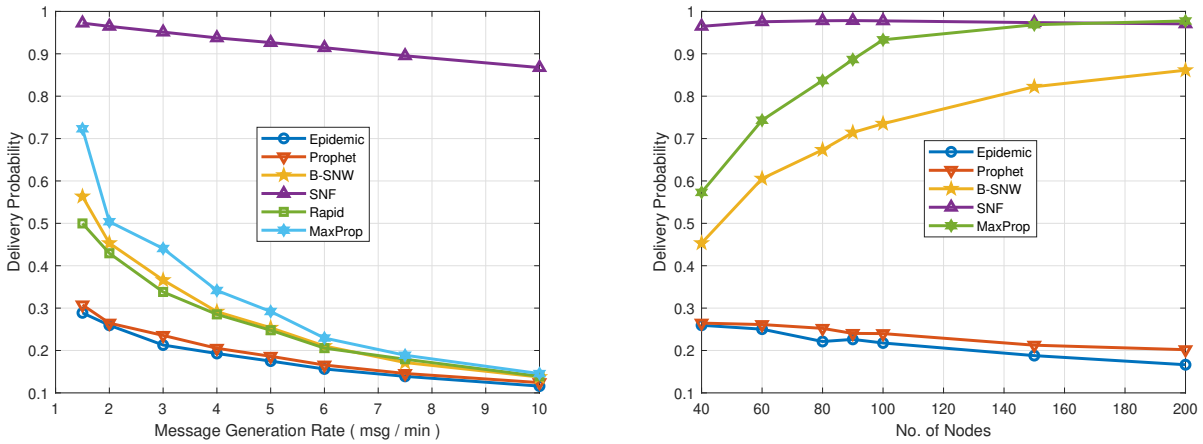


FIG. 5.3. *Overhead ratio with varying number of nodes.*

Therefore, considering our scenario it is apparent that SPMB mobility is preferred instead of using RW and RD due to ensure higher delivery, lower latency and lower overhead (in the case of B-SNW and SNF).

5.2. SPMB Analysis on DTN Routing. From the investigation of Sec. 5.1, it is evident that for our scenario routing techniques show better performance using SPMB mobility instead of using RW and RD. We do the analysis of routing techniques discussed in Sec. 2.3 using SPMB mobility for changing certain network parameters with respect to three metrics.

5.2.1. Delivery with Message Copies and Node Density. In replication based routing, major problem is to control the limit of replicating of messages. More techniques have been proposed where SNF can control the maximum replication while Epidemic can forward the copies randomly with the nature of flooding. So, in both cases (shown in Figs. 5.4(a), and (b)) i.e., for changing message copies and mobile devices, we see that SNF routing exhibits much delivery while Epidemic the less. Since rapid routing needs more resources for the heavy density of devices, we can not include this in Fig. 5.4(b) due to the complexity of calculation.



(a) Delivery probability with changing message copies. (b) Delivery probablity with changing node density.

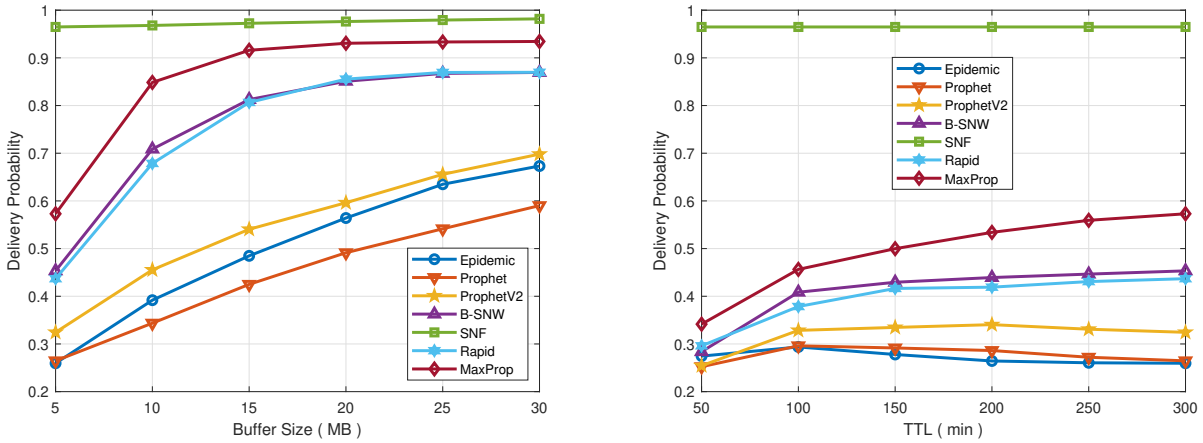
FIG. 5.4. *Delivery probability with changing message copies and node density.*

5.2.2. Delivery with Buffer and TTL. We know if we have more space, then we can store more soft materials. The same thing here is that with the increase of buffer, devices can store more copies that's why delivery increases. While TTL is a lifetime of a message copy to exist in the network and so increases delivery. In both cases shown in Figs. 5.5(a) and 5.5(b), we see better performance for SNF while lower for Epidemic and Prophet.

Therefore, from the investigation of Sec. 5.2.1 and 5.2.2, it is clear that SNF routing shows better delivery and Epidemic the lower compared to other routing protocols mentioned in Sec. 2.3.

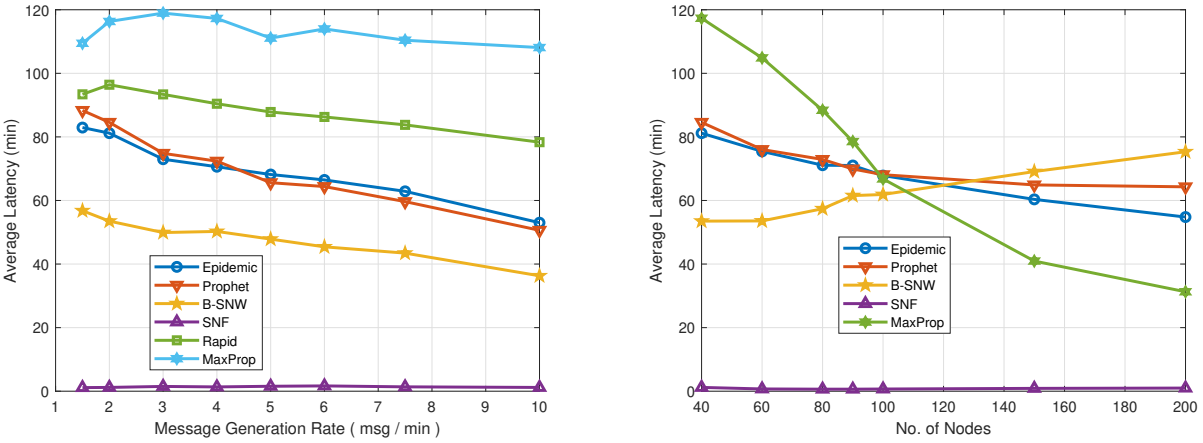
5.2.3. Latency with Message Copies and Node Density. We see that with the increase of message copies and mobile devices, latency decreases for all routing techniques except for B-SNW which increases latency for increasing devices. In both cases, we see lower delay for SNF and higher for Maxprop as shown in Figs. 5.6 (a) and 5.6 (b). For the same reason discussed in 5.2.1, we can not include Rapid routing (Fig. 5.6 (b)).

5.2.4. Latency with Buffer and TTL. As shown in Figs. 5.7 (a) and 5.7 (b), we see that latency increases with the increase of buffer and TTL for all routing techniques except Maxprop (greater than the latency in SNF). For varying buffer, Rapid and Epidemic shows very high latency because Epidemic does not follow any strategy for the limit of copies of messages while Rapid is resource thirsty which causes more delay of transferring copies from one to another. While for changing TTL, Maxprop provides very high growing latency. Latency for SNF is constant and lower for changing TTL. In both cases, we see lower delay for SNF.



(a) Delivery probability with varying buffer size. (b) Delivery probability with varying TTL.

FIG. 5.5. *Delivery probability with varying buffer and TTL.*



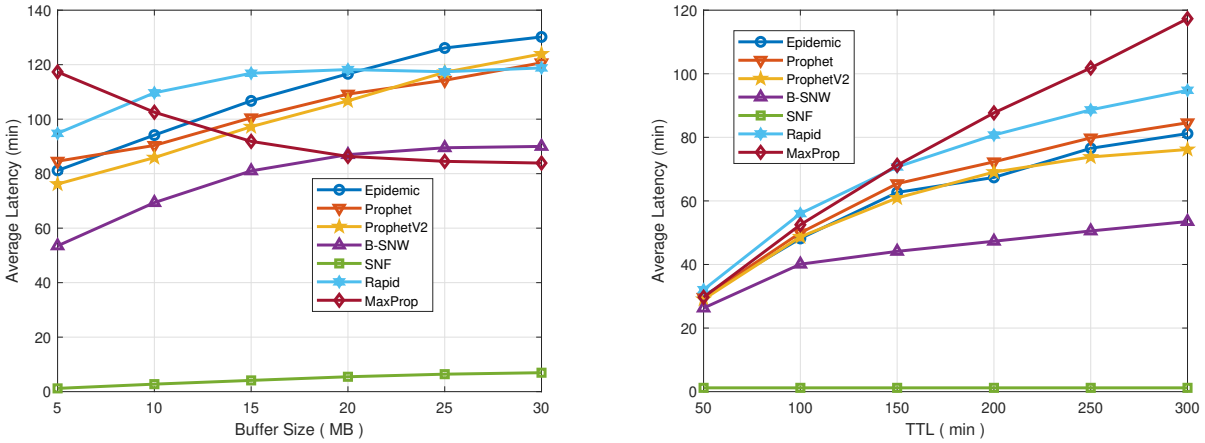
(a) Average latency with varying message copies. (b) Average latency with varying node density.

FIG. 5.6. *Average latency with varying message copies and node density.*

Therefore, from the investigation of Sec. 5.2.3 and 5.2.4, it is clear that SNF shows lower delay compared to all routing techniques mentioned in Sec. 2.3 while Maxprop the higher for changing message copies, node density, and TTL. For changing buffer size, Epidemic and Rapid shows high latency.

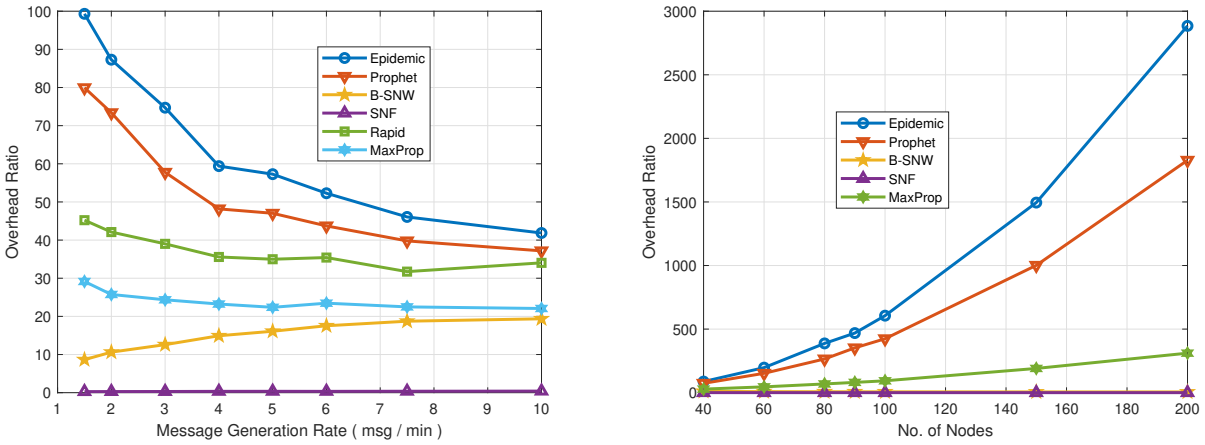
5.2.5. Overhead with Message Copies and Node Density. As the above discussion we know that Epidemic uses the concept of widespread outbreak of something, i.e. message copies which causes the chance of congested network while SNF follow the most technical way to limit the replication of copies. Hence, for this reason, we see higher overhead for Epidemic and lower for SNF in both cases i.e., for changing message copies and number of devices as shown in Figs. 5.8 (a) and 5.8 (b). Rapid has not been simulated for the cause discussed above.

5.2.6. Overhead with Buffer and TTL. From the discussion of Epidemic and SNF in 5.2.5 we see greater overhead for Epidemic and lower for SNF in both cases i.e. for changing buffer and TTL as in Figs. 5.9 (a) and 5.9 (b). In Fig. 5.9 (a), we see Prophet has higher overhead than Epidemic since it stores the record of encountering a node in buffer to forward a copy to the next node.



(a) Average latency with changing buffer size. (b) Average latency with changing TTL.

FIG. 5.7. *Average latency with changing buffer and TTL.*



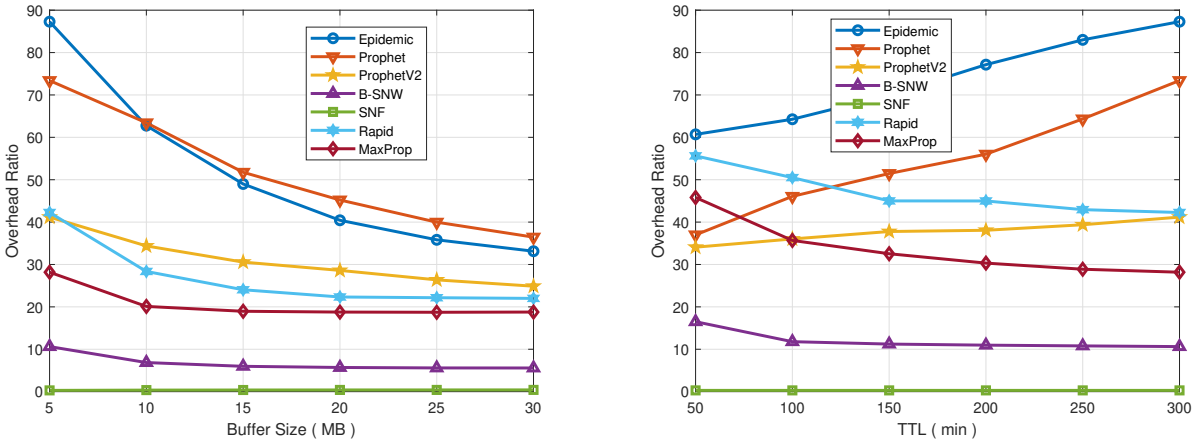
(a) Overhead ratio with changing message copies. (b) Overhead ratio with varying mobile nodes.

FIG. 5.8. *Overhead ratio with changing message copies and density of devices.*

Therefore, from the investigation of Sec. 5.2.5 and 5.2.6, it is clear that SNF routing shows lower overhead ratio and Epidemic the higher compared to other routing protocols mentioned in Sec. 2.3 except for the case of varying buffer where Prophet shows higher overhead.

6. Conclusion and Future Works. In delay tolerant network, devices need to be intermittent as they use the store-and-forward model to deliver a packet successfully to the desired device. This intermittent ad hoc network is featured by topology partitions, long delays, etc. In this research paper, at first we see the impact of three mobility models namely random walk, random direction, and shortest path map based movement mobility only for node density. Then, considering the impact of node density, we evaluate the performance of several DTN routing protocols namely Epidemic, Prophet, Prophetv2, Maxprop, Rapid, Binary spray-and-wait, and Spray-and-focus in an intermittently connected mobile network scenario against changing message copies, density of nodes, buffer and time-to-live using ONE simulator in terms of three performance measurements namely delay, delivery, and overhead. The investigated results demonstrate that spray-and-focus routing exhibits the best performance while epidemic the poor under the consideration of all the metrics.

In near future, we would like to extend this work by using opportunistic mobility models to evaluate the



(a) Overhead ratio with changing buffer. (b) Overhead ratio with varying TTL.

FIG. 5.9. *Overhead with changing buffer and TTL.*

social aware routing strategies.

Authors' Contribution. Md. Sharif Hossen performed the primary literature review, data collection, experiments, and also drafted the manuscript. Muhammad Sajjadur Rahim has conducted overall supervision of the research works and also suggested modifications to improve the quality of the manuscript. Both authors have read and approved the final manuscript.

Acknowledgements. This research was partially supported by the Government of the Peoples' Republic of Bangladesh. We would like to thank Bangladesh government for providing us funding on ICT-2015 research fellowship.

REFERENCES

[1] X. CHEN, C. SHANG, B. WONG, W. LI AND S. OH, *Efficient Multicast Algorithms in Opportunistic Mobile Social Networks using Community and Social Features*, Els. J. of Computer Networks, (2016), pp. 1389–1286.
[2] R. I. CIOBANU, R. C. MARIN, C. DOBRE, V. CRISTEA AND C. X. MAVROMOUSTAKIS, *ONSIDE: Socially-aware and Interest-based Dissemination in Opportunistic Networks*, IEEE Net. Operations and Management Sym. (NOMS), Krakow, Poland, (2014), pp. 1–6.
[3] J. FAN, J. CHEN, Y. DU, W. GAO, J. WU AND Y. SUN, *Geocommunity-Based Broadcasting for Data Dissemination in Mobile Social Networks*, IEEE Tran. Parallel Distributed Syst., 24 (4), (2013), pp. 734–743.
[4] F. D. RANGO F.D., A. SOCIEVOLE AND S. MARANO, *Exploiting Online and Offline Activity-Based Metrics for Opportunistic Forwarding*, Springer J. of Wireless Networks, 21 (4), (2015), pp. 1163–1179.
[5] A. SOCIEVOLE, E. YONEKI, F. D. RANGO AND J. CROWCROFT, *ML-SOR: Message Routing using Multi-layer Social Networks in Opportunistic Communications*, Els. Jnl. of Comp. Net., 81 (22), (2015), pp. 201–219.
[6] J. WU AND Y. WANG, *Opportunistic Mobile Social Networks*, Taylor & Francis CRC Press (2014).
[7] M. XIAO, J. WU AND L. HUANG, *Community-aware Opportunistic Routing in Mobile Social Networks*, IEEE Trans. on Computers, 63 (7), (2014), pp. 1682–1695.
[8] H. ZHOU, J. CHEN, J. FAN, Y. DU AND S. K. DAS, *ConSub: Incentive-based Content Subscribing in Selsh Opportunistic Mobile Networks*, IEEE Journal on Selected Areas in Communication, (2013), pp. 669–679.
[9] K. ZHU, W. LI, X. FU AND L. ZHANG, *Data Routing Strategies in Opportunistic Mobile Social Networks: Taxonomy and Open Challenges*, Computer Networks: The Int. Jnl. of Com. and Tel. Net., (2015), pp. 183–198.
[10] K. FALL, *A Delay-Tolerant Network Architecture for Challenged Internets*, In Proc. of ACM SIGCOMM, Germany, (2003), pp. 27–34.
[11] T. SPYROPOULOS, K. PSOUNIS AND C. X. RAGHAVENDRA, *Single-Copy Routing in Intermittently Connected Mobile Networks*, In Proc. of IEEE SAHCN, Santa Clara, CA, USA, (2005), pp. 235–244.
[12] G. E. PRESCOTT, S. A. SMIT AND K. MOE, *Real-time Information System Technology Challenges for NASAs Earth Science Enterprise* In Proc. of IEEE RTSS, Arizona, (1999).

[13] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh and D. Rubenstein, *Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet* In Proc. of ACM ASPLOS, San Jose, California, USA, 30 (5), (2002), pp. 96–107.
[14] J. Ott and D. Kutscher, *A Disconnection-Tolerant Transport for Drivethru Internet Environments*, In Proc. of IEEE INFOCOM Annual Joint Conf. of the IEEE Com. and Comm. Societies. Miami, FL, USA, (2005), pp. 1849–1862.
[15] T. Spyropoulos, K. Psounis and C. S. Raghavendra, *Efficient Routing in Intermittently Connected Mobile Networks: the Single-Copy Case*, IEEE/ACM Trans. on Net., 16 (1), (2008), pp. 63–76.
[16] K. Zhu, W. Li and X. Fu, *SMART: A Social- and Mobile-Aware Routing Strategy for Disruption-Tolerant Networks*, IEEE Trans. on Veh. Tech., 63 (7), (2014), pp. 3423–3434.
[17] M. Liu, Y. Yang and Z. Qin, *A Survey of Routing Protocols and Simulations in Delay-Tolerant Networks*, In Proc. of WASA, Springer Berlin, Heidelberg, vol. 6843, (2011), pp. 243–253 .
[18] M. J. F. Alenazi, Y. Cheng, D. Zhang and J. P. G. Sterbenz, *Epidemic Routing Protocol Implementation in ns-3*, In Proc. of ACM WNS3, Barceloan, (2015), Spain, pp. 83–90.
[19] W. Moreira and P. Mendes, *Social-Aware Forwarding in Opportunistic Wireless Networks: Content Awareness or Obliviousness?*, In Proc. of IEEE WoWMoM, Sydney, NSW, Australia, (2014).
[20] R. I. Ciobanu , C. Dobre and V. Cristea, *SPRINT: Social Prediction-Based Opportunistic Routing*, In Proc. of IEEE WoWMoM, Madrid, Spain, (2013).
[21] L. Pelusi, A. Passarella and M. Conti, *Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks*, In Proc. of IEEE Comm. Magazine, 44 (11), (2006), pp. 134–141.
[22] S. Misra, B. K. Saha and S. Pal, *Opportunistic Mobile Networks: Advances and Applications*, Chapter 2, Springer International Publishing, Computer Communications and Networks, (2016).
[23] Z. Zhang, *Routing in Intermittently Connected Mobile Ad-hoc Networks and Delay Tolerant Networks: Overview and Challenges*, IEEE Com. Sur. & Tut., vol. 8, (2006).
[24] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass and J. Scott, *Impact of Human Mobility on Opportunistic Forwarding Algorithms* IEEE Tran. on Mob. Com., vol. 6, (2007).
[25] J. Miaoa, O. Hasana, S. B. Mokhtara, L. Bruniea and G. Gianini, *A delay and cost balancing protocol for message routing in mobile delay tolerant networks*, Els. J. of Ad Hoc Net., (2015), pp. 430–443.
[26] S. Jain, K. Fall and R. Patra, *Routing in a Delay-Tolerant Network*, In Proc. of ACM SIGCOMM, Portland, (2004), pp. 145–158.
[27] M. Grossglauser and D. Tse D., *Mobility increases the capacity of ad hoc wireless networks*, IEEE/ACM Transaction on Networking, USA, (2002), pp. 477–486.
[28] A. Doria, M. Uden and D. P. Pandey, *Providing Connectivity to the Saami Nomadic Community*, In Proc. of ICOCDSI, Bangalore, India (2002).
[29] L. Pelusi, A. Passarella and M. Conti, *Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad-hoc Networks*, IEEE Comm. Mag., 44 (11), (2006), pp. 134–141.
[30] D. Henriksson, T. F. Abdelzaher and R. K. Ganti, *A Caching-Based Approach to Routing in Delay-Tolerant Networks* In Proc. of ICCCN, Honolulu, HI, USA, (2007), pp. 69–74.
[31] A. Keränen, J. Ott and T. Kärkkäinen, *The ONE Simulator for DTN Protocol Evaluation*, In Proc. of ICSTT, Rome, Italy, (2009).
[32] A. Vahdat and D. Becker, *Epidemic Routing for Partially Connected Ad-hoc Networks*, Dept. of Com. Sci., Duke University, Tech. Rep., (2000).
[33] A. Lindgren, A. Doria and O. Scheln, *Probabilistic Routing in Intermittently Connected Networks*, ACM SIGMOBILE Mobile Compu. and Communs. Rev., 7 (3), (2003), pp. 19–20.
[34] S. Grasic, E. Davies, A. Lindgren and A. Doria, *The Evolution of a DTN Routing ProtocolPRoPHETv2*, ACM SIGCOM, Las Vegas, Nevada, USA, (2011), pp. 27–30.
[35] A. Balasubramanian, B. N. Levine and A. Venkataramani, *DTN Routing as a Resource Allocation Problem*, In Proc. of ACM SIGCOMM, Kyoto, Japan, (2007), pp. 373–384.
[36] J. Burgess, B. Gallagher, D. Jensen and B. N. Levine, *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks*, In Proc. of IEEE INFOCOM, Barcelona, Spain, (2006).
[37] T. Spyropoulos, K. Psounis and C. S. Raghavendra, *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*, In Proc. of ACM SIGCOMM WDTN, USA, (2005), pp. 252–259.
[38] T. Spyropoulos, K. Psounis and C. S. Raghavendra, *Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility*, In Proc. of IEEE PerCOMW, White Plains, NY, USA, (2007).
[39] *Project page of the ONE simulator*, https://www.netlab.tkk.fi/tutkimus/dtn/theone/.
[40] A. Kernen, T. Krkkinen and J. Ott,*Simulating Mobility and DTNs with the ONE*, J. of Comm., 5(2), (2010), pp. 92–105.
[41] L. Song and D. F. Kotz, *Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces*, In Proc. of ACM CHANTS, Montreal, Quebec, Canada, (2007), pp. 35–42.
[42] L. You, J. Li, C. Wei, C. Dai, J. Xu and L. Hu, *A Hop Count Based Heuristic Routing Protocol for Mobile Delay Tolerant Network*, The Scientific World Journal, (2014).
[43] H. Theus, T. Spyropoulos and F. Legendre, *Putting Contacts into Context: Mobility Modeling beyond Inter-Contact Times*, ACM Int. Sym. Mob. Ad Hoc Net. and Com., Paris, France, (2011), pp. 1–8.
[44] E. M. Royer, P. M. Melliar-Smith and L. E. Moser, *An Analysis of the Optimum Node Density for Ad hoc Mobile Networks*, IEEE ICC, Helsinki, Finland, vol. 3, (2001), pp. 857–861.
[45] Z. J. Haas, *A New Routing Protocol for the Reconfigurable Wireless Networks*, In Proc. of IEEE ICUPC, San Diego, CA, USA, vol. 2, (1997), pp. 562–566.

[46] C. Bettstetter, *Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects*, In Proc. of ACM SIGMOBILE Mob. Compu. and Comms. Rev., 5 (3), (2001), pp. 55–66.
[47] A. Kernen and J. Ött, *Increasing Reality for DTN Protocol Simulations. Technical Report*, Networking Laboratory, Helsinki University of Technology, (2007).

Edited by: Khaleel Ahmad
Received: Nov 6, 2018
Accepted: Feb 20, 2019

DOI 10.12694/scpe.v20i1.1479
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 27–40. http://www.scpe.org © 2019 SCPE

ESTABLISHING RELIABILITY FOR EFFICIENT ROUTING IN OPPORTUNISTIC NETWORKS

DEEPAK KUMAR SHARMA* AND DEEPIKA KUKREJA†

Abstract. Opportunistic network (Oppnet) is a class of networks where connections between the nodes are not permanent. The nodes are continuously moving and some nodes even switch off their batteries to conserve energy. Reliable delivery of messages in Opportunistic network is one major inherent issue. It is unreliable in the sense that once the source node has forwarded its message, then it will never get to know about its status in the network like whether the message has got discarded at an intermediate node or at the destination node (due to buffer overflow) or the successful delivery of the message has taken place. This work tries to make Oppnet as much reliable as possible. It proposes a reliability protocol named as Reliability in Oppnet (RIO). RIO improves the routing in Oppnet and works in parallel with the existing routing protocols. It makes the source node aware about the status of message so that if an error occurs then the source node can take suitable action to resend the message. It considers the redirection error, buffer overflow error, Time Limit Exceeded (TLE), parameter problem and destination unreachable errors that may occur inside the network. RIO has been tested using ONE simulator and implemented with Spray and Wait routing protocol. Results show that the RIO with Spray and Wait protocol outperforms normal Spray and Wait protocol in terms of average message delivery probability.

Key words: Routing Protocol, Reliability, Opportunistic networks, Average message delivery, Network errors, ONE simulator

AMS subject classifications. 68M12

1. Introduction.

Opportunistic network [1] as the name signifies, is the type of network that is based on the opportunities of contacts that exist in the network. The connections between the nodes are created and terminated on demand i.e. based on the availability of suitable nearby node that can take the message in the vicinity of destination node. The nodes try to utilize the best opportunity to connect to a node that will take the message in close proximity of intended destination node. Opportunistic networks inherit its characteristic features from two super classes that are Delay tolerant networks (DTN) [2, 3] and Mobile Ad hoc Networks (MANETs) [4, 5, 6, 7]. Oppnet inherits the property of discontinuous network connections from DTN. The connections between the nodes in Oppnets are not permanent like TCP/IP model, so these networks are prone to long unpredictable delay in packet transmission [8]. Oppnet acquires the property of nodes from MANETs as the nodes are mobile in these types of networks. Hence it can be said that Oppnet is a kind of network that is somewhat like a mixture of MANET and DTN i.e. is a network with no continuous node connections with nodes being mobile.

Routing in Opportunistic network is a challenging task as nodes connections are not permanent. A message can be delivered instantly (if the connections that leads to successful delivery of message are available at that time) or the message can take hours or even days before it gets successfully delivered to the intended destination. Traditional routing protocol does not solve the problem of routing of message in Oppnet. Routing protocols in Oppnet are broadly classified in two categories; these are infrastructure-based routing protocols and infrastructure less routing protocols. Infrastructure based routing protocols use infrastructure in some form like message ferries, info stations and cloud computing. It is obvious that these types of routing protocols require expensive technologies and equipment. HVF Scheme [9], Message ferry scheme [10], Cloud Computing based routing protocol (CCBRP) [11] and Infostation Model [12], are some examples of infrastructure base routing protocols. Infrastructure less routing protocols are those in which no infrastructure is required to support routing in Oppnet. In these type of routing protocols congestion control, node buffer storage, node battery [13, 14], and traffic in the network are main issues of concern in routing. Many protocols have been developed to tackle these problems for example epidemic routing protocol [15], PROPHET [16] routing, Spray and Wait routing protocol [17] to name a few. Protocols that handles security issues [18, 19], machine learning techniques [20, 21], and other miscellaneous routing related works [22, 23, 24] has also been designed for Oppnets recently.

*Division of Information Technology, Netaji Subhas University of Technology (Formerly Netaji Subhas Institute of Technology), New Delhi, India (dk.sharma1982@yahoo.com), corresponding author.

†Division of Information Technology, Netaji Subhas University of Technology (Formerly Netaji Subhas Institute of Technology), New Delhi, India (deepikakukreja18@gmail.com)

All the previously designed protocols are unreliable in the sense that once the source node sends message in the Oppnet, it assumes that the successful transfer will take place to the destination. But this is not always true in many cases. Consider a scenario in which the forwarded message gets dropped at the intermediate node as it was unable to store message in its buffer due to full buffer capacity (buffer overflow). Consider one another scenario where the forwarded message reaches some intermediate node and this intermediate node finds that Time to Live (TTL) has been expired, and then this node simply drops the message. Message can even be discarded at destination node. For example, consider a situation where a destination node cannot accept the message due to full buffer capacity at that time. In order to resolve unreliabilities like these, a reliability protocol named as RIO has been proposed. The proposed protocol works in parallel with the existing Oppnet routing protocols and hence enhance routing in Oppnet.

The errors that may occur in the Oppnet are classified into five categories: redirection error, buffer overflow error, parameter problem, Time Limit Exceeded i.e. TTL expiration error and destination unreachable. In this work, solutions are provided for the above-mentioned errors, and through simulation using Opportunistic Network Environment simulator i.e. ONE simulator [25] it has been found that RIO enhances routing of existing protocols while working in parallel with them. For testing purpose, RIO is combined with Spray and Wait protocol. Through simulations it has been proved that Spray and Wait routing gets enhanced when reliability protocol RIO works in parallel with it.

The paper is organized as follows. Section 2 discusses the related work done in past. Section 3 explains the proposed work and demonstrates the working of the RIO protocol in detail. Simulation results and observations are given in section 4. Finally, Section 5 concludes this work and talk about future work.

2. Background and Related Work. This section is dedicated to explain briefly some existing infrastructure less routing protocols with their merits and demerits.

PROPHET PROPHET [16] protocol is a probability-based routing where nodes forward messages to other nodes based on a probabilistic metric. This metric is estimated based on history of encounters. Whenever nodes interact with each other to exchange messages, they also exchange node encounter history with each other. In this way, this protocol is transitive in nature i.e. if N1 and N2 nodes share information with each other and N2 node and N3 node share information with each other, then it is equivalent to sharing information between N1 node and N3 node. In this protocol, a node forwards the message to that neighboring node that has highest probability metric.

EPIDEMIC Epidemic routing [8] is based on the flooding technique. Every node in the network floods the message to its neighboring nodes which then flood this message to their neighbors. Each node has two buffers, one for its own messages and second for storing the messages received from other nodes. Every message in the system is flagged with a unique ID. On meeting, two nodes swap their summary vectors and exchange those messages which are not present in their buffers. This procedure is followed at each pair of nodes and eventually every node in the network gets the message copy. This Protocol creates lot of message copies which make it robust against network failure and result in lesser delay, but at the same time creates network congestion.

HBPR HBPR [26] stands for History Based Routing Protocol and is based on behavior and stability of nodes. This protocol tries to deliver message using a path which is most visited by the destination node. In this way the intermediate node selects those nodes as next hop that tries to deliver message to the path that is most visited by the destination and thus harness the behavior of nodes in the network. HBPR has high average message delivery probability as compared to tradition Oppnet protocol like epidemic. The average latency and overhead ratio are also less in HBPR as compared to epidemic routing protocol.

EDR EDR [27] is an Encounter and Distance based protocol that forwards the message based on a metric known as forward parameter. Forward parameter is calculated based on encounter history of nodes with destination node and on the distance of nodes from destination node. This metric is calculated for each neighboring node and the message is given to that node which has the metric value above a threshold.

Spray and Wait Spray and Wait protocol [17] is a controlled flooding-based routing protocol. It limits the number of messages forwarded in the network unlike epidemic routing protocol where the nodes forward message to every node it encounters. This routing protocol works in two phases the first phase is the

Spray phase and the other is Wait phase. In the Spray phase, the source node forwards the message to L nodes. In the Wait phase, if the message is not yet delivered to destination node then it performs a direct delivery of message. Spray and Wait reduces the traffic and the number of messages dropped in the network as compared to epidemic routing protocol and the delivery probability using Spray and Wait protocol is also high due to controlled flooding.

GAER GAER [28] is Genetic Algorithm based Efficient Routing protocol. This is an energy efficient protocol as the energy spent in routing of the messages is lowest as compared to other available routing protocols. This protocol uses personal information of nodes and applies genetic operations and algorithm which are used to decide the next hop. It calculates a fitness function based on some parameters and then forwards the message to a node whose fitness value is above the cut off threshold value.

Spray and Focus Spray and Focus [29] is mobility assisted routing protocol. It has two phases; one is the Spray phase and the other is Focus phase. A node forwards message to L relay nodes in Spray phase, and in Focus phase the message can be forwarded to different relay nodes based on some forwarding criterion. In Focus phase, the transmission is not direct unlike Spray and Wait protocol. Spray and Focus is much more efficient than available mobility assisted routing protocol and it outperforms them in terms of average message delivery probability and average latency.

3. Proposed Work. In Oppnets, nodes contact with each other opportunistically that is no stable or permanent path exist between the sender and receiver node. Hence, Oppnets are highly unreliable in terms of message delivery. Once the source node forwards message, it will never be able to know whether its message successfully reaches to the destination or not. The message might get lost in the network or it might be discarded by some intermediate node due to the following reasons.

- Intermediate nodes buffer overflow that is, there might be a case where the buffer queue of the intermediate node might be full.
- Ambiguity in the addresses that are incorporated in the header of the message.
- Their might be a case where the Time to Live (TTL) of the message might become zero when it reaches an intermediate node.

There can be numerous cases of message not getting successfully delivered to the intended destination. So, now the question arises that how to cope up with this unreliability that exist in the network? How to devise a way so that the sender of the message must be able to track the status of its forwarded messages and if there occurs some error in delivery of messages, then the sender node can act accordingly.

The solution to these problems is to ensure reliability in routing of messages in Opportunistic networks. Reliability can be ensured through error reporting method. Whenever an error occurs during routing of messages, an observing node (which has noticed the error) sends an error reporting message to the source node. After receiving these error reporting messages, the source node can enquire for the cause of failure and can resends the respective message or takes some other suitable action.

3.1. Routing Errors. Errors in routing can be broadly specified into following five types.

3.1.1. Redirection Error. There might be a case where the message forwarded by the source node travels in wrong direction and it gets continuously forwarded in the same wrong direction by the intermediate nodes like in case of MoVe(Motion Vector)[30] routing protocol which forwards the message to its nearby node that is closest to the node. In MoVe protocol, there might occur a case where the nodes continuously forward the message in the wrong direction due to their characteristic property of forwarding message to the nodes that have shortest distance to it. In First Contact routing protocol nodes forward messages randomly based on the available contacts at a particular instant of time. In this protocol their might occur a case where the nodes continuously forward the message in the wrong direction because the node that is forwarding the message always finds a connection to a node that is in the wrong direction. To handle such situations in message routing and to make routing reliable, the intermediate node that notices such wrong redirection of messages sends a redirection error message to the source node and simultaneously tries to forward the message in the right direction which is towards the intended destination. Figure 3.1 shows a scenario of redirection error. In the figure, the source node forwards the message in wrong direction and the intermediate nodes also forward the message in the same wrong direction. After passing through several intermediate nodes, one of the intermediate nodes notices this

redirection error and tries to forward the message in the direction of destination node and reports the same to the source node.

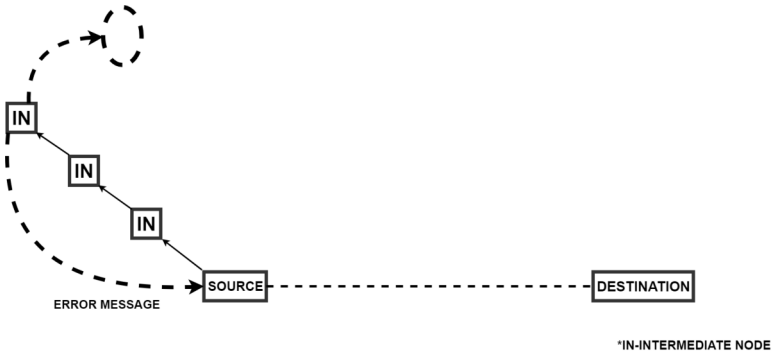


FIG. 3.1. *Redirection Error Reporting*

3.1.2. Buffer overflow. While routing the messages from source node to destination node, there is very high probability of encountering an intermediate node whose buffer is full. In this case, the intermediate node drops or discards the message without bothering about it much. Epidemic routing protocol follows flooding-based approach for message passing which means that a node forwards multiple copies of messages in the network. In this, the buffer occupancy time of messages is very high which leads to high chances of getting a relay node in routing path which has full buffer or overflowed buffer. This same reasoning follows for Spray and Wait, PROPHET and MaxProp [31] routing protocols. In order to deal with such situations, using RIO, the node that notices such a buffer overflow problem sends a buffer overflow error message to the source node which has generated the message. This error message is sent to the source node before the source node discards the message. Figure 3.2 depicts a scenario of buffer overflow problem. In this, the source nodes forward a message which passes through numerous intermediate nodes and finally arrives at destination node, after arriving at destination node, the message gets discarded because the destination node notices that its buffer is full and there is no room to store the incoming message, now the destination node sends an error message to the source node.

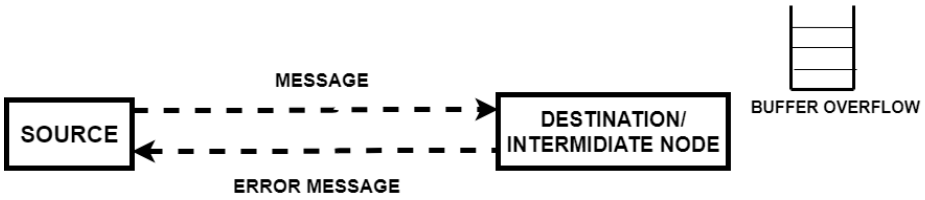


FIG. 3.2. *Buffer Overflow Error Reporting*

3.1.3. Time Limit Exceeded. There might occur a case while routing of the message from source to destination, that the Time to Live (TTL) of the message has got expired that is the value of TTL becomes zero. At that time, the receiving node of the message simply discards the message and the source node of the message never be able to know that whether the message gets successfully delivered or not. RIO protocol is implemented to deal with such situations, using RIO, the node that notices such phenomenon sends a TLE message to the source node. The TLE message is sent to the source node before the source node discards that message, so that it can resend the message in order to get its message successfully delivered to its intended destination. Figure 3.3 shows a scenario of Time Limit Exceeded. In the figure, the source node forwards a message and the message passes through numerous nodes and finally its TTL expires. Now the node at which TTL expires sends an error message to the source node.

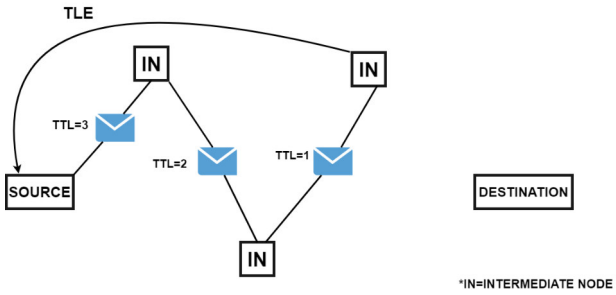


FIG. 3.3. *TLE Error Reporting*

3.1.4. Parameter Problem. Their might occur a case where the receiving node of a particular message notices that the fields in the header of the message are ambiguous. For example, the receiving node might notice that the source address or the destination address is ambiguous. It might happen that the destination address is set to all zeros or all ones. So, to handle situations like these, using RIO protocol, the receiver of the message sends a parameter problem error message to the source node before the source node discards it. The source node resends the message after rectifying. Figure 3.4 depicts a scenario of parameter problem. In the figure, the source node forwards the message and after passing through several nodes, one of the node notices that the destination field in the header of the message is ambiguous. It notifies the source node by sending a parameter problem error message to it.

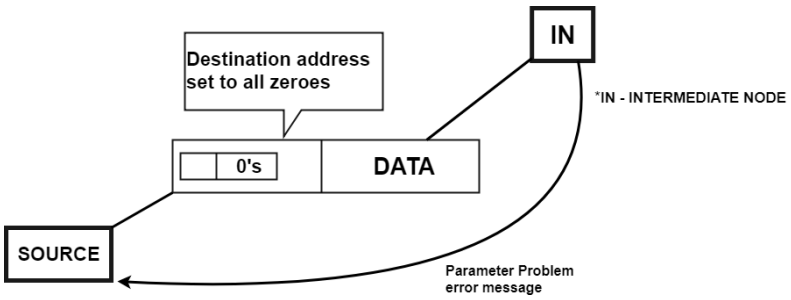


FIG. 3.4. *Parameter Problem Error Reporting*

3.1.5. Destination Unreachable. In Oppnets, nodes sometimes switch off or turn off their batteries in order to conserve the energy. So, in Oppnet there might occur a case where the destination node power is not on and the message that was intended for it might get lost in the network because the node for which it was destined was invisible to the network. To cope up with situations like this, using RIO protocol, the node that is currently holding the message defers the routing of the message until the node for which it is destined turns on its power. If the destination node doesnt turn on its power until a certain threshold value of time duration then the node (i.e. currently holding the message) sends a destination unreachable error message to the source node. Figure 3.5 depicts a destination unreachable scenario where the source node forwards the message and this message passes through nodes IN-1, IN-2 and finally IN-3. IN-3 node notices that the destination is not in the network (i.e. its battery is off) so it defers sending the message until some threshold time (assuming twice of remaining TTL). If until then, the destination node does not switch on its power then a destination unreachable error message is sent to the source node.

3.2. Working of RIO. RIO is a dependent reliability protocol. It improves the performance of an existing routing protocol, if reliability features that are established in RIO are allowed to work in parallel with other existing infrastructure less routing protocol. In RIO, every node first checks if any incoming message is an error message. If an incoming message is an error message then it matches the destination address (i.e. embedded in the error message) of error message with its own address. If match occurs then it suspects whether the error

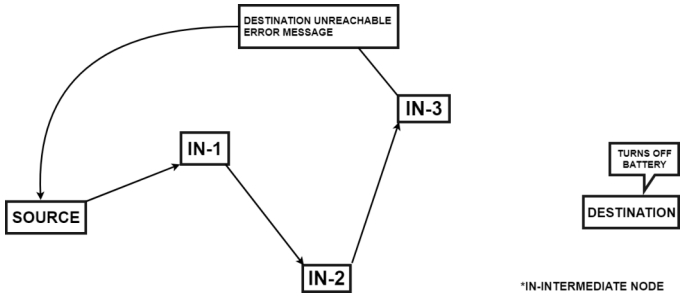


FIG. 3.5. *Destination Unreachable*

message is redirection error message or not. If it is not redirection error message then it resends the message or take some suitable action. If the address does not match then it routes this error message in the Oppnet because this error message is not intended for this node. If the incoming message is not an error message then it checks for the errors that were specified in the previous section i.e. the node checks for redirection error, buffer overflow error, Time Limit Exceeded, Parameter problem and destination unreachable error.

In the following paragraphs, strategies are discussed to deal with these errors. All these errors are checked at a node when the node receives any incoming message. For example, a node when receives a message checks whether its buffer is full or not, whether TTL of message is greater than zero or not, likewise it checks for other errors.

The redirection error is checked by calculating the angle between the two imaginary lines that are formed by joining the coordinates of the sender node to the coordinates of the current node and the coordinates of the sender node to the coordinates of the destination node. The first line is between the coordinates of the sender of the message and the current node. The second line is between the coordinates of destination node of the message and the sender node of the message (i.e. the node from which the current node has received the message). If these two lines are making an angle greater than ninety degrees then there is a redirection error and the current node tries all connections which result in angle less than ninety degrees between the above two specified lines and simultaneously sends a redirection error message to the source of this message. The two possible scenarios have been shown in figure 3.6 and figure 3.7. In figure 3.6 i.e. scenario 1, the intermediate node notices that the angle formed between the two lines that are specified above is greater than ninety degrees, so it sends a redirection error message to the source node. In figure 3.7 i.e. scenario 2 the angle formed between the lines specified above is less than ninety degrees, so it continues with normal routing. For finding the coordinates of the nodes, Global positioning system (GPS) is used, which is now-a-days pre-installed on every mobile and laptops. GPS does not require any other technologies like internet to operate, it operates independently. Hence this technology is handy to find out the coordinates of the nodes.

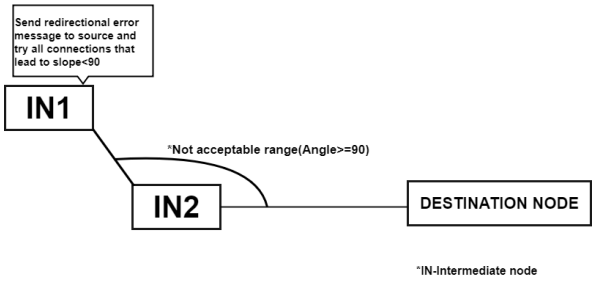


FIG. 3.6. *Redirection error (scenario 1)*

The buffer overflow problem at a node is detected by checking the current free buffer space. If the free buffer space is less than or equal to zero then the buffer overflow error reporting message is sent to the source node of incoming message before this node discards the message.

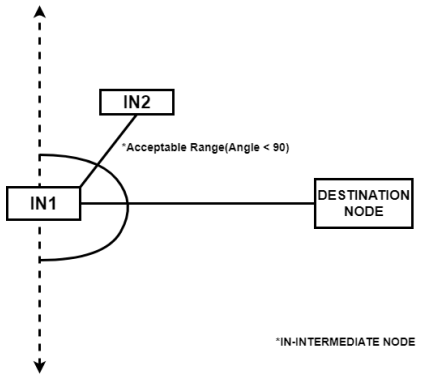


FIG. 3.7. *Redirection error (scenario 2)*

The TLE error at a node is detected by checking the values of TTL for all the incoming messages, if the TTL is found to be zero then the TLE error reporting message is sent to the source node.

The parameter problem error is detected by checking the validity of destination address specified in the incoming message. If the destination address that is embedded in the incoming message is found to be invalid then a parameter problem error message is sent to source of incoming message before discarding it.

The Destination unreachable error is detected by checking the visibility of destination node address through GPS. The node holds the message until the destination node gets visible and keeps the message in its buffer until some threshold is reached. If the destination node is not visible until a certain threshold (twice of remaining TTL) then a destination unreachable error message is sent to the source node.

4. Simulation Results. Opportunistic Network Environment (ONE) [25] simulator has been used to evaluate the performance of RIO protocol. ONE is an open source Oppnet simulator. It is built with java programming language. It comes with several built-in modules that facilitate routing in Oppnet. ONE generates reports based on the settings that have been made to run the simulation. There are infinite number of different scenarios that can be run in ONE simulator. This characteristic makes it one of the most popular simulators for simulating Opportunistic network environment protocols.

4.1. Simulation Setup. Table 4.1 shows the simulator parameters that have been chosen for the set up to simulate RIO protocol. The pseudo code that has been presented in the above section has been implemented in update method of the routing class of ONE simulator. To test the performance of RIO with existing routing protocol, Spray and Wait routing protocol is used with RIO. RIO has been simulated by varying following metrics while keeping the other parameters constant.

1. Variation in the nodes: The nodes in the network are changed from 40 to 240 by increasing 40 nodes at each step and keeping all others parameters constant.
2. Variation in TTL of nodes: The message TTL has been changed to different values from 100 to 300 with an increment of 50 while keeping all others parameters constant.
3. Variation in message generation interval: This interval has been varied (while keeping other parameters constant) that changes the messages generated in the network, for example if this interval is 0 to 5 then the message will be generated at each node at a timestamp that ranges from 0 to 5.

The following metrics are used to evaluate the performance of RIO:

1. Average Message Delivery probability: It is the probability of successfully delivered messages.
2. Average Hop Count: It denotes the average number of hops or nodes that a message travels to reach its intended destination.
3. Average Message Delay: It denotes the average time taken from message creation t its delivery.

4.2. Results and Observations. In this section, the results and observations after performing the simulation are presented.

TABLE 4.1
Simulation Parameters

Parameter	Simulation Value
Area of simulation	4500 x 3400 sq. m
Transmission Rate	250 Kbps
Groups of nodes	6
Transmission range	10 m
Movement model	Shortest Path Map Based Movement
Each nodes Storage Space	5 Mb
Simulation time	43200 s
Speed Range	1-14m/s
Range of Wait Time	0-120 s
Message size	500 Kb to 1Mb
Generation time of Messages	25-35 s

4.2.1. Average Message Delivery Probability. This section compares the results of average message delivery probability for Spray and Wait (SAW) protocol and SAW with RIO protocol by varying the parameters like number of network nodes, TTL and message generation interval. Figure 4.1 depicts the average message

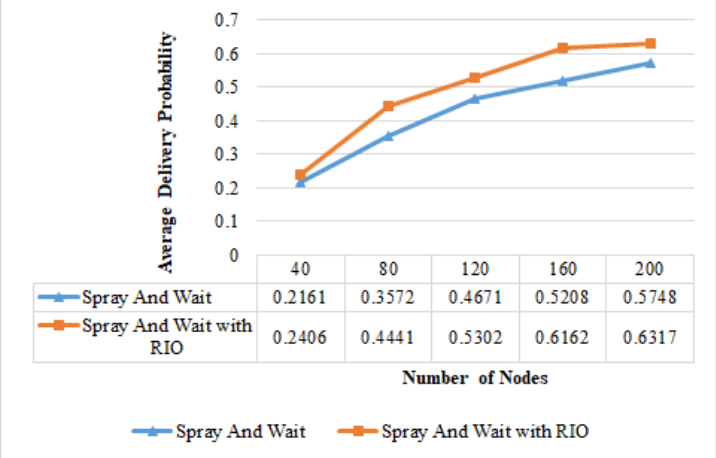


FIG. 4.1. *Average Message Delivery Probability v/s Number of Nodes*

delivery probability v/s the number of nodes. As the number of nodes increases the average message delivery probability increases in Spray and Wait because the network becomes dense and messages easily find nodes that can deliver them to destination node. The probability is further increased in Spray and Wait with RIO because now some of the messages which were not delivered to destination in Spray and Wait are retransmitted by the source node after receiving the error message. The cause of failure is specified within the error message. In figure 4.1, the average message delivery probability increases by 15.29% in Spray and Wait with RIO as compared to Spray and Wait. Hence, on an average the delivery probability of messages increases in RIO with Spray and Wait as compared to Spray and Wait. It can be justified by the fact that the messages which previously get discarded by the intermediate nodes (due to various reasons like buffer overflow, TTL expire, parameter problem etc.) now get successfully delivered to their intended destination because of the reliability imposed. If the message gets discarded, an error reporting message is sent to source node which leads to retransmission of message and thus resulting in more number of successfully delivered messages.

Figure 4.2 depicts average message delivery probability v/s TTL of the messages. It has been observed that RIO with Spray and Wait has better message delivery probability as compared to Spray and Wait. As the TTL

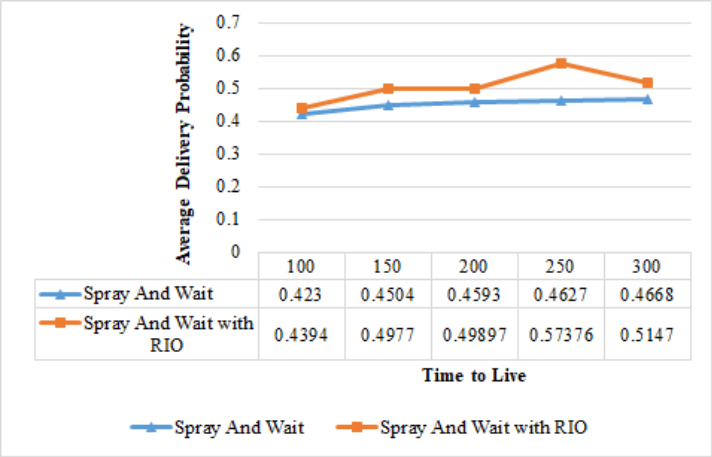


FIG. 4.2. *Average Message Delivery Probability v/s TTL of the messages*

increases, the time that a message can search for their respective destination increases. If some messages get discarded or by some means do not reach to their destination, now have more probability for successful delivery because of the error reporting mechanism of RIO. Using RIO, the average message delivery probability of Spray and Wait protocol has increased by 11.59%.

Figure 4.3 depicts average message delivery probability v/s message generation interval. As this interval

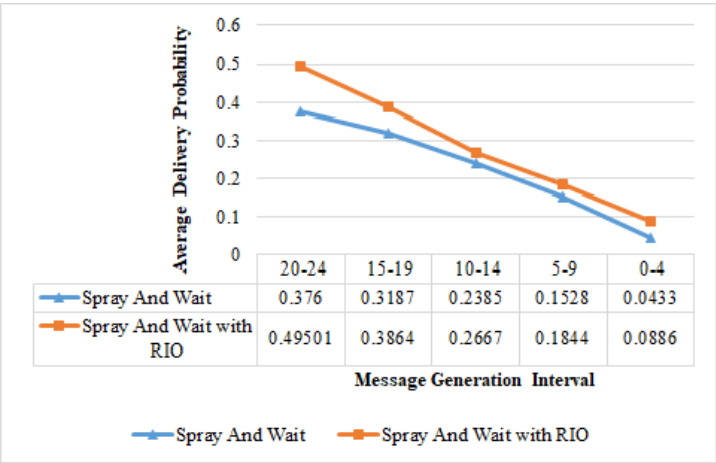


FIG. 4.3. *Average Message Delivery Probability v/s Message Generation Interval*

decreases, more number of messages get generated in the network, leading to increase in network traffic. Spray and Wait protocol is a controlled flooding-based technique which in turns add to the already increased traffic. Hence the number of messages dropped at intermediate nodes increases which results in decrease in average message delivery probability of Spray and Wait routing protocol. The average message delivery probability increases when RIO works alongside Spray and Wait because now some of the messages which were previously gets dropped are delivered to their respective destination because of the added reliability features of RIO. The average message delivery probability of SAW with RIO has increased by 25.83%.

4.2.2. Observing Average Hop Count. Implementing RIO with Spray and Wait, the average hop count increases as now the hop counts of the messages which were previously get discarded by the intermediate nodes now travel again from source node to destination node because of the error reporting mechanism used in RIO.

This increases the number of hops through which the message has been passed in order to reach its destination node. Figure 4.4 shows the average hop count v/s number of network nodes. The average hop count of Spray and Wait using RIO increases by 15.83% because now the discarded messages travel through more nodes as the network is dense, thus this will add to hop counts.

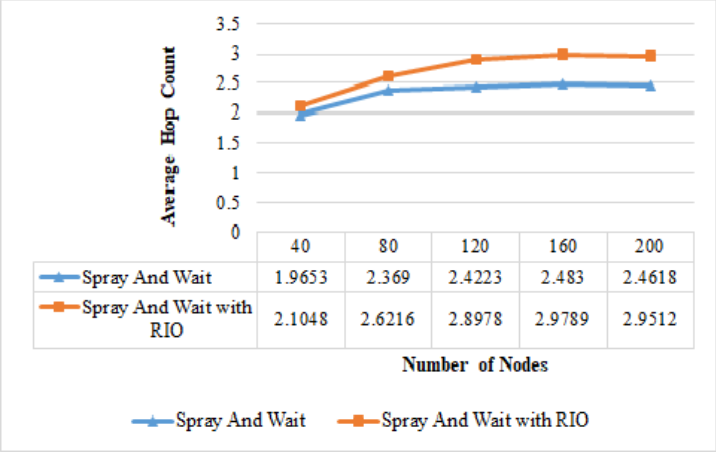


FIG. 4.4. *Average Hop Count v/s Number of Nodes*

Figure 4.5 depicts average hop count v/s Time to Live. In the figure the average hop count increases by 15.71%, which is justified by the fact that as the TTL increases the messages have more time to live in the network and if some messages get discarded by some intermediate nodes then an error message is sent to the source node, which leads to retransmission of the messages and thus increases the average hop count.

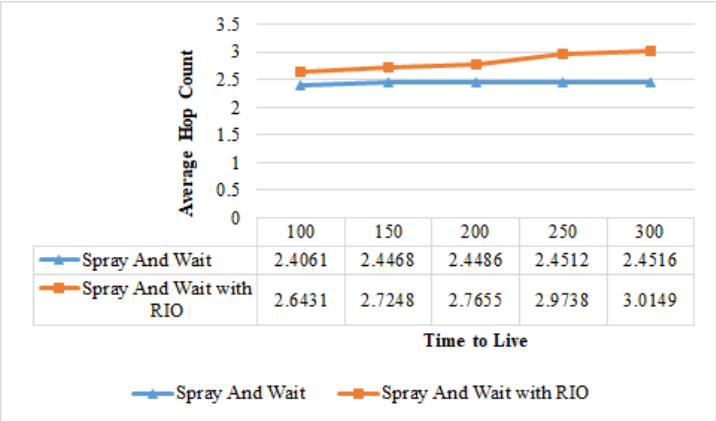


FIG. 4.5. *Average Hop Count v/s Time to Live*

Figure 4.6 shows the average hop count v/s message generation interval. The average hop count of Spray and Wait has increased by 19.45% using RIO which is justified by the fact that as the message generation interval decreases the number of messages generated increases, this increases the number of undelivered messages and some of these undelivered messages are retransmitted because of the reliability features of RIO and thus this further adds to the hop count which is evident from the figure.

4.2.3. Observing Average Latency. In Spray and Wait using RIO, on an average the latency increases as compared to Spray and Wait. This is justified by the fact that now the transmission time of sending the

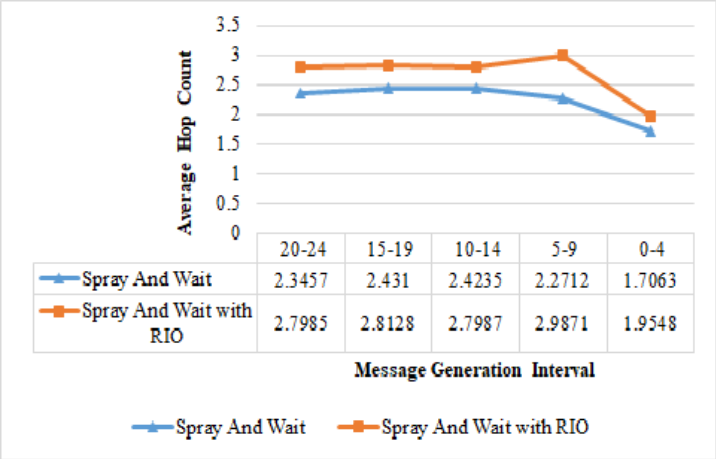


FIG. 4.6. *Average Hop Count v/s Message Generation Interval*

error reporting message back to source node is added to the total latency. And also, the time to retransmit the message (which was previously discarded) from the source to destination node has also been added to latency leading to higher latency as compared to Spray and Wait routing protocol. This is evident from figures 4.7, 4.8 and 4.9. Figure 4.7 depicts that the average latency of Spray and Wait increases by 8.02% with varying number of network nodes. Figure 4.8 shows that the average latency of Spray and Wait with RIO increases by 3% by

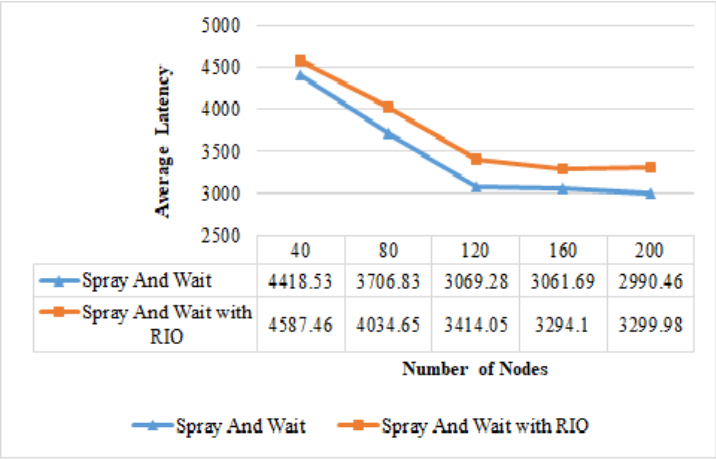


FIG. 4.7. *Average Latency v/s Number of Nodes*

increasing TTL. Figure 4.9 depicts that the average latency of Spray and Wait increases by 11.94% while using RIO v/s message generation interval.

5. Conclusion and Future Work. The work establishes the reliability in Opportunistic Network through the application of error reporting. In this, a novel reliability protocol, RIO has been proposed, implemented and tested. RIO works in parallel with the existing routing protocols and it enhances routing performance because of the added reliability features. The proposed work classifies errors that may happen in the Oppnet into five categories, Redirection error, Time Limit Exceeded (TLE) error, Buffer overflow error, Parameter Problem error and Destination unreachable error. RIO protocol has been implemented along with Spray and Wait routing protocol. It has been observed that RIO causes increase in average message delivery probability, average hop count and average delay of Spray and Wait protocol.

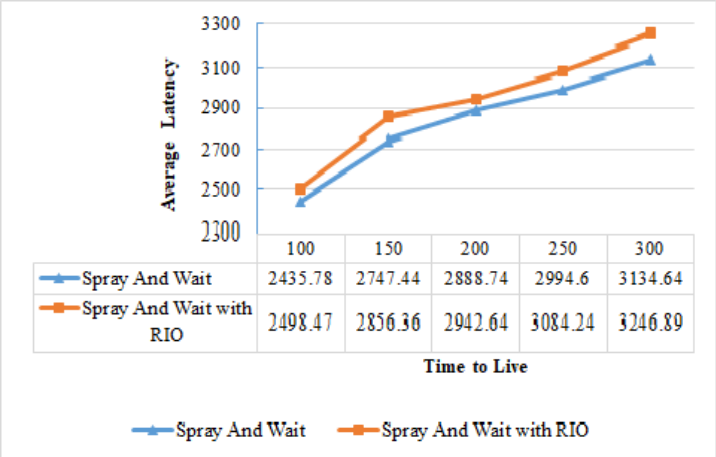


FIG. 4.8. *Average Latency v/s Time to Live*

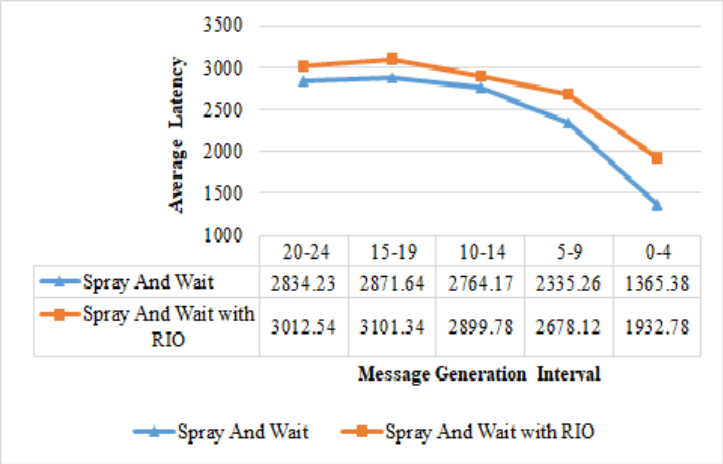


FIG. 4.9. *Average Latency v/s Message Generation Interval*

In future, we will try to establish reliability in Cloud Computing based routing protocol and hence can improve the efficiency of Oppnet so that it can be used for practical purposes.

REFERENCES

[1] L. LILIEN, Z. H. KAMAL, V. BHUSE, A. GUPTA A, *Opportunistic networks: the concept and research challenges in privacy and security*, In: Proceedings Of NSF intl. workshop on research challenges insecurity and privacy for mobile and wireless networks (WSPWN 2006), Miami, March 2006, pp. 134–147.

[2] K. FALL, *A delay-tolerant network architecture for challenged internets*, In: Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, 2529 Aug 2003, pp. 27–34.

[3] S. JAIN, K. FALL, R. PATRA, *Routing in a delay tolerant network*, In: Proceedings of ACMSIGCOMM 2004, pp. 145–158.

[4] C. K. TOH, *Ad hoc mobile wireless networks: protocols and systems*, Prentice Hall PTR, Englewood Cliffs.

[5] D. KUKREJA, S. K. DHURANDHER AND B. V. R. REDDY, *Power aware malicious nodes detection for securing MANETs against packet forwarding misbehavior attack*, Journal of Ambient Intelligence and Humanized Computing, Springer, April 2017, doi:10.1007/s12652-017-0496-2, pp. 1–16.

[6] D. KUKREJA, S. K. DHURANDHER AND B. V. R. REDDY, *Enhancing the security of dynamic source routing protocol using energy aware and distributed trust mechanism in MANETs*, Intelligent Distributed Computing (Springer) Series, Volume 321, 2015, pp. 83–94.

[7] M. MIGLANI, D. KUKREJA, S. K. DHURANDHER, B. V. R. REDDY, *Power aware and secure dynamic source routing protocol*

in mobile ad hoc networks, Security in Computing and Communications, Communications in Computer and Information Science, Volume 467, 2014, pp. 45–56.

[8] S. K. Dhurandher, D. K. Sharma, I. Woungang, and H.C. Chao, *Performance evaluation of various routing protocols in opportunistic networks*, in Proceedings of IEEE GLOBECOM Workshop 2011, Houston, Texas, USA , 5-9 December, 2011, pp. 1067–1071.

[9] S. K. Dhurandher, D. K. Sharma, S. Gupta, I. Woungang, and M. S. Obaidat, *Integration of fixed and mobile infrastructure for message passing in opportunistic networks*, in proceedings of JOURNAL OF NETWORKS, vol. 10, No. 12, Acadmey Publisher, December 2015, pp. 642–657.

[10] W. Zhao, M. Ammar, and E. Zegura, *A message ferrying approach for data delivery in sparse mobile ad hoc networks*, in proceedings of 5th ACM Intl. Symp. Mobile Ad Hoc Networking and Computing 2004 (MobiHoc 04), ACM Press, Tokyo, Japan, 24-26 May 2004, pp. 187–198.

[11] D. K. Sharma, S. K. Dhurandher, A. Kumar, A. Kumar, and A. K. Jha, *Cloud computing based routing protocol for infrastructure-based opportunistic networks*, in proceedings of IEEE India International Conference on Information Processing (IICIP), D.T.U., Delhi, India, 12-14 August, 2016.

[12] D. J. Goodman, J. B., N. B. Mandayam and R. D. Yates, *INFOSTATIONS: A new system model for data and messaging services*, IEEE Vehicular Technology Conference 1997(VTC97), vol. 2, May 1997, pp. 969–973.

[13] A. Chhabra, V. Vashishth, and D. K. Sharma, *SEIR: A stackelberg game based approach for energy-aware and incentivized routing in selfish opportunistic networks*, in proceedings of 51st Annual Conference on Information Sciences and Systems (CISS), 2017, 22-24 March 2017, Baltimore, MD, USA, pp. 1–6.

[14] S. K. Dhurandher, D. K. Sharma, I. Woungang, and A. Saini, *An energy-efficient history based routing scheme for opportunistic networks*, International Journal of Communication Systems, Special Issue - Energy Efficient Networking, Volume 30, Issue 7, May 2017, Wiley, DOI: 10.1002/dac.2989.

[15] A. Vahdat, and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Technical Report CS-2000-06, Dept. of Computer Science, Duke University, Durham, NC, 2000.

[16] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, ACM SIGMOBILE, Mobile Computing and Communications Review, vol. 7, Issue 3, July 2003, pp. 19–20.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, *Spray and wait: An efficient routing scheme for intermittently connected mobile networks*, in proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 05), Philadelphia, PA, USA, 22-26 Aug. 2005, pp. 252–259.

[18] D. K. Sharma, S. K. Dhurandher, I. Woungang, J. Arora, and H. Gupta, *History-based secure routing protocol to detect blackhole and greyhole attacks in opportunistic networks*, Journal of Recent Advances in Communications and Networking Technology, Vol. 5, No. 2, Bentham Science, November 2016, ISSN (Print): 2215-0811, ISSN (Online): 2215-082X, DOI: 10.2174/2215081106666161206124014, pp. 73–89.

[19] A. Chhabra, V. Vashishth, and D. K. Sharma, *A fuzzy logic and game theory based adaptive approach for securing opportunistic networks against black hole attacks*, International Journal of Communication Systems, Wiley, 2017, DOI: 10.1002/dac.3487.

[20] D.K. Sharma, S.K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. P. C. Rodrigues, *A machine learning-based Protocol for efficient routing in opportunistic networks*, IEEE SYSTEMS JOURNAL, December 2016, ISSN (Print): 1932-8184, ISSN (Online): 1937-9234, DOI: 10.1109/JSYST.2016.2630923, , pp. 1-7.

[21] D. K. Sharma, A. Yadav, A. Sharma, and J. Kumar,*KNNR:K-nearest neighbour classification based routing protocol for opportunistic networks*, in proceedings of IEEE Tenth International Conference on Contemporary Computing (IC3 2017), 10-12 August 2017, Noida, India.

[22] D. K. Sharma, S. K. Dhurandher, I. Woungang, A. Bansal, A. Gupta, *GD-CAR: A genetic algorithm based dynamic context aware routing protocol for opportunistic networks*, in proceedings of 20th Intl.Conference on Network-Based Information Systems (NBiS-2017), Aug. 24–26, 2017, Toronto.

[23] D. K. Sharma, D. Kukreja, P. Aggarwal, M. Kaur, A. Sachan, *Poisson's probability-based Q-routing techniques for message forwarding in opportunistic networks*, International Journal of Communication Systems, Wiley, 2018; e3593. https://doi.org/10.1002/dac.3593.

[24] D. K. Sharma, S. K. Dhurandher, D. Agarwal, K. Arora, *kROp: k-Means clustering based routing protocol for opportunistic networks*, Journal of Ambient Intelligence and Humanized Computing, Springer, ISSN: 1868-5137 (print version), ISSN: 1868-5145 (electronic version),pp. 1-18, https://doi.org/10.1007/s12652-018-0697-3.

[25] A. Keranen, *Opportunistic network environment simulator*, Special assignment report. Helsinki University of Technology, Dept. of Communications and Networking, May 2008.

[26] S. K. Dhurandher, D. K. Sharma, I. Woungang, S. Bhati, *HBPR: history based prediction for routing in infrastructureless opportunistic networks*, In: Proceedings of IEEE 27th international conference on advanced information networking and applications (AINA 2013), Barcelona, Spain, 2528 March 2013, pp. 931–936.

[27] S. K. Dhurandher, S. Borah, I. Woungang, D. K. Sharma, K. Arora, D. Agarwal, *EDR: An encounter and distance based routing protocol for opportunistic networks*, In: Proceedings of IEEE 30th international conference on advanced information networking and applications (AINA 2016), Crans-Montana, Switzerland, 23-25 March 2016.

[28] S. K. Dhurandher, D. K. Sharma, I. Woungang, R. Gupta, S. Garg, *GAER: genetic algorithm-based energy-efficient routing protocol for infrastructure-less opportunistic networks*, The Journal of Supercomputing, September 2014, Volume 69, Issue 3, pp. 1183–1214.

[29] T. Spyropoulos, K. Psounis, C. S. Raghavendra, *Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility*, In: Proceedings of the fifth IEEE international conference on pervasive computing and communications workshops (PerComW 07), White Plains, NY, 1923 March 2007, pp. 79–85.

[30] Y. Zhao, *Motion vector routing protocol: A position based routing protocol for mobile ad hoc networks*, In the Graduate College, The University of Arizona.
[31] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, *Maxprop: Routing for vehicle-based disruption-tolerant networks*, Proc. of IEEE INFOCOM 2006, 2006, pp. 1–11.

Edited by: Khaleel Ahmad
Received: Nov 25, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1443
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 41–54. http://www.scpe.org © 2019 SCPE

ENHANCED CLUSTERING ALGORITHM BASED ON FUZZY LOGIC (E-CAFL) FOR WSN

PAWAN SINGH MEHRA*, MOHAMMAD NAJMUD DOJA, AND BASHIR ALAM†

Abstract. Since longer lifetime of the network is utmost requirement of WSN, cluster formation can serve this purpose efficiently. In clustering, a node takes charge of the cluster to coordinate and receive information from the member nodes and transfer it to sink. With imbalance of energy dissipation by the sensor node, it may lead to premature failure of the network. Therefore, a robust balanced clustering algorithm can solve this issue in which a worthy candidate will play the cluster head role. In this paper, an enhanced clustering algorithm based on fuzzy logic E-CAFL is propound which is an improvement over CAFL protocol. E-CAFL takes account of the residual energy, node density in its locality and distance from sink and feed into fuzzy inference system. A rank of each node is computed for candidature of cluster coordinator. Experiments are performed for the designed protocol to validate its performance in adverse scenarios along with LEACH and CAFL protocol. The results illustrate better performance in stability period and protracted lifetime.

Key words: Energy Efficiency, WSN, Cluster Formation, Network Lifetime, Fuzzy Logic, Cluster Head

AMS subject classifications. 68M10, 94D05

1. Introduction.

In current scenario, with rapid development in wireless communication, embedded computing and diverse sensor technology, WSN is emerging very swiftly. With enormous and cheap micro sensors deployed in the area of interest, WSN collects the data from surrounding and forward the same to the sink for further processing by end-users. There are several applications of WSN e.g. defence, structural monitoring, industrial monitoring, environmental monitoring, climatic and weather monitoring , natural disaster, health care etc.[1, 2, 3]. WSN is resource restricted with regard to energy, computation and communication.This network possess restricted power supply and requires proficient and proper utilization of sensor node (SN) power for longer lifespan of the network [4, 5]. Since SN are light weight and tiny devices with low power, their transmission range is restricted to conserve energy [6]. Longer distance transmission of data is generally accomplished by multi-hoping or intermediate nodes. In some applications, by making use of auxiliary resource like solar cells, the power source of SN can be top up [7] nonetheless it is not continuous which can hamper the functioning of the device. The key addressing issues for improving the lifetime are network topology and efficient energy consumption. Cluster based schemes effectively maintains network topology by partitioning the field and forming clusters. To drag out the lifespan, designing of clustering algorithm which is efficient in conserving energy is inevitable . Either single hop or multi-hop methods are adapted for communicating sensed information to sink/base station (BS). Evidence from experiments portray that computation is lesser energy expensive than communication [9]. According to [10], for transfer of one bit, the energy dissipated is comparable to approximately 1K operations performed by SN. For conserving the energy, emphasis should be given on two constituents: number of operations performed by SN and the communication method appertain. Blending diverse efficient method may result in extendable lifetime of WSN [11, 12] .

This paper propounds enhanced clustering algorithm using fuzzy logic (E-CAFL) for protracting the network lifespan. Most of the clustering protocols are probabilistic and chooses cluster head on the basis of maximum remnant energy and its farness from BS which is not sufficient to choose the best candidate. This algorithm is distributive in nature which uses deterministic method. E-CAFL is an improvement over CAFL [13]. CAFL considers the remnant power and closeness to BS for calculation of chance of SN while selecting the CH whereas when the nodes choose their CH, it uses remnant power of CH and closeness to their tentative CH. Both the parameters are crucial but the density around the tentative CH is also equally important as more members in cluster will lead to more energy dissipation by CH in transmission and reception of data. Also, CAFL is randomized protocol where the election of CH is dependent upon random number (RN) generated by each SN. SN gets elected as CH if its generated RN is less than a calculated threshold which can sometimes lead to no CH selection in a round as there are chances that RN generated by all the SN are greater than threshold.

*G.L.Bajaj Institute of Technology and Management, Greater Noida, India (pawansinghmehra@gmail.com)
†Jamia Millia Islamia, New Delhi, India

Overcoming the limitations of CAFL, E-CAFL considers node density along with remnant power and separation distance from BS while choosing the best candidates for CH role. Also member nodes intelligently select their CH on the basis of CH-Chance which takes into account the rank of the CH and distance to that CH nodes during cluster formation. Along with that, E-CAFL also eradicates the randomization by selecting only the p% best candidates for CH candidature. For assessing the performance of E-CAFL, simulation experiments are carried out and compared with CAFL and LEACH.

Rest of the outline of this literature is as follows: Relevant work is discussed in Section 2. Preliminary discussion is done in Section 3, E-CAFL protocol is discussed in Section 4. Performance evaluation after simulation is done in Section 5. Finally, concluding remarks with summary of the contribution is discussed in Section 6.

2. Relevant Work. In recent past, ample number of exploration and research work have been carried out on clustering protocols in WSN. In this section, key highlights of some popular and recent clustering techniques especially fuzzy logic (FL) based are talked about. LEACH is the pioneer protocol for cluster formation of SN in WSN [14]. It is a distributive protocol that makes decisions locally for the selection of CH. It is randomized in terms of rotation of CH role to distribute the load evenly. This protocol also performs data compression at CH level so as to minimize the amount of data directed to BS. First implementation of Fuzzy based clustering approach is propound in [15] which is an improvement over LEACH. Network lifetime is efficiently increased by making use of three input constituents (node centrality, degree of node and its energy) in Fuzzy Inference System (FIS). Twenty seven fuzzy if-then rules are used and selection of CH is done by BS using these rules. Since this approach is centralized, therefore it is not suitable for scalable networks. Author in [16] propounds CHEF protocol in which there are two input variables for FIS: nodes remnant energy and local distance. There are nine fuzzy rules to evaluate the fuzzified inputs and calculation of nodes chance to be elected as coordinator of cluster. LEACH-FL [17] employs similar approach to [15]. It considers three descriptor (farness from BS, density around node and node power level) in computing the chance of CH candidature. In [18], a clustering approach with fuzzy logic (FL) is propound to protract the life time of WSN. This protocol is scalable to large WSN. EAUCF [19] propound fuzzy based distributive clustering algorithm in which remnant energy and farness from BS is used for election of CH. There are nine fuzzy IF-THEN rules to select tentative CH. Each tentative CH calculate the competitive radius to compete for CH candidature. But this algorithm doesnt contemplate energy dissipation due to huge intra communication which deteriorates the performance of the protocol. DCFP[20] protocol lessens total network energy dissipation by building the network infrastructure once at the beginning and remain same throughout the lifetime. Fuzzy c means algorithm is adopted to allocate SN to most apposite cluster. MOFCA [21] is another approach in which CH are chosen based on remnant energy and aloofness to BS. If a CH is closer to BS then its competitive radius is greater and it can perform more task like data collection and transmission. The author [22] proposed HEEDML-FL-o which is enhancement over HEED [23] with fuzzy logic incorporation. The inputs considered for FIS are remnant energy and node density. It improves the energy efficacy by 193.84%. DUCF[24] protocol ensures load balancing by forming clusters using fuzzy logic. There are three input parameters to fuzzifier: node density, distance from BS and remaining energy of node. The two output parameters are chance and size. MCFL [25] is somewhat diverse approach to form clusters. Most eligible nodes are selected as CH and are trusted for few rounds to minimize the number of exchange messages in clustering algorithm. Experimental results portray better performance of this algorithm than its comparatives.

The author [26] proposed FZSEP-E which divides the area of interest into zones with heterogeneous nodes. Fuzzy logic is adapted to form clusters with parameters like density, energy and farness from BS and substantially improves the lifetime of WSN. Author in [27] propound two step fuzzy system for selecting suitable CH. It uses descriptors like vulnerability index, energy, farness from BS, density, distance among CH and centrality and successfully balance the network energy consumption. FBECS [28] is distributive algorithm which assigns predefined probability to SN as per their farness from BS. It uses fuzzy logic to select optimal candidates for CH role. Author in [8] propound two methods to improve the efficacy of LEACH. It altered the threshold for best CH candidate selection and modified the TDMA schedule for better transmission mechanism. MACHFL-FT [29] is fuzzy based algorithm for clustering heterogeneous nodes using fixed threshold to avoid re-clustering up to some rounds. The criteria for comparison is based on remnant energy, dead nodes, first dead node, half dead node and last dead node. El Alami and Najid propound CAFL[13] to enhance CFFL protocol. In this

protocol, for the CH selection, two inputs i.e. remnant energy and closeness to BS are fed to FIS in order to compute rank and for the formation of cluster, two parameters which are considered for cluster formation are residual energy of tentative CH and closeness to that CH for increasing efficiency of network.

3. Preliminaries. The objective of proposing E-CAFL is to overcome the limitations of CAFL protocol and improving its efficiency. Before we begin with the description of the E-CAFL protocol, some characteristics of network system are discussed hereafter.

3.1. Some Assumptions and Network Structure. While designing the proposed protocol, some assumptions are made in this protocol which are mentioned below:

- The deployment of SN is randomized.
- The SN are homogeneous with battery level at parity
- BS as well as SN are immobile.
- The power supply to SN is irreplaceable and non-rechargeable whereas BS has continuous power supply.
- The separation distance is computed by RSSI.
- BS is kept aloof from field.
- The communication between any two devices is symmetric.
- SN will be presumed dead only if its power supply gets exhausted.

The target field is presumed to be of 100×100 size as shown in Figure 3.1. SN are scattered indiscriminately over the area of interest and the BS is located at (175, 50).

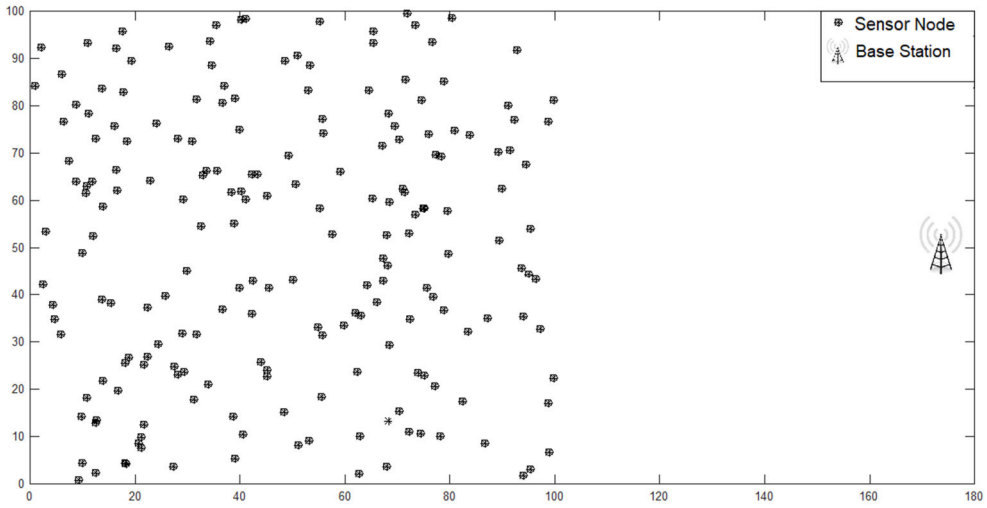


FIG. 3.1. *Network Layout*

3.2. Radio Energy Model. In E-CAFL, first order radio energy model is adapted which is identical to the work presented in [13]. The amount of energy dissipated for transmission and reception of s bits is represented as E_{Tx} and E_{Rx} respectively. However, the system behavior is dependent upon Equations 3.1 and 3.2.

$$E_{Tx}(s,d) = \begin{cases} sE_{elec} + s\epsilon_{fs}d^2, & d < d_o \\ sE_{elec} + s\epsilon_{mp}d^4, & d > d_o \end{cases} \tag{3.1}$$

where $d_o = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$

For receiving s-bits data, the energy required is calculated as follows:

$$E_{Rx}(s) = E_{Rx-elec}(s) = sE_{elec} \tag{3.2}$$

The total energy exhausted by a CH in each round is calculated by Equation 3.3

$$E_{CH} = ns(E_{elec} + \epsilon_{fs} d_{BS} + E_{DA}) \tag{3.3}$$

where the distance between CH and BS is denoted by d_{BS} and n is the count of member nodes. Similarly, the amount of energy dissipated by a cluster member (E_{CM}) is computed by Equation 3.4 in which d_{CH} is the distance to its CH.

$$E_{CM} = s(E_{elec} + \epsilon_{fs} d_{CH}) \tag{3.4}$$

4. E-CAFL. This section discusses enhanced clustering algorithm using fuzzy logic (E-CAFL). It is an improvement over the CAFL protocol which is discussed hereafter. First, the BS is kept aloof from the target field in both the scenario while performing simulation experiments so as to make the application suitable for situation where human reachability is rare or impossible. Second, In addition to the parameters used in CAFL while electing a node as CH, the overhead of the CH is also considered in terms of node density as third parameter because it determines the intra cluster communication cost. Third, randomisation of CH selection is eliminated from CAFL and decision for CH selection is made purely on the basis of rank rather than randomly generated number. Fourth, during the cluster formation, instead of using remnant energy of CH, we have used rank of CH while making any decision to choose CH for member nodes because it will provide profound status of the CH node . The E-CAFL protocol is based on rounds. Once the SN are scattered in the area of interest, E-CAFL protocol comes into play. CH selection and Cluster formation in E-CAFL is explained hereafter.

4.1. Selection of Cluster Head. In every round, only p% CH are elected from alive node. To form clusters, a packet (BS_LOC) is broadcasted by BS in the field which comprises of BS coordinates and a schedule. Each SN broadcasts a packet (INFO_MSG) in field within the transmitter range as per the schedule provided by BS. After the completion of all the broadcasts, local parameters like density, remnant energy and farness from BS are computed by SN. The CH selection algorithm is described in Algorithm 1.

Algorithm 1 CH candidate selection algorithm in E_CAFL

```
SN ← Overall SN in the field
i ← Unique Identity
SN(i).Energy← current power level
SN(i).Type← member
SN(i).Rank← 0
SN(i).Density← Total SN inside Transmitter Range(Rc)
SN(i).DBS← Distance from BS to SN(i)
List_CH ← 0
Count_CH ← 0
Compute rank of each Node(i) using Fuzzy_Logic(Closeness to BS, Node Density,Remnant Energy)
Broadcast rank to proclaim CH candidature
While(Count_CH <=p% )
{
if SN(i).Rank > Received_SN(j).Rank then
    SN(i).Type ← CH
    Count_CH++
    Add SN(i) to List_CH
    Broadcast CL_HEAD packet
end if
}
```

In E-CAFL, to select the CH candidates, fuzzy logic is incorporated. The decision making behaviour of human is efficiently handled by fuzzy logic. In order to calculate the rank of each node, three input variables; density, remnant energy and farness from BS are applied to FIS as depicted in Table 4.1.

TABLE 4.1
Input and its linguistic variables for CH selection

Input Variables	Linguistic Variable		
Node Density	High(H)	Medium(M)	Low(L)
Closeness to BS	Far(F)	Medium(M)	Near(N)
Remnant Energy	High(H)	Medium(M)	Low(L)

Triangular and Trapezoidal MF are used for intermediate and boundary variables respectively since it is simpler with faster computation. Each MF must satisfy a condition that it should range from 0 to 1. Other MF are also there like Bell,Sigmoid, Gaussian etc. which can also be used but proposed work has shown better results with these two MF. The triangular and trapezoidal MF used are given in Equations 4.1 and 4.2 respectively.

$$f(y;t,u,v) = max(min(\frac{y-t}{u-t}, \frac{v-y}{v-u}), 0) \tag{4.1}$$

$$f(y;t,u,v,w) = max(min(\frac{y-t}{u-t}, 1, \frac{w-y}{w-v}), 0) \tag{4.2}$$

These membership function (MF) of crisp input values are framed by using [13] and experimental experience as illustrated in Figure 4.1-4.3. After the fuzzification, the values obtained are provided to rule base to test for IF-THEN conditions. There are twenty seven rules as depicted in Table 4.2 which are applied to obtain the rank.

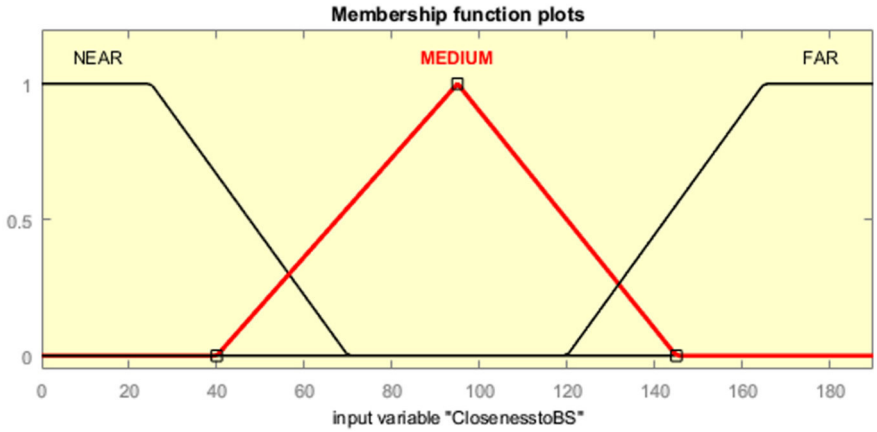


FIG. 4.1. *MF-Closeness to BS*

In this protocol, Mamdani inference system is adapted which is most commonly used [22, 28] because of its characteristics and simplicity. A value is obtained by using AND and OR operators. FIS endeavour to imitate the human inference system in making conclusion from the given set of constraints in knowledge base. The procedure of defuzzification maps the fuzzy set obtained from inference engine into crisp value for drawing conclusion. The center of area (C*) method used for defuzzification process is given in Equation 4.3

$$C^* = \frac{\int \mu_A(y) y dy}{\int \mu_A(y) dy} \tag{4.3}$$

For crisp output, the fuzzy variables used are shown in Table 4.3. Defuzzifier changes the obtained input from inference engine into crisp set using triangular and trapezoidal MF as shown in Figure 4.4 and rank for each node is obtained.

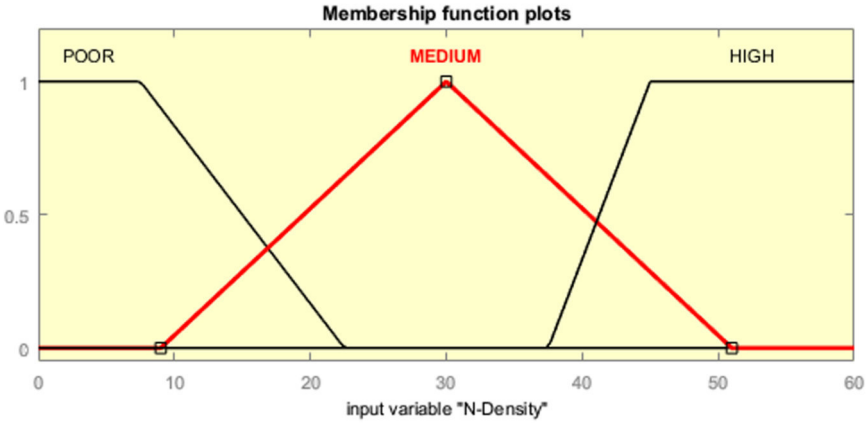


FIG. 4.2. *MF- Node Density*

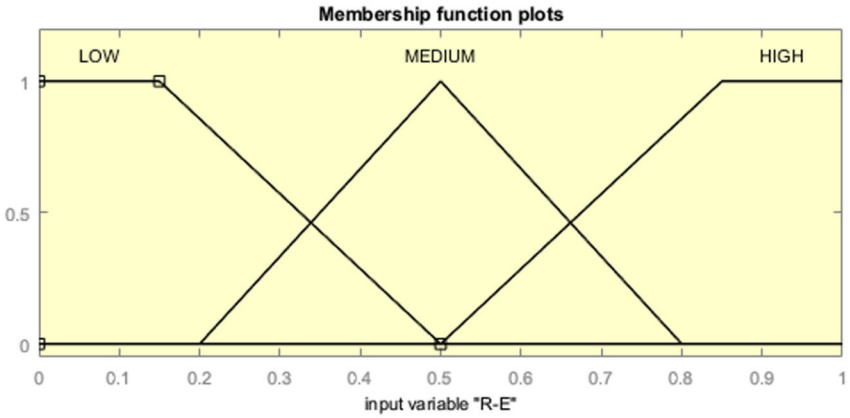


FIG. 4.3. *MF-RES-Energy*

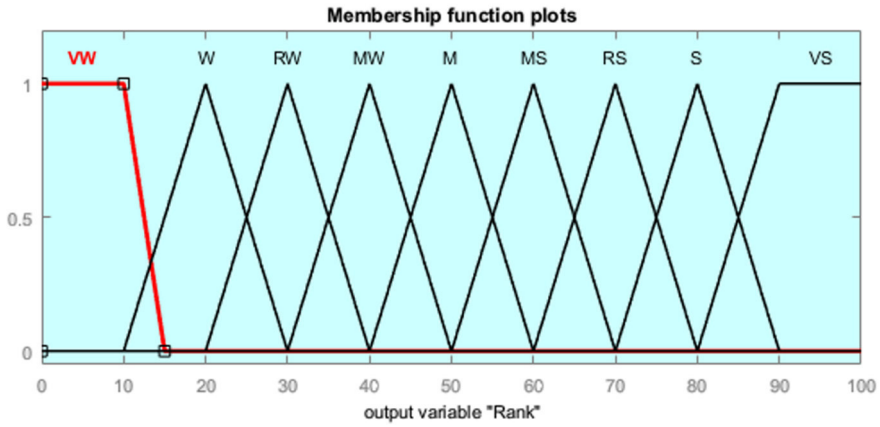


FIG. 4.4. *MF-Rank*

TABLE 4.2
Fuzzy Rules for Node Ranking

Rule No.	R_Energy	DBS	Density	Rank
1	H	F	H	S
2	H	M	H	VS
3	L	M	P	VW
4	H	N	H	VS
5	M	M	H	MS
6	H	F	M	RS
7	M	N	P	MW
8	L	F	H	W
9	H	M	M	S
10	M	M	M	M
11	H	N	M	S
12	M	F	P	RW
13	L	N	H	RW
14	L	F	M	VW
15	H	F	P	MS
16	M	N	M	M
17	M	M	P	MW
18	H	M	P	RS
19	L	F	P	VW
20	L	N	M	W
21	H	N	P	RS
22	M	F	H	M
23	L	M	M	W
24	M	N	H	MS
25	L	N	P	W
26	M	F	M	MW
27	L	M	H	RW

TABLE 4.3
Output and its Linguistic Variables

Output	Linguistic Variable
Rank	Very Strong(VS), Strong(S), Rather Strong(RS), Medium Strong(MS),Medium(M), Medium Weak(Mw), Rather Weak(RW), Weak(W),Very Weak(VW)

Once the rank of all the nodes is computed, each SN broadcasts its rank within the transmitter range as per the schedule. After the completion of all the broadcast, each sensor node compare its rank with all the received rank. If its rank is highest, it will proclaim its candidature for CH by broadcasting (CL_HEAD) packet which contains its density and remnant energy otherwise it will wait to join the optimal cluster. Thus the candidates with higher rank will be elected as CH.

4.2. Cluster formation in E-CAFL. After the selection of CH, rest of the nodes (Non-CH nodes) will now have to make a decision to join one of the cluster. For this, the nodes will calculate the chance of all the elected CH whose packet is received by the node. This cluster formation is depicted in Algorithm 2. Since, the node has CH_Rank received earlier, it will calculate its closeness to that CH. With these two parameters, node will pass it to Fuzzy logic in order to calculate the chance of each CH so that it can join the optimal cluster.

The input and its linguistic variables are listed in Table 4.4 and its MF is shown in Figure 4.5 and 4.6.

Algorithm 2 Formation of Cluster in E_CAFL

```
REC_CH_LIST ← All CH whose packet(CL_HEAD) is received
i ← Distinct ID of sensor nodes
Compute chance of each CH_Node in REC_CH_LIST using Fuzzy logic
OPT_CH ← 0
While(End of REC_CH_LIST )
{
if CH_Node(j).Chance> OPT_CH then
    OPT_CH ← j
end if
}
SN(i).CH=j //SN with ID as j is selected as CH for SN(i)
SN(i) will send a packet (JOIN_REQ) to SN(j)
SN(j) will send ACK to SN(i) with TDMA slot
```

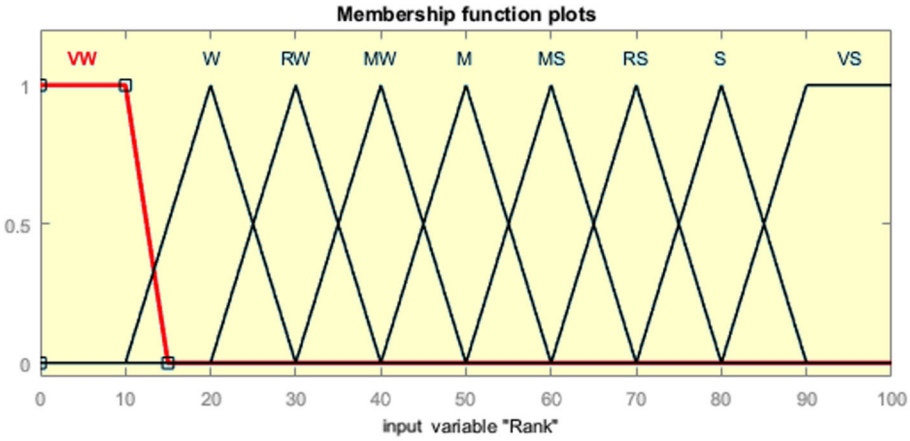


FIG. 4.5. *MF-Rank-CH*

TABLE 4.4
Input and its Linguistic Variables for cluster formation

Input Variables	Linguistic Variable
CH_Rank	Very Strong(VS), Strong(S), Rather Strong(RS), Medium Strong(MS), Medium(M), Medium Weak(Mw), Rather Weak(RW), Weak(W), Very Weak(VW)
Closeness to CH	Far(F),Medium(M),Near(N)

After completion of fuzzification process, the values obtained are passed to rule base to test for IF-THEN rules. For calculating the chance of CH, there are twenty seven rules to be applied as shown in Table 4.5. In this cluster formation, Mamdani model used. The defuzzification process uses (C*) as given in Equation 4.3 and converts the received input into crisp set output. The fuzzy output variables used are similar to output linguistic variables used in Table 4.3. Triangular and Trapezoidal MF used are shown in Figure 4.7 and chance for each CH is obtained.

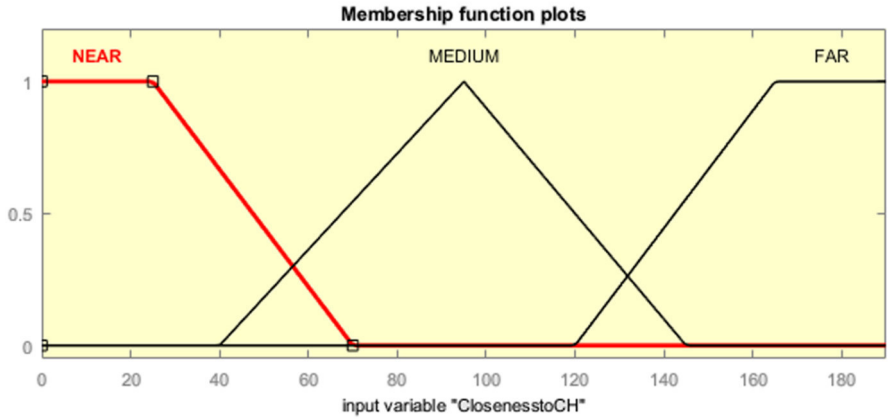


FIG. 4.6. *MF-Closeness to CH*

TABLE 4.5
Fuzzy Rules for CH_Chance

Rule No.	CH_Rank	Closeness to CH	CH_Chance
1	VS	F	S
2	VS	M	VS
3	VS	N	VS
4	S	F	RS
5	S	M	S
6	S	N	S
7	RS	F	MS
8	RS	M	RS
9	RS	N	RS
10	MS	F	M
11	MS	M	MS
12	MS	N	MS
13	M	F	MW
14	M	M	M
15	M	N	M
16	MW	F	RW
17	MW	M	MW
18	MW	N	MW
19	RW	F	W
20	RW	M	RW
21	RW	N	RW
22	W	F	VW
23	W	M	W
24	W	N	W
25	VW	F	VW
26	VW	M	W
27	VW	N	RW

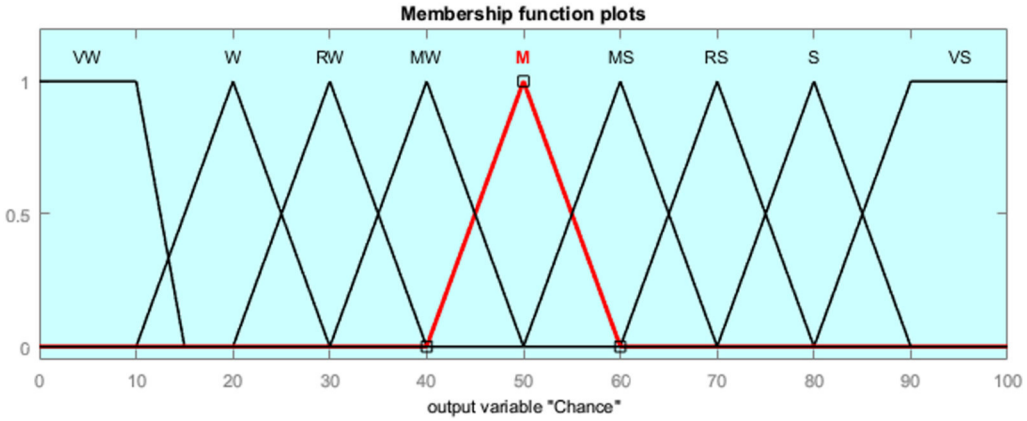


FIG. 4.7. *MF-Closeness to CH*

After obtaining the chance of each CH, node will choose to join the cluster whose CH is having highest chance. Node will send a (JOIN_REQ) packet to the CH. The CH will accept the request and send acknowledgement packet(ACK) to the node for confirmation. In this way, all clusters are successfully formed. Afterwards, CH will send TDMA slot to all member nodes to collect data evading collision. After the collection of information sensed by member nodes, CH fuses the data for minimizing the communication cost. The fused data will transferred to BS for further processing by end user. Thus, E-CAFL will complete one round.

5. Simulation and Performance Evaluation. To analyse the performance of E-CAFL, simulation experiments are carried out and compared with LEACH and CAFL protocol. The field size is chosen to be $100m^2$ with static SN scattered in random fashion. The BS is located at (50,175) which is remotely located from the field in order to make the protocol widely applicable for any environment. Matrix laboratory is used for simulation work as it is easier for Fuzzy logic implementation and graphical results can be seen. The configuration of simulation parameters are shown in Table 5.1. The simulation experiments are performed extensively to obtain normalised results. For performance evaluation, two scenarios are considered in which some of the simulation parameters are varied from one another. In scenario 1, the SN are available with $E_o = 0.5J$ and N=100 where as in scenario 2, $E_o = 1J$ and N=200. For fair comparison, simulation parameters are kept similar for E-CAFL, LEACH and CAFL. The performance comparison is based on following metrics:

- **Alive Nodes** : It depicts the count of nodes still alive in each round.
- **Average Energy** : It shows the total energy of the network available in each round.
- **FND, QND and HND** :FND is the First Node death, QND is the Quarter Node death and HND is the Half Node Death of the protocols.
- **Throughput**: It is the number of information messages successfully delivered to the BS.

TABLE 5.1
Configuration of Simulation parameters

Description	Symbol	Values
Total SN in the Field	N	100/200
Amplifier energy for free space	ϵ_{fs}	$10pJ/bit/m^2$
Amplifier energy for multipath	ϵ_{mp}	$0.0013pJ/bit/m^4$
Battery level before deployment	E_o	$0.5J/1.0J$
Data packet Size	M	4000bits
Electronic Circuitry	E_{elec}	$50nJ/bit$
Data Fusion	E_{DA}	5nJ/bit/report

5.1. Number of Alive Nodes. More information can be collected from the network if large count of alive SN are available in the field. As shown in Figure 5.1, a plot for number of alive nodes in the network per round is depicted for both the scenarios. The death of SN occurs at 822 round where as for CAFL and LEACH it is

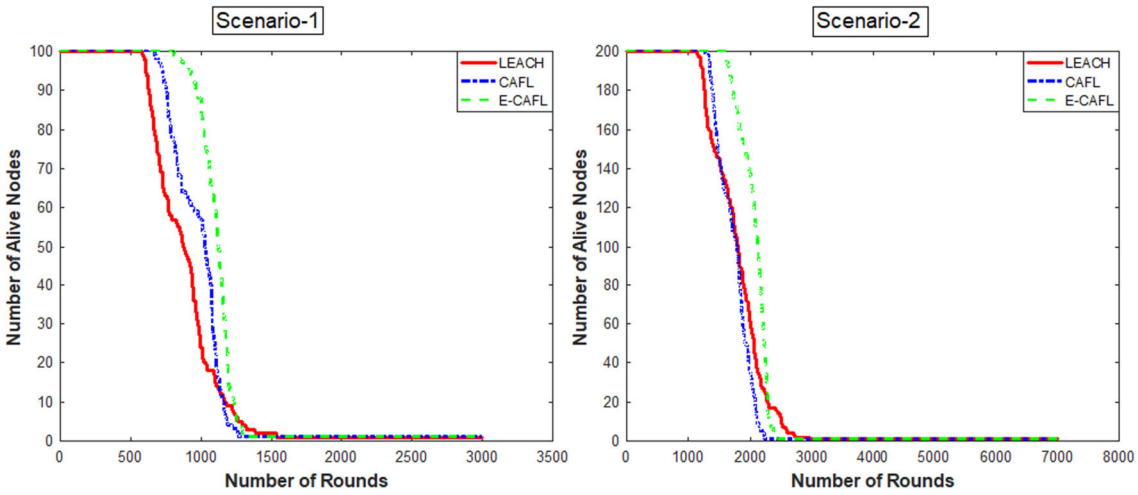


FIG. 5.1. *Number of alive nodes*

579 and 641 round respectively for scenario 1 whereas for scenario 2, stability period is achieved till 1502 rounds where as in case of CAFL and LEACH it is 1133 and 1220 rounds for death of first node. From these results we can conclude that E-CAFL dissipated energy in balanced manner. Both the protocols have performed poor due to non-consideration of node density which also plays an important role. With early expiration of SN, network becomes unstable as well as unreliable due to poor coverage. Better reliability and stability requires perishing of SN after later round.

5.2. Average Energy dissipation rate. Wireless communication fritters a large amount of energy. With the increase of rounds, average energy of the network decreases resulting in death of nodes. Figure 5.2 depicts the average energy dissipation rate of the network. When both the scenarios are considered, we can see that

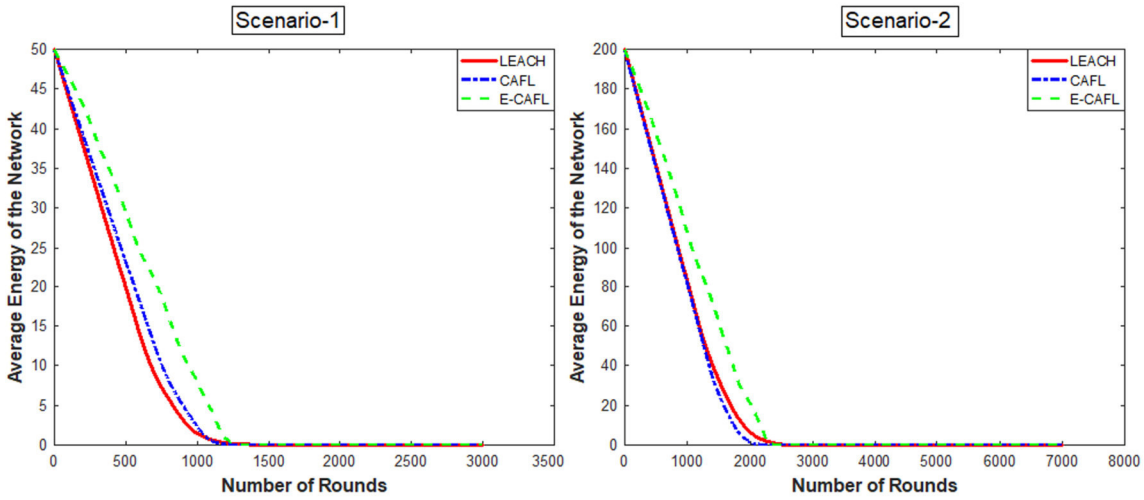


FIG. 5.2. *Average energy dissipation rate*

E-CAFL dissipates less energy per round as compared to CAFL and LEACH protocol which reveals balanced load distribution of the proposed protocol resulting in longer stability period.

5.3. FND, QND and HND. After establishing the network, target is to gather more information from the field where network lifetime is essential. Premature expiration of SN may lead to uncovered regions of the field resulting in poor performance. Figure 5.3 shows the first node death, quarter node death and half node death metrics gathered from simulation experiments. In scenario 1, E-CAFL achieves 29.56% and 21.04%

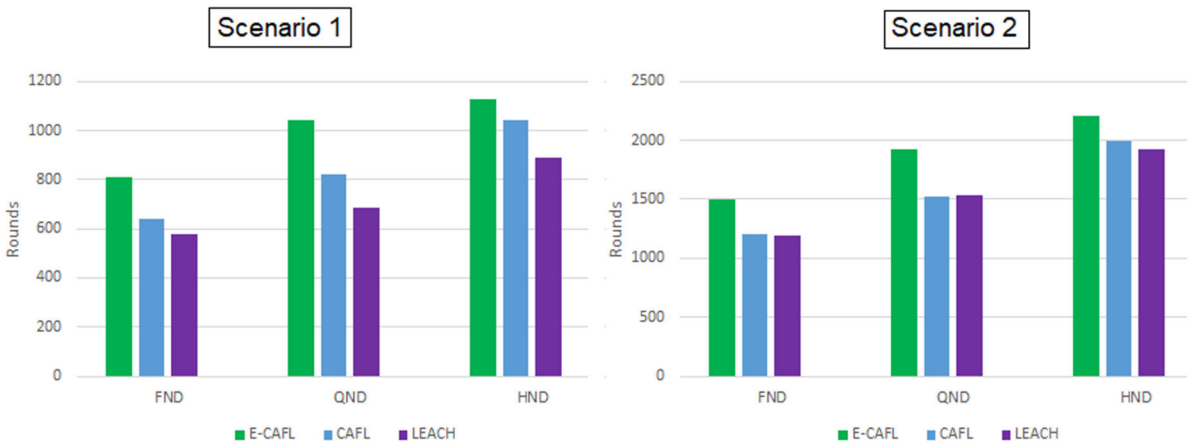


FIG. 5.3. *FND, QND and HND*

improvement over LEACH and CAFL in FND where as it is 21.02% and 20.42% for scenario 2. QND is prolonged by 34.06% and 21.20% as compared to LEACH and CAFL for scenario 1, likewise for scenario 2, it is 19.89% and 20.83% better than LEACH and CAFL respectively. E-CAFL protracts HND by 21.20% and 8.2% in comparison with LEACH and CAFL respectively for scenario 1. Likewise for scenario 2, it is improved by 13.12% and 10.02% over LEACH and CAFL respectively. These simulation results illuminate the proficiency of E-CAFL in better lifetime and protracted stable region.

5.4. Throughput of the network. More successful delivery of packet to BS demonstrates better throughput of the network. As we can see in Figure 5.4, for scenario 1, E-CAFL delivers 38.28% and 34.66% more packet as compared to LEACH and CAFL respectively which means large amount of information is collected at BS. Similarly, for scenario 2, E-CAFL has better throughput of 34.54% and 28.68% over LEACH and CAFL protocol respectively.

6. Conclusion. For proliferation of energy efficiency of the network and balanced energy consumption by the SN, fuzzy based clustering algorithm (E-CAFL) is propound. E-CAFL improves the CAFL protocol by considering the node density while making any decision be it rank while selecting CH candidature or chance while joining clusters by member nodes. Also, randomization in protocol may sometimes lead to zero CH which is eradicated in E-CAFL protocol. It is not possible always to place the BS within the network as there are some applications where WSN is unattended and SN are deployed on the fly. So in proposed work, BS is kept at distant place in order to satisfy all kind of applications. From the simulation experiments, the designed protocol (E-CAFL) performs better than its comparative i.e. LEACH and CAFL. E-CAFL significantly enhances the stability period in both the scenarios with better throughput by balancing the load of the network.

REFERENCES

[1] J. YICK, B. MUKHERJEE, AND D. GHOSAL, *Wireless sensor network survey*, Computer networks, 52 (2008), pp. 2292–2330.
[2] P. S. MEHRA, M. N. DOJA, AND B. ALAM, *Zonal based approach for clustering in heterogeneous WSN*, International Journal of Information Technology, (2017), pp. 1–9.

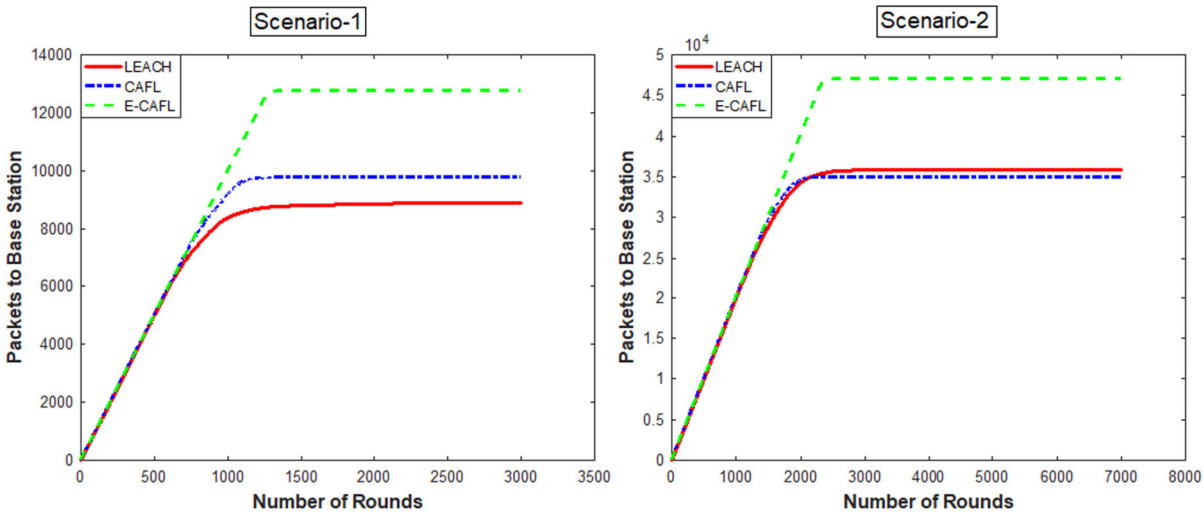


FIG. 5.4. *Throughput of the network*

[3] S. TANWAR, N. KUMAR, AND J. J. RODRIGUES, *A systematic review on heterogeneous routing protocols for wireless sensor network*, Journal of Network and Computer Applications, 53 (2015), pp. 39–56.
[4] S. ARJUNAN AND S. POTHULA, *A survey on unequal clustering protocols in Wireless Sensor Networks*, Journal of King Saud University - Computer and Information Sciences, (2017).
[5] S. TANWAR, S. TYAGI, N. KUMAR, AND M. S. OBAIDAT, *LA-MHR: Learning Automata Based Multilevel Heterogeneous Routing for Opportunistic Shared Spectrum Access to Enhance Lifetime of WSN*, IEEE Systems Journal, (2018), pp. 1–11.
[6] P. S. MEHRA, M. N. DOJA, AND B. ALAM, *Codeword Authenticated Key Exchange (CAKE) light weight secure routing protocol for WSN*, International Journal of Communication Systems, 32 (2019).
[7] R. WANT, K. FARKAS, AND C. NARAYANASWAMI, *Guest Editors' Introduction: Energy Harvesting and Conservation*, IEEE Pervasive Computing, 4 (2005), pp. 14–17.
[8] M. ELSHRKAWEY, S. M. ELSHERIF, AND M. ELSAYED WAHED, *An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks*, Journal of King Saud University - Computer and Information Sciences, 30 (2018), pp. 259–267.
[9] V. RAGHUNATHAN, C. SCHURGERS, SUNG PARK, AND M. SRIVASTAVA, *Energy-aware wireless microsensor networks*, IEEE Signal Processing Magazine, 19 (2002), pp. 40–50.
[10] G. J. POTTIE AND W. J. KAISER, *Wireless integrated network sensors*, Communications of the ACM, 43 (2000), pp. 51–58.
[11] S. TANWAR, N. KUMAR, AND J.-W. NIU, *EEMHR: Energy-efficient multilevel heterogeneous routing protocol for wireless sensor networks*, International Journal of Communication Systems, 27 (2014), pp. 1289–1318.
[12] P. S. MEHRA, M. N. DOJA, AND B. ALAM, *Stable Period Enhancement for Zonal (SPEZ)-Based Clustering in Heterogeneous WSN*, in Proceedings of First International Conference onSmart System, Innovations and Computing, (2018), pp. 887–896.
[13] H. EL ALAMI AND A. NAJID, *Fuzzy Logic Based Clustering Algorithm for Wireless Sensor Networks*, International Journal of Fuzzy System Applications (IJFSA), 6 (2017), pp. 63–82.
[14] W. HEINZELMAN, A. CHANDRAKASAN, AND H. BALAKRISHNAN, *Energy-efficient communication protocol for wireless microsensor networks*, in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol. vol.1, IEEE Comput. Soc, 2000, p. 10.
[15] I. GUPTA, D. RIORDAN, AND S. SAMPALLI, *Cluster-Head Election Using Fuzzy Logic for Wireless Sensor Networks*, in 3rd Annual Communication Networks and Services Research Conference (CNSR'05), IEEE, 2005, pp. 255–260.
[16] J. KIM, S. PARK, Y. HAN, AND T. CHUNG, *CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks*, in Proceedings of 10th International Conference on Advanced Communication Technology, 2008, pp. 654–659.
[17] G. RAN, H. ZHANG, AND S. GONG, *Improving on LEACH Protocol of Wireless Sensor Networks Using Fuzzy Logic*, Journal of Information & Computational Science, (2010).
[18] J.-S. LEE AND W.-L. CHENG, *Fuzzy-Logic-Based Clustering Approach for Wireless Sensor Networks Using Energy Predication*, IEEE Sensors Journal, 12 (2012), pp. 2891–2897.
[19] H. BAGCI AND A. YAZICI, *An energy aware fuzzy approach to unequal clustering in wireless sensor networks*, Applied Soft Computing, 13 (2013), pp. 1741–1749.
[20] O. M. ALIA, *A decentralized fuzzy c-means-based energy-efficient routing protocol for wireless sensor networks*, The Scientific World Journal,(2014).
[21] S. A. SERT, H. BAGCI, AND A. YAZICI, *Mofca: Multi-objective fuzzy clustering algorithm for wireless sensor networks*, Applied Soft Computing, 30 (2015), pp. 151–165.
[22] S. SINGH, S. CHAND, AND B. KUMAR, *Energy Efficient Clustering Protocol Using Fuzzy Logic for Heterogeneous WSNs*,

Wireless Personal Communications, 86 (2016), pp. 451–475.
[23] O. Younis and S. Fahmy, *HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks*, Mobile Computing, IEEE Transactions on, 3 (2004), pp. 366–379.
[24] B. Baranidharan and B. Santhi, *DUCF: Distributed load balancing Unequal Clustering in wireless sensor networks using Fuzzy approach*, Applied Soft Computing, 40 (2016), pp. 495–506.
[25] M. Mirzaie and S. M. Mazinani, *MCFL: an energy efficient multi-clustering algorithm using fuzzy logic in wireless sensor network*, Wireless Networks, (2017), pp. 1–16.
[26] S. A. Sahaaya Arul Mary and J. B. Gnanadurai, *Enhanced Zone Stable Election Protocol based on Fuzzy Logic for Cluster Head Election in Wireless Sensor Networks*, International Journal of Fuzzy Systems, 19 (2017), pp. 799–812.
[27] Y. K. Tamandani, M. U. Bokhari, and Q. M. Shallal, *Two-step fuzzy logic system to achieve energy efficiency and prolonging the lifetime of WSNs*, Wireless Networks, 23 (2017), pp. 1889–1899.
[28] P. S. Mehra, M. N. Doja, and B. Alam, *Fuzzy based enhanced cluster head selection (FBECS) for WSN*, Journal of King Saud University - Science, (2018).
[29] M. Mirzaie and S. M. Mazinani, *MACHFL-FT: a fuzzy logic based energy-efficient protocol to cluster heterogeneous nodes in wireless sensor networks*, Wireless Networks, (2018), pp. 1–13.

Edited by: Khaleel Ahmad
Received: Nov 15, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1432
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 55–70. http://www.scpe.org © 2019 SCPE

ZONE-BASED ENERGY EFFICIENT ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS

RAJAN SHARMA*, BALWINDER SOHI†, AND NITTIN MITTAL‡

Abstract. This paper proposes a novel zone or grid-based network deployment framework for energy efficient selection and reselection process of Zone-Head (ZH) in the WSNs. The proposed zone head reselection process ensures energy efficiency, load balancing, and stability which further prolongs the network lifetime. Instead of carrying out periodic reselection of Zone-Head (ZH) that leads to extra energy consumption and network overhead, the protocol dynamically initiates the process of reselection based on residual energy level of ZH. In the proposed approach the process is segregated into four phases; deployment phase, the zone formation phase, zone head selection phase, data transmission phase and reselection phase. We implemented the proposed algorithm in MATLAB and its result outcomes reveal that the proposed method outperforms the competitive algorithms for parameters such as load balancing, total energy consumption and network lifetime.

Key words: DRESEP, SEECP, WSN, Residual energy, Network lifetime

AMS subject classifications. 68M10, 68M12

1. Introduction. In Wireless Sensor Networks (WSNs), routing protocol plays key role in increasing network energy efficiency and is a source of inspiration for those researchers who attempts to save the energy of wireless sensor node and to enhance the lifetime of the network in parallel [1]. The operation which consumes most of the energy in WSNs is the activity of data packet routing. The characteristics of the WSNs are unique in contrast to traditional networks. These unique characteristics are often taken into account for addressing the issues and challenges related to network coverage, node distribution, node administration, network deployment, energy efficiency, security [2-4] and so forth. In recent years, WSNs have been widely investigated [3- 5]. WSN typically consists of large number of low cost unattended multifunctioning sensing nodes that are typically deployed in large quantities and in a high density manner with limited energy resource [1]. These sensing nodes are linked by wireless medium using radio, infrared, or optical frequency band. These networks have various applications like flood and fire detection in remote areas, traffic surveillance, air traffic control, and so forth. Sensors jointly gather ambient condition information such as temperature, pressure, and humidity from their surrounding environment and forward it towards static data sink. In many scenarios, as nodes are deployed in remote and dangerous area, replacement of their batteries becomes impossible. So they must work without replacing their batteries for many years [6]. Thus power management has become one of the fundamental issues of WSNs. The factors which causes energy consumption in WSNs and deteriorates the network lifetime are collision, overhead, overhearing, idle listening, complexity and traffic fluctuations. These factors deplete the energy resources of WSNs. In WSNs, single-hop routing consumes more energy and leads to unbalancing the energy distribution to the nodes which are far from base station (BS). On the other hand, limited radio range of the node and other environmental factors (obstacles, noise, interference, etc.) make single-hop communication infeasible [7]. In WSNs nodes are often randomly distributed across a given geographical area. In such situation some regions in the network get densely populated whereas others receive less number of nodes. Cluster and grid-based algorithms [8-9] are employed to cope up with this problem. Cluster based schemes minimize energy consumption and simplify network management by treating related nodes in groups. Cluster-based approaches increase scalability, robustness, provide load balancing and data aggregation [1, 6, 9]. These algorithms are utilized for clustering in efficient manner where the entire terrain is segregated into virtual grids, and are widely used because of its simplicity, scalability, and uniformity in energy consumption across the network [10]. In literature, different energy efficient cluster-based and grid-based algorithms have been proposed such as LEACH [11], TEEN [12], CBDAS and GBDD [13] but still load balancing and energy efficiency are open issues because of the randomized nature of WSN. The iterative process of cluster formation and CH reselection requires transmitting continuous control messages which results in extensive energy consumption of the nodes

*Research Scholar, IKG Punjab Technical University, Kapurthala, Jalandhar.(rajansharma.ece@gmail.com).
†CGC Group of Colleges.(balsohi18@gmail.com)
‡Chandigarh University.(mittal.nitin84@gmail.com)

and leads to poor performance of the network. This paper focuses on a grid or zone-based technique that can ensure load balancing and intelligent selection and reselection of zone head (ZH) to maximize network lifetime. Our main contribution is to develop a robust network model, which has been developed to deal with variability in deployment area and node density. The network is divided into equal squared size grids and the number of nodes in each zone is determined by their coordinates. Dual-hop communication strategy between ZH and BS is used to attain load balancing. After the topology construction, ZH selected is the one having the maximum energy level and distance value. The role of ZH is rotated based on some threshold to increase network stability and overall operational lifetime. This paper is segregated as follows. Section 2 describes about related work. The Section 3 and 4 are focused on the network model and proposed ZEERP protocol respectively. Simulation results of ZEERP are discussed in Section 5. Stability-aware model of ZEERP is given in Section 6 and its performance is presented in Section 7. In section 8, the proposed work is concluded with future scope.

2. Related work. The main operational sustainability concern in WSN is its energy resource constraint. In recent years, numerous routing protocols have been proposed for WSNs based on the network organization and the routing protocol operations [6]. Some of these focused on minimizing the communication distance to reduce the energy consumption and a handful of them focused on fair energy distribution to avoid the hot spot problems. In existing literature, many researchers have proposed different clustering techniques that are briefly discussed below. Heinzelman et al. presented a low energy efficient adaptive clustering hierarchy (LEACH) algorithm in which sensor nodes arrange themselves into clusters for data aggregation [11]. In this protocol, data aggregation is performed periodically by cluster heads (CHs) to reduce the redundancy in information to be communicated to BS. In setup phase, CHs are nominated on the basis of two parameters; percentage of CH selection and number of times a particular node has been appointed as CH. To make a decision, a random number (between 0 and 1) is chosen by the node. To become a CH for the present round, the number picked by node should be less than threshold value. During steady state phase, the data transmission takes place between non CHs to CH, and then data is transmitted finally to BS. LEACH-centralized (LEACH-C) [14] uses BS as a centralized point to create clusters in an optimal manner. The overhead of collecting clustering information at BS to form the clusters is a drawback of LEACH-C. Manjeshwar et al. presented a reactive clustering algorithm named threshold-sensitive energy efficient sensor network (TEEN). It senses the medium continuously and transfers the data less frequently [12]. In TEEN, the network consists of homogeneous nodes and is divided into two level CHs. First level and second level CHs are distinguished in such a way that the CHs which are far from BS are termed as first level CHs and the closer ones are called second level CHs. It is efficient for the time-critical data sensing applications. In this, a CH sends its members the values of hard and soft threshold [12]. Once hard threshold value is achieved, nodes forward the data to CH. The subsequent data is transmitted only when the environment changes by a minimum of soft threshold value. The main drawback of TEEN is that sensor nodes can never transfer data to BS if the threshold values are not achieved. To get better performance, stable election protocol (SEP) (Smaragdakis, Matta, & Bestavros, 2004) is proposed to maintain the hierarchical routing in WSNs where two types of nodes have their own election probability. Kumar et al. presented a clustering protocol named EEHC in the heterogeneous model [16]. The network is divided into three categories according to the initial energy of nodes; normal, advanced and super nodes. Apparently, normal nodes have the least energy, and super nodes have the highest level of energy. EEHC is based on SEP, and the three types of nodes in EEHC have their own election probability to be CHs within a fixed time to keep stable. Energy efficient and scalable sensor network can be achieved through clustering and multi-hop transmission. It has attracted much attention of the researchers. In [17], Kumar et al. improved EEHC further and proposed a multi-hop clustering protocol called MCR. In MCR, the multi-hop path is built to reduce the energy consumption. Mittal et al. proposed a clustering algorithm using dual-hop communication between CH and BS, suitable for event driven applications called DRESEP [18] and its stability-aware algorithm named SEECP [19] to improve its stability period. Researchers combined the cluster scheme with the biologically inspired routing scheme, and they proposed the evolutionary algorithms (EAs). The EAs are used to handle the cluster-based problem to minimize energy consumption and improve network lifetime with heterogeneity [20-23]. These routing schemes which are inspired by EAs demonstrated their advantages in prolonging the lifetime of hierarchical WSNs. The techniques discussed here are used to minimize the energy consumption of network. Besides all these amenities, clustering leads to hotspot problem in which certain number of sensors

expire early because of their excessive usage. This results in network partitioning and polarization of nodes. In Grid-based clustering, the whole network area is divided into virtual grids. In GBDD [13], the network is divided into grids (also called cells) initiated by the BS. The first node interested in communicating data is set as the crossing point for the grid and its coordinates become the reference point for the grid creation. In this approach, it is often difficult to achieve preferred number of grids required by the network scenario. In a similar approach, the whole network is partitioned into grids based on the node location where midpoints are computed using the membership degree [24-25]. In another approach, the network is divided into two levels of square shaped grids; low level and high level [26]. Low level is for in-cluster data gathering whereas high level is used for inter-cluster data transmission. In unequal clustering mechanism, whole network is segregated into variable size clusters in which CH reserves more energy for inter-cluster communication in order to avoid hotspot problem [27]. Authors have used multi-hop energy aware routing scheme to balance and minimize the energy load of the CH for inter-cluster communication. Variable size clustering algorithms can result in balanced energy consumption maximizing network lifetime. However, extra advertisements for cluster head selection may lead to extra computation and energy overhead. There are various protocols, techniques and algorithms designs which saves the energy of WSNs. Distance-based Residual Energy-efficient Stable Election Protocol (DRESEP) [18] and its stability-aware version named Stable Energy Efficient Clustering Protocol (SEECP) [19] are energy efficient solutions in WSNs. In [28], proposed a Hierarchical Energy Efficient MAC protocol (HEEMAC) which combines the supremacy of LEACH and CSMA approach. This protocol utilizes the concept of CSMA based data transmission along with neighbor acknowledgement (N-Ack). Results proves that proposed protocol delivers high throughput, energy saving and prolonging lifetime of network. In [29], authors proposed an ANN based framework for evaluating optimal cost routing using BB-BC optimization approach in the WSNs. The integrated link cost is a function of average delay from end to end, residual energy of node and throughput of the sensor network. Authors focused on the literature review of variety of MAC protocols for WSNs based upon different parameters such as energy efficiency, throughput delay, and packet loss [30]. In the literature, we observed that the main aim is to keep the wireless sensor nodes in energy saving mode to the extreme extent and to minimize the parameters which are source of energy wastage such as re-transmission, congestion, overhear, idle channel sensing and overhead due to control packets so that lifespan of network can be prolonged. R. Sharma et al. investigates the impact of DVFS and DMS schemes on energy consumption and lifetime of sensor node [31]. In [32-37], aithors focused on performance, energy efficiency and secured solutions in WSNs. Looking at the above discussion, we can summarize that the services provided by different clustering techniques still have several shortcomings that need to be addressed, for instance, network management overhead, hotspot problem, and broadcasting issues. The above discussion also shows that grid-based system is a better option but the dynamic nature of sensor networks makes it difficult to predict the size of grids and number of nodes. In the proposed technique, the problems of hotspot, non-uniform distribution of nodes (load balancing), and computation overhead have been addressed. Furthermore, the proposed technique is not only energy efficient but also performs better on load balancing when compared with state-of-the-art techniques.

3. Energy dissipation radio model. Power consumption is one of the key factors in designing WSN protocols as sensor nodes are highly energy limited. Sensors consume energy for data processing, wireless communication and sensing data. The network energy is consumed on both sides of the communication (transmission and reception) according to a wireless energy consumption model as shown in Figure 3.1 [11, 18, 22]. The model consists of two parts reflecting transmission and reception as shown in equation 1 and 2 respectively. SN needs to consume energy E TX to run the transmitter circuit and E amp to activate the transmitter amplifier, whereas a receiver consumes E RX power for running the receiver circuit. Energy consumption in wireless communication also depends on message length (l) [18, 23]. The transmission cost for a l-bit message having communication distance d, is calculated as

$$E_{TX} = \begin{cases} lE_{elec} + l\varepsilon_{friss\_amp}d^2, & if\ d < d_0 \\ lE_{elec} + l\varepsilon_{two\_ray\_amp}d^4, & if\ d \geq d_0 \end{cases} \tag{3.1}$$

whered_0 is crossover distance and is given by:

$$d_0 = \sqrt{\varepsilon_{friss\_amp}/\varepsilon_{two\_ray\_amp}} \tag{3.2}$$

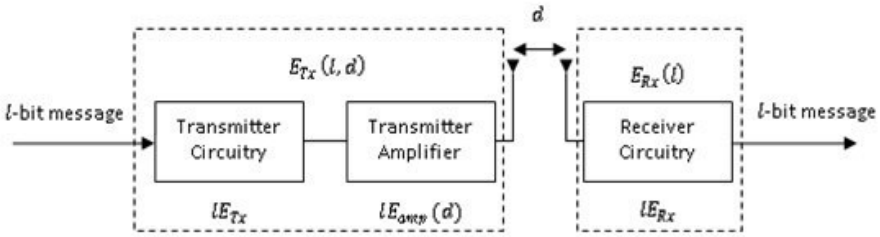


FIG. 3.1. *Energy radio model diagram.*

The term E_{elec} signifies the per-bit energy expended for transmission. The parameters ε_{friss_amp} and $\varepsilon_{two_ray_amp}$represent energy expended to run radio amplifier in free space and two ray ground propagation models, respectively. The reception cost for the l-bit data message is given by:

$$E_{RX} = lE_{elec} \tag{3.3}$$

whereE_{elec} is the per-bit energy consumption for reception.

4. ZEERP. To address the problem of load balancing and energy consumption mentioned in the previous section, we proposegrid or zone-based network framework named Zone-based energy efficient routing protocol (ZEERP). It is a reactive, load balancing and energy efficient protocol for WSNs. The proposed approach evenlydistributes the load across the network, improves networkmanagement, and extends network lifetime. Figure 4.1 presents the proposed framework.The process is segregated into the following main phases.

(i) Deployment Phase: The total number of nodes (where = 1, 2, 3, . . . ,) are randomly deployed in a square targeted area (A = FH × FW), where and are the field height and width, respectively. We assume some default node parameters, for instance, coordinates, node ID, and energy level. Once the topology is built and nodes are deployed, they share this configuration information with the BS. This information is later used by BS for carrying out the grid formation procedure more efficiently.

(ii) Zone Formation Phase: In this phase, the data collected from different sensors is used to form zones and construct topology as presented in Algorithm 1. The grid denotes a single zone recognized by unique zone ID. Once the zone formation phase is completed, BS governs the number of nodes per zone by computing the initial and finish point of each zone as mentioned in Algorithm 1. Figure **??**shows zone formation where nodes are randomly deployed across grids. In this figure, C0 to Cm-1 are the columns and R0 to Rm-1 represent rows. (Zxs, Zys) and (Zxe, Zye) represent the start and end of each zone. Zone height (Z_H) and zone width (Z_W) of each grid are calculated as

$$Z_H = F_W/M \tag{4.1}$$

$$Z_W = F_W/M \tag{4.2}$$

where F_H and F_W represent the height and width of field.

(iii) Zone Head Selection Phase: Zone Head (ZH) selection is very crucial for any energy efficient protocol. ZH is responsible for processing or making any decision upon the received data. ZH selection is an important process; therefore, it is required to define criteria before selection of the ZH. The performance of a zone depends on the ZH; therefore, it is significant to elect the best node as the ZH among available nodes. In the proposed technique two parameters: residual energy level R_E and average distance value A_D are aggregated to come up with ZH election value (ZH_{EV}) of a single node i as follows:

$$ZH_{EV}(i) = 0.5 * R_E(i) + 0.5 * (1/A_D(i)) \tag{4.3}$$

The energy level of the node i is represented by R_E; initially it will be the same for all nodes. Higher value of R_E increases the chance of the candidate node for becoming the ZH. A_D is the average distance

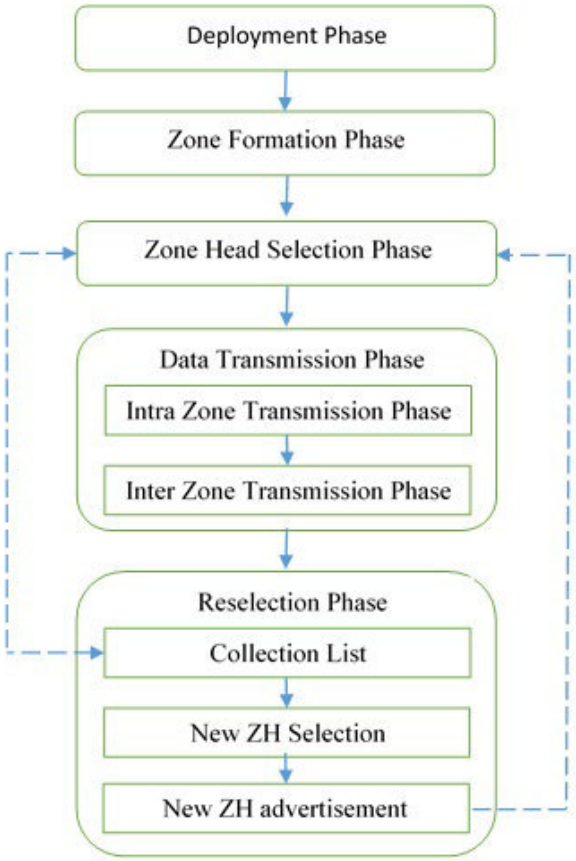


FIG. 4.1. *Framework of the proposed technique.*

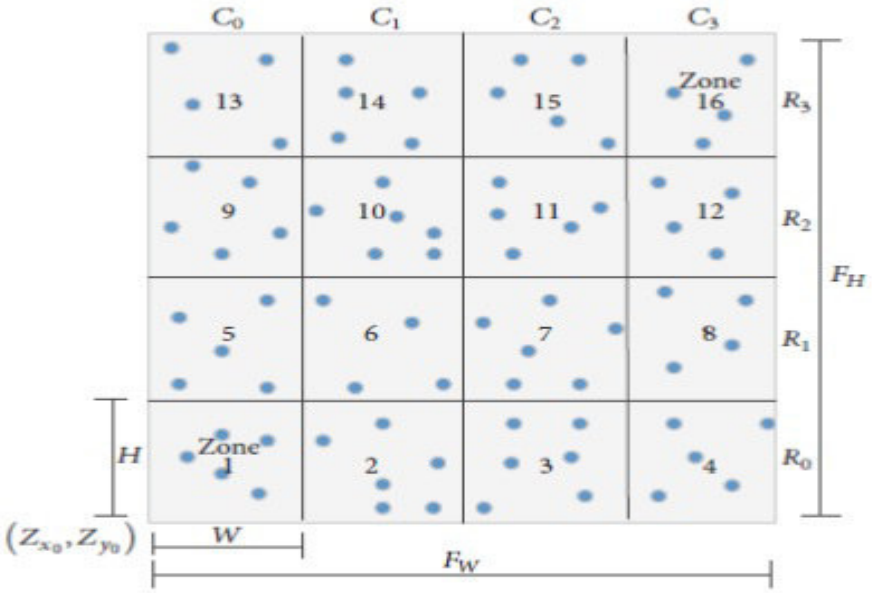


FIG. 4.2. *Zone formation.*

Algorithm 1 Zone formation

begin
Input: Number of Zones ; Height and Width of Deployment Area ; Zone Height Z_H and width Z_W ;
for Zn = 1: N **do**
R = Z_n/M /* Number of rows */
R = $[(Z_n - 1) mod M] + 1$ /* Number of columns */
$Z_{xs} = C \times Z_H - Z_W$ /* Starting x -coordinates of zone*/
$Z_{ys} = R \times Z_W - Z_H$ /* Starting y -coordinates of zone*/
$Z_{xe} = Z_{xs} + Z_W$ /* Ending x -coordinates of zone*/
$Z_{ye} = Z_{ys} + Z_H$ /* Ending y -coordinates of zone*/
for i = 1: N **do**
$if(nx >= Zxx \& nx < Zxe \& ny >= Zys \& ny < Zye) then$
increment ZD(Zn)
NodeInZone(n) = Zn
End if
end for
end for
End

value of each individual node in that specific zone as given in (7). A_D is the distance of a node from all other nodes within the zone and from center of the zone as shown in (8) and (9), respectively. Minimum value of A_D, calculated by BS, will increase nodes chance to be a ZH. BS will get the A_D of all nodes within the network, which will be calculated once

$$A_D(i) = cent(i,c) + \frac{1}{n-1}\sum_{j=1}^{n} d(i,j),\ j \neq i \tag{4.4}$$

where $d(i,j)$ is distance of a node from other nodes in its zone. In order to know how far a node is from other nodes which are in direct transmission with it, consider

$$d(i,j) = \sqrt{(ix - jx)^2 + (iy - jy)^2} \tag{4.5}$$

where$cent(i,c)$ is the center of zone. In order to know the position of the node in its zone, consider

$$cent(i,c) = \sqrt{(ix - cx)^2 + (iy - cy)^2} \tag{4.6}$$

Once the ZH criteria are set and zones are formed, ZH is selected for each individual zone according to (6). Base station will have the collection list that will have ZH_{EV} of all nodes against each zone in the network. Node with maximum ZH_{EV} will be selected as ZH for that specific zone. The nominated ZHs announce their status to the network using advertisement message that contains its ID. Each node replies the join request to ZH. Each ZH builds a TDMA plan for its zone members to permit their communication. It specifies time slots during which zone members need to be in active state only when they are authorized to transmit the data.

The main advantage of having a collection list is to avoid any broadcast and communication of any maintenance messages during reselection process of ZH. The reselection process is decentralized where the base station is not involved. This approach significantly reduces the number of messages exchanged in the reselection process of ZH in a zone eventually reducing energy consumption and thus maximizing network lifetime. The lifetime of a ZH for one complete iteration is determined by a threshold valuediscussed in reselection phase.

(iv) Data Transmission Phase: Once nodes join ZH, it senses the environment continuously. Each non-ZH node switch ON their radio and transmit their sensed information to their respective head only when

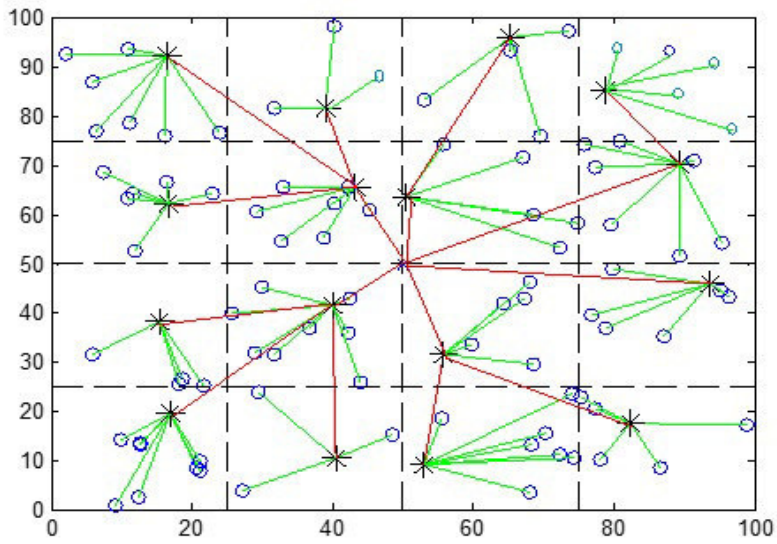


FIG. 5.1. *Zone formation and data transmission for ZEERP simulation network.*

an event is activated (i.e. current sensed value surpasses hard threshold value H(T)), Once H(T) value is attained, the subsequent transmission is feasible if variation in sensed value surpasses the soft threshold value S(T) (Mittal & Singh, 2015)-(Mittal, Singh, & Sohi, 2017). The nodes send their sensed data to the ZH as per their assigned TDMA schedule of transmission. This enables nodes to keep their radio off until its transmission time occurs. The sleep periods save node energy. In wakeup periods, ZHs will aggregate and compress the received data. ZHs transmit their aggregated information to BS using direct or dual-hop basis as a function of distance between them (Mittal & Singh, 2015), (Mittal, Singh, & Sohi, 2017).

(v) Reselection Phase: In this phase, the attention is to minimize the energy consumption in reselection process of ZH. Instead of carrying out periodic reselection of ZH that leads to extra energy depletion and network overhead, the protocol dynamically initiates the process of reselection based on the election value ZH_{EV} of the ZH. In a given iteration, if the ZH_{EV} value is less than or equal to average energy of the respective zone, the corresponding ZH will change. The number of iterations is independent of the zone and the reselection is carried out per zone when required. The number of iterations can be different for each zone to minimize the traffic generated in the network and also not to disturb the overall network. In order to select the new ZH in next round, the value of average zone energy is therefore periodically monitored by ZH. For this purpose every ZH maintains the collection list that contains ZH_{EV} and residual energy of nodes.

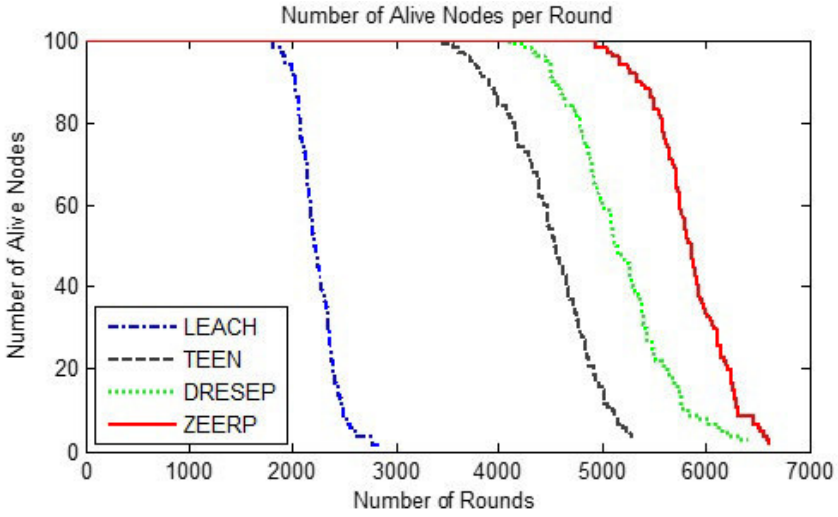
5. Performance analysis of ZEERP. The performance of proposed algorithm is evaluated by carrying out extensive simulations. All simulations were carried out using MATLAB R2015a. The performance of ZEERP is compared with several state-of-the-art energy efficient cluster-based protocols including LEACH,TEEN and DRESEP.

This research work is carried out to investigate the effects of proposed algorithm in homogeneous setup. 100 nodes are randomly deployed in a 100m × 100m area with an initial energy of E_0 $=1J$ as presented in Figure 5.1 Non-ZH nodes are represented by 'o' and ZHs are represented by '*'. The parameters setting for proposed protocol is given in Table 5.1.

The simulation results reveal that in ZEERP, total energy consumption during zone formation,ZH selection, reselection of ZH, and transmission has been reduced. To analyze the network lifespan against each method, various simulations were run for initial energy of 1 J for 100 nodes and packet size of 4000 bits. In Figure 5.2, it is

TABLE 5.1
Simulation parameters used for ZEERP.

Parameter	**Value**
Number of nodes	100
Network size	$100m \times 100m$
Location of BS	(50, 50)
Initial energy, E_0	$1\ J$
Grid Size	4×4
Radio electronics energy, $E_{Tx} = E_{Rx}$	$50\ nJ/bit$
Energy for data-aggregation, E_{DA}	$5\ nJ/bit$
Radio amplifier energy, ε_{friss_amp}	$100\ pJ/bit/m^2$
Radio amplifier energy, $\varepsilon_{two\_ray\_amp}$	$0.0013\ pJ/bit/m^4$
Temperature range on the field	$0℉ - 200℉$
Hard threshold	$50℉$
Soft threshold	$2℉$



6.

FIG. 5.2. *Network lifetime comparison for LEACH, TEEN, DRESEP and ZEERP.*

clear that ZEERP surpasses the competitive algorithms in terms of energy efficiency and prolong lifetime. This is because the control messages in ZEERP are reduced in ZH selection and reselection process. This increases the number of rounds and maximizes network lifetime. Table 5.2 shows the round history of dead nodes and average network lifetime in terms of number of rounds it takes until first node dies (FND), half of nodes die (HND) and last node dies (LND) for homogeneous setup. The lifetime comparison of ZEERP protocol with respect to FND, HND and LND is shown in Figure 5.3.

Figure 5.4 shows total energy consumed against the number of rounds. The graph shows that energy consumed by the proposed protocol is less than others with increase in number of rounds. This is due to the even distribution of nodes across the network resulting in steady energy consumption. In comparison with LEACH, TEEN and DRESEP, the proposed protocol maximizes the lifetime approximately by 225.14%, 12.12% and 2.12% respectively.

To evaluate the impact of grid size on the performance of ZEERP, whole network is partitioned into grid sizes of 4 × 4, 6 × 6, and 8 × 8 grids by keeping the same parameters such as number of nodes (100) with initial energy $E_0 = 1J$. In all three approaches, the proposed technique with grid size 8 × 8 has achieved maximum number of rounds thereby improving network lifetime as illustrated in Figure 5.5. This technique is

TABLE 5.2
Round history of dead nodes for simulated protocols

% dead Nodes	LEACH	TEEN	DRESEP	ZEERP
1 (FND)	1805.2	3518.3	4101.6	**4923.5**
10	2022.8	4048.3	4504.2	**5325.8**
20	2069.3	4228.9	4769.9	**5567.5**
30	2141	4379.2	4881.4	**5670.1**
40	2169.4	4460.7	4983.5	**5732.3**
50 (HND)	2215.2	4569.6	5126.7	**5851.4**
60	2280.1	4731.5	5294.2	**5914.2**
70	2346.3	4820.1	5395.4	**6066**
80	2394.8	4972.3	5621.1	**6198.3**
90	2485.6	5154.7	5770.7	**6291.2**
100 (LND)	2763.5	5388.2	6402.2	**6576.8**

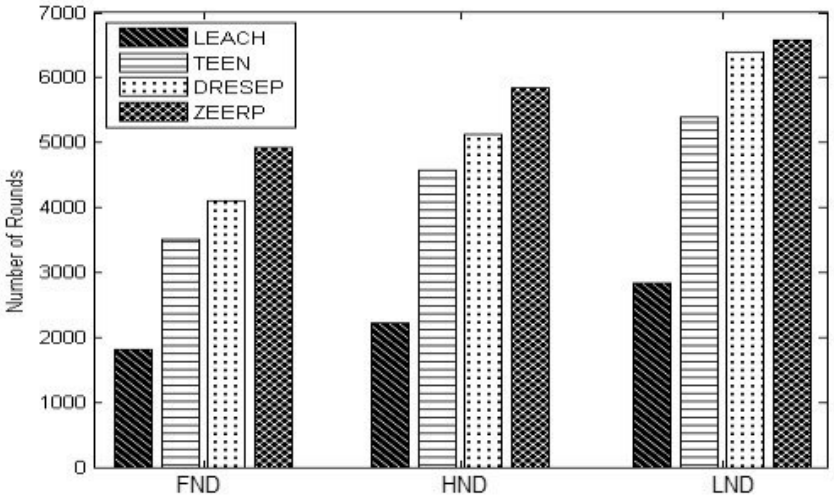


FIG. 5.3. *Lifetime comparison of simulated protocols for homogeneous set-up.*

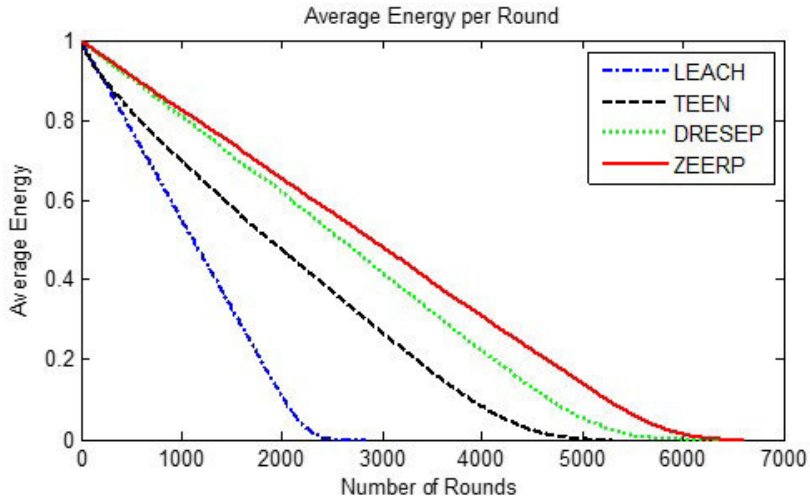


FIG. 5.4. *Average energy per round for simulated protocols.*

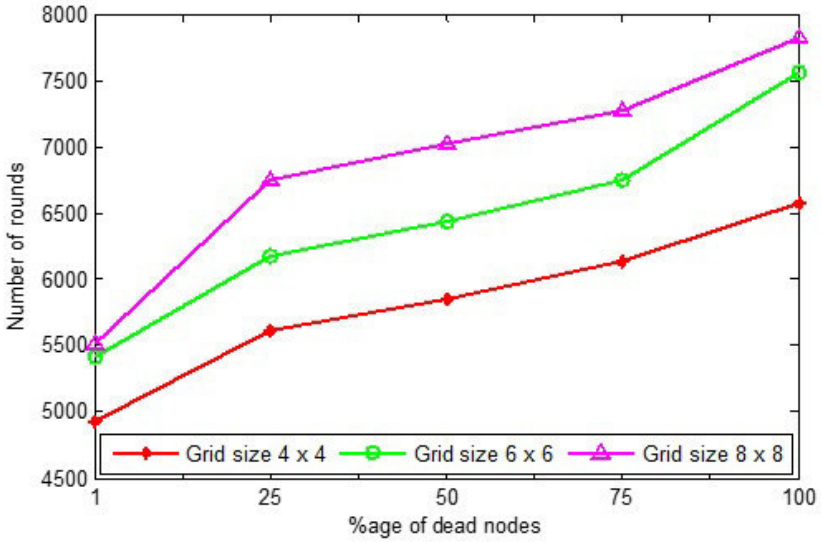


FIG. 5.5. *Effect of grid size on the performance of ZEERP*

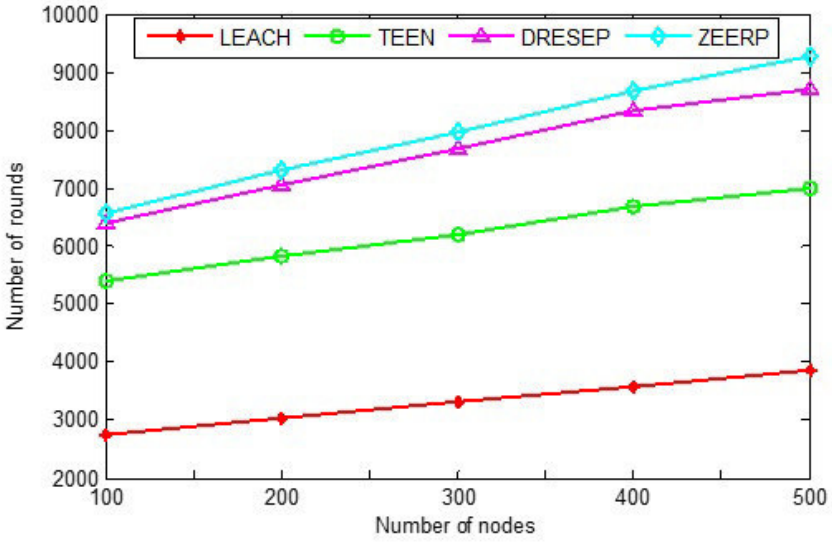


FIG. 5.6. *Effect of node density on the performance of ZEERP*

approximately 1.1 and 1.2 times better for grid sizes 4×4 and 6×6 respectively.

To evaluate the effect of node density in each protocol, the number of nodes is varied from 100 to 500 with initial energy 1 J as shown in Figure 5.6. The number of rounds increases by increasing the node density. ZEERP has more number of rounds than other approaches for varying node density. By increasing the node density, the resulting network has better lifetime as the responsibilities of each ZH is distributed.

The initial energy of sensor nodes is set to 0.25 J, 0.5 J, and 1.0 J for evaluating LEACH, TEEN, DRESEP and ZEERP to determine the number of rounds when 1%, 25%, 50%, 75%, and 100% nodes of the network die. Figures 5.7, 5.8 and 5.9 show that the proposed algorithm has larger number of rounds, this is because the control messages are reduced in ZH selection and reselection process. This increases the number of rounds and maximizes network lifetime.

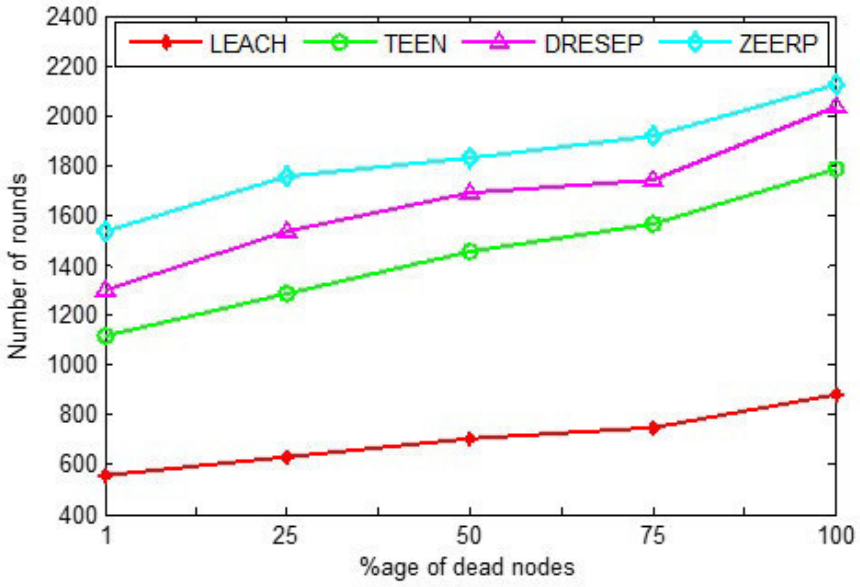


FIG. 5.7. *Number of rounds when 1with initial energy E 0= 0.25 J.*

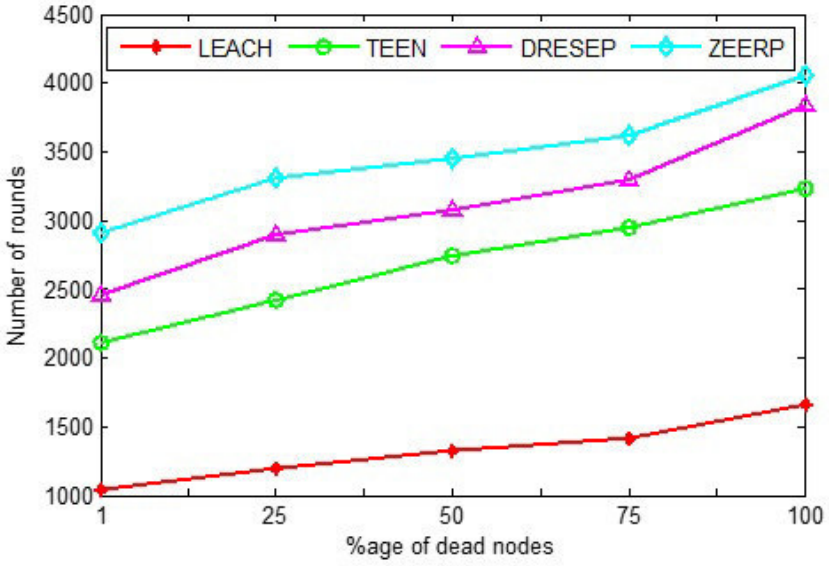


FIG. 5.8. *Number of rounds when 1%, 25%, 50%, 75%, and 100% nodes die in the network with initial energy E 0= 0.5 J.*

6. Stability-aware ZEERP. By a careful analysis of ZEERP, one can see that while this protocol succeeded in prolonging the WSN lifetime but is failed to ensure a longer stability (i.e., reliable) period until FND and reduced instability period (difference between LND and FND). For applications such as agricultural monitoring and smart environments, the protocol aims at maximizing the total lifetime of the application while minimizing the energy consumed by participating SNs. For crucial applications such as environmental monitoring, factory automation and security, the protocol aims at maximizing the stability period of the application

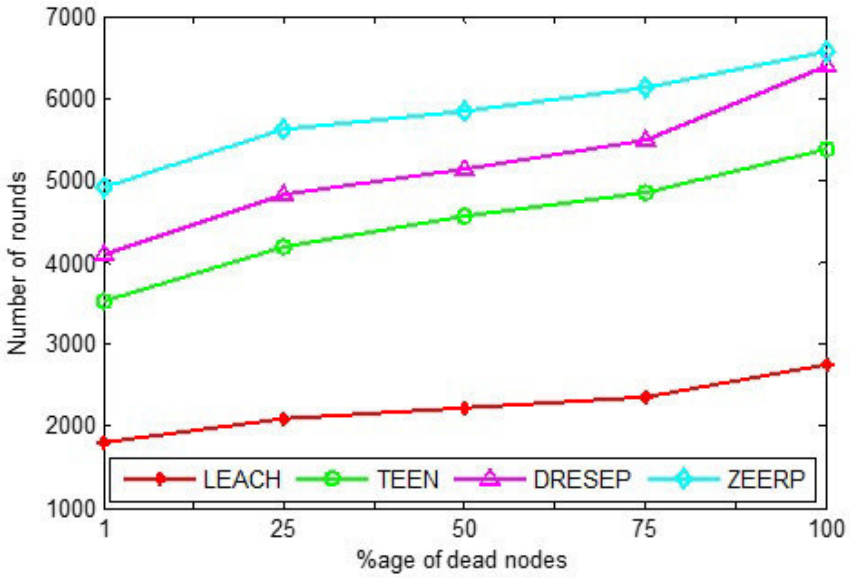


FIG. 5.9. *Number of rounds when 1%, 25%, 50%, 75%, and 100% nodes die in the network with initial energy E 0= 1 J.*

TABLE 7.1
Network lifetime of simulated protocols together with stability and instability periods

Protocol	**FND**	**HND**	**LND**	**Stability Period**	**Instability Period**
LEACH	1805.2	2215.2	2763.5	1805.2	958.3
DRESEP	4101.6	5126.7	6402.2	4101.6	2300.6
SEECP	5109.5	5143.6	5154.3	5109.5	**44.8**
ZEERP	4923.5	**5851.4**	**6576.8**	4923.5	1653.3
SZEERP	**5493.3**	5546.8	5573.2	**5493.3**	79.9

while minimizing the instability period. Usually, there is a tradeoff between stability and lifetime of the system. An optimal balance between these two contradictory goals is a challenge, and is essential to improve the overall network performance. To extend stability period and to balance the load effectively, another protocol named stable version of ZEERP (SZEERP) is proposed. The operation of the protocol is studied in terms of rounds, similar to ZEERP. The essential functioning of SZEERP is same as that of ZEERP. However, SZEERP uses residual energy as a parameter for ZH election (similar to SEECP) in selection and reselection phase for approaching robust performance. The deployment phase and zone formation phase follow the same procedure given in ZEERP. In SZEERP, ZH is selected on the basis of residual energy (R_E) but not on average distance value (A_D) as given in (6) for ZEERP. Similarly, in reselection phase for a given iteration, if R_E of ZH is less than or equal to average energy of the respective zone, the corresponding ZH will be replaced by highest energy node of that zone.

7. Performance analysis of SZEERP. Figure 7.1 shows SZEERP outperforms ZEERP by increasing the stability period for a considerably higher number of rounds. By restricting ZH election criteria to choose the highest energy node, the energy level of all nodes are uniformly preserved throughout the simulation. SZEERP minimizes the energy variance of the network and provides an indistinguishable flat variance plot as shown in Figure 7.2.

Table 7.1 presents the number of rounds taken for FND, HND and LND for simulated protocols together with stability and instability periods. Figure 7.3 shows that there is an improvement of 204.32%, 33.94%, 9.87%

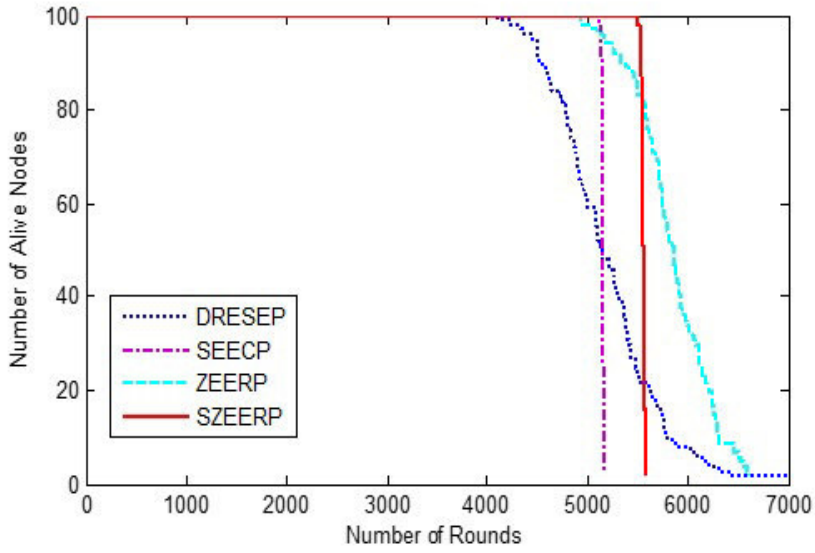


FIG. 7.1. *Number of alive nodes per round for simulated protocols.*

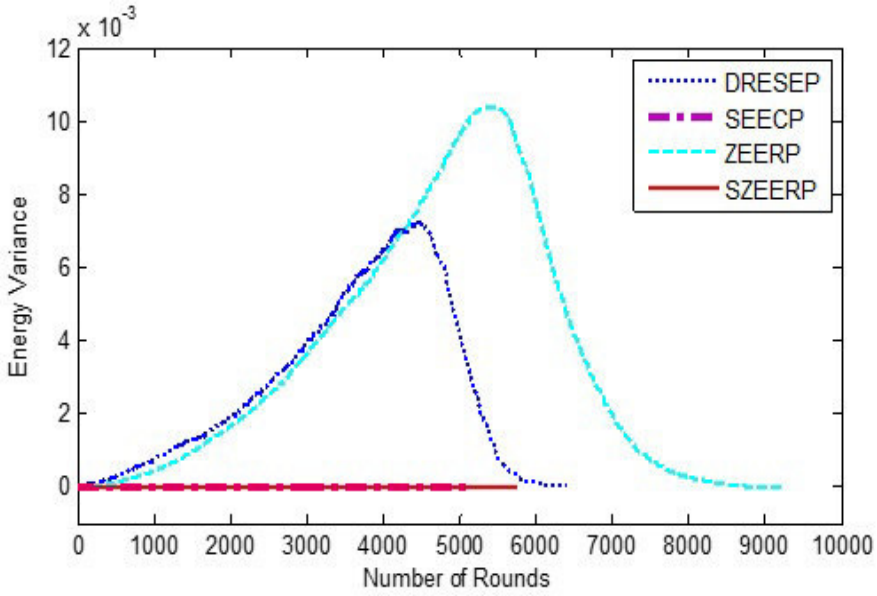


FIG. 7.2. *Energy variance per round for simulated protocols.*

and 11.57% in stability period for SZEERP as against LEACH, DRESEP, SEECP and ZEERP respectively. In addition, the instability period reduces to 96%, 98.33% and 98.77% in comparison with LEACH, DRESEP and ZEERP respectively. The results show that SZEERP is crucial for applications like military applications and health care applications in which each node is equally important and that require complete coverage of the network.

8. Conclusions and future scope. In this paper, a grid based randomly deployed sensor node network frame work is proposed that is well suited for time critical applications. The proposed approach evenly

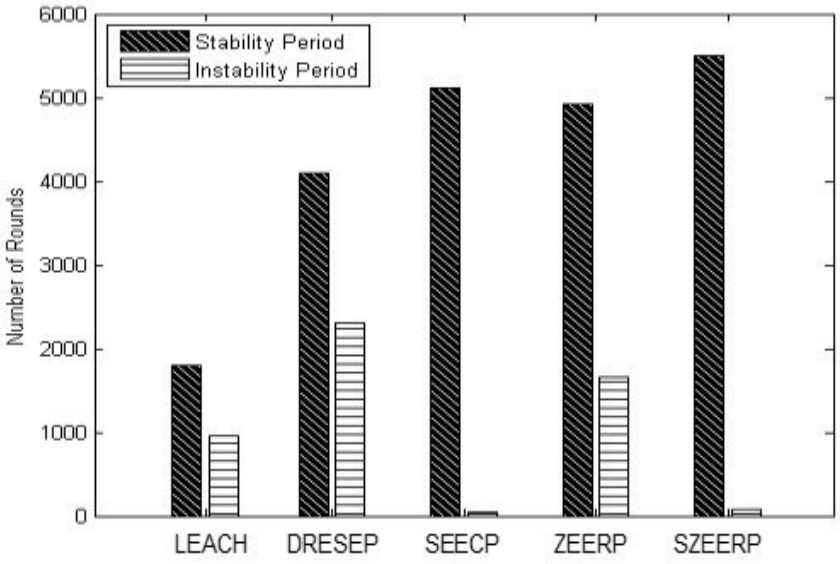


FIG. 7.3. *Performance results of simulated protocols for network lifetime.*

distributes the load across the network, improves network management, and extends network lifetime. The simulation results reveal that in ZEERP, total energy consumption during zone formation, ZH selection, reselection of ZH, and transmission has been reduced. In comparison with other existing protocols such as LEACH, TEEN and DRESEP, the proposed ZEERP protocol maximizes the lifetime approximately by 225.14%, 12.12% and 2.12% respectively. To extend stability period and to balance the load effectively, another protocolnamed stable version of ZEERP (SZEERP) is proposed. There is an improvement of 204.32%, 33.94%, 9.87% and 11.57%in the stability period for SZEERP in opposition to LEACH, DRESEP, SEECP and ZEERP respectively. In addition, the instability period for SZEERPs reduces to 96%, 98.33% and 98.77% in contrast to LEACH, DRESEP and ZEERP respectively. The proposed algorithms also outperform other approaches for parameters such as network lifetime, and total energy consumption.

The proposed algorithm opens a lot of gates for the future researchers. This work can be extended as an underlying topology for other energy efficient routing and load balancing protocols. In addition, further research is required to make the network framework adaptive by automatically optimizing number of grids, lower and upper bound for a specified number of senor nodes and terrain area. The future researchers may try their hands in improvising the selection and reselection process of Zone Head (ZH) in order to increase efficiency and to prolong lifetime of sensor network. It would be also interesting to see the impact of evolutionary algorithms in the proposed technique.

REFERENCES

[1] KIM, H. Y., An energy-efficient load balancing scheme to extend lifetime in wireless sensor networks. Cluster Computing, 19(1), 279283, 2016.

[2] SINGH, A. M., BHATIA, T., SHARMA, G., SHRIVASTAVA, G. "Artificial Intelligence Based Intrusion Detection System to Detect Flooding Attack in VANETs." In Handbook of Research on Network Forensics and Analysis Techniques, pp. 87-100. IGI Global, 2018.

[3] SHRIVASTAVA, G. KUMAR, P., GUPTA, B. B., BALA, S., DEY, N. Handbook of Research on Network Forensics and Analysis Techniques, 2018.

[4] GUPTA, B. B., JOSHI, R. C., MISRA, M., MEENA, D. L., SHRIVASTAVA, G., SHARMA, K. Detecting a wide range of flooding DDoS attacks using linear prediction model, In IEEE 2nd International Conference on Information and Multimedia Technology (ICIMT 2010), vol. 2, pp. 535-539.2010.

[5] ARZOO, M., BHATIA, T., SHARMA, G., SHRIVASTAVA, G. An Energy Efficient and Trust Aware Framework for Secure Routing

in LEACH for Wireless Sensor Networks. Scalable Computing: Practice and Experience 18, no. 3 (2017): 207-218.

[6] Chiti, F., Fantacci, R., Mastandrea, R., Rigazzi, G., Sarmiento, lvaro Surez, & Lpez, E. M. M. . A distributed clustering scheme with self nomination: proposal and application to critical monitoring. Wireless Networks, 21(1), 329345, 2015 https://doi.org/10.1007/s11276-014- 0785-z

[7] Younis, M., Senturk, I. F., Akkaya, K., Lee, S., & Senel, F. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. Computer Networks (Vol. 58). Elsevier B.V.. 2014 https://doi.org/10.1016/j.comnet.2013.08.021

[8] Liu, X., A survey on clustering routing protocols in wireless sensor networks. Sensors (Switzerland) (Vol. 12), 2012 https://doi.org/10.3390/s120811113

[9] Zaman, N., Low, T. J., & Alghamdi, T, Enhancing routing energy efficiency of Wireless Sensor Networks, 17th International Conference on Advanced Communication Technology (ICACT), 4(2), 587595, 2015 https://doi.org/10.1109/ICACT.2015.7224928

[10] Chang, J.-Y., & Ju, P.-H., An energy-saving routing architecture with a uniform clustering algorithm for wireless body sensor networks, Future Generation Computer Systems, 35, 128140, 2014 https://doi.org/10.1016/j.future.2013.09.012

[11] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H., Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol.1(c), 10, 2000, https://doi.org/10.1109/HICSS.2000.926982

[12] Manjeshwar, A., & Agrawal, D. P. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks, Proceedings 15th International Parallel and Distributed Processing Symposium, 0(C), 2001 https://doi.org/10.1109/IPDPS.2001.925197

[13] Sharma, T. P., Joshi, R. C., & Misra, M., GBDD: Grid Based Data Dissemination in Wireless Sensor Networks, 16th International Conference on Advanced Computing and Communications, 234240, 2008, https://doi.org/10.1109/ADCOM.2008.4760454

[14] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, 1(4), 660670, 2009, https://doi.org/10.1109/TWC.2002.804190

[15] Smaragdakis, G., Matta, I., & Bestavros, A., SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004), 111, 2004, https://doi.org/10.3923/jmcomm.2010.38.42

[16] Kumar, D., Aseri, T. C., & Patel, R. B., EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks, Computer Communications, 32(4), 662667, 2009 https://doi.org/10.1016/ j.comcom.2008.11.025

[17] Kumar, D., Aseri, T. C., & Patel, R. B., Multi-hop communication routing (MCR) protocol for heterogeneous wireless sensor networks. International Journal of Information Technology, Communications and Convergence, 1(2), 130145, 2011, https://doi.org/10.1504/IJITCC.2011.039281

[18] Mittal, N., & Singh, U., Distance-Based Residual Energy-Efficient Stable Election Protocol for WSNs. Arabian Journal for Science and Engineering, 40(6), 16371646, 2015, https://doi.org/10.1007/s13369-015-1641-x

[19] Mittal, N., Singh, U., & Sohi, B. S., A stable energy efficient clustering protocol for wireless sensor networks, Wireless Networks, 18091821, 2017, https://doi.org/10.1007/s11276- 016-1255-6

[20] Attea, B. A., & Khalil, E. A., A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks. Applied Soft Computing, 12(7), 19501957, 2012 https://doi.org/10.1016/ j.asoc.2011.04.007

[21] Mittal, N., Singh, U., Salgotra, R., & Sohi, B. S., A boolean spider monkey optimization based energy efficient clustering approach for WSNs. Wireless Networks, 117, 2017, https://doi.org/10.1007/s11276-017-1459-4

[22] Mittal, N., Singh, U., & Sohi, B. S., A Novel Energy Efficient Stable Clustering Approach for Wireless Sensor Networks, Wireless Personal Communications, 95(3), 29472971, 2017, https://doi.org/10.1007/ s11277-017-3973-1

[23] Mittal, N., Singh, U., Sohi, B. S., An Energy Aware Cluster-based Stable Protocol For Wireless Sensor Networks" Neural Computing and Applications (NCAA), pp 1-18, 2018.

[24] Zeng, J., A Clustering method of combining grid and genetic algorithm in wirless sensor netwroks, 216, 773779, 2013, https://doi.org/10.1007/978-1-4471-4856-2

[25] Pantazis, N. A., Nikolidakis, S. A., Vergados, D. D., & Member, S., Energy-Ef fi cient Routing Protocols in Wireless Sensor Networks: A Survey, 15(2), 551591, 2013.

[26] Chen, C., He, Z., Sun, H., Kuang, J., Bai, D., & Yang, C., A grid-based energy efficient routing protocol in Wireless Sensor Networks. International Symposium on Wireless and Pervasive Computing (ISWPC), 16, 2013, https://doi.org/10.1109/ISWPC.2013.6707444

[27] Ismail, W. Z. W., & Manaf, S. A., Study on Coverage in Wireless Sensor Network using Grid Based Strategy and Particle Swarm Optimization, (December), 69, 2010.

[28] Sharma, R., Sohi, B. S., & Mittal, N., Hierarchical Energy Efficient MAC protocol for Wireless Sensor Networks, International Journal of Applied Engineering Research, 12(24), pp 14727-14738, 2017.

[29] Sharma, R., Sohi, B. S., Kumar, Shakti., Ann based framework for energy efficient routing in multi-hop wsns, International Journal of Advanced Research in Computer Science, 8(5), 1298-1308, 2017.

[30] Sharma, R., & Sohi, B. S., A Compartitve Study on MAC Protocols for Wireless Sensor Networks on Energy Reduction. Internatioanl Journal of Computer Science and Information Security, 15(11), 3540, 2017

[31] Rajan Sharma, B. S. S., The Impact of Dynamic Scaling on Energy Consumption at Node Level in Wireless Sensor Networks, 13(1), 175188, 2018.

[32] Ojha, R. P., Sanyal, G., Srivastava, P. K., & Sharma, K., Design and Analysis of Modified SIQRS Model for Performance Study of Wireless Sensor Network. Scalable Computing: Practice and Experience, 18(3), 229241

https://doi.org/10.12694/scpe.v18i3.1303
[33] Amar Singh, Shakti Kumar, Sukhbir Singh Walia, FW-AODV : An Optimized AODV Routing Protocol for Wireless Mesh Networks, International Journal of Advanced Research in Computer Science, Volume, Volume 8, No. 3, March April 2017, pp. 1131-1135,2017.
[34] Shakti Kumar, S.S. Walia, Amar Singh, Parallel Big Bang-Big Crunch Algorithm, International Journal of Advanced Computing, ISSN:2051-0845, Vol.46, Issue. 3, September 2013, pp. 1330-1335, 2013
[35] Shakti Kumar, Amar Singh, Sukhbir Singh Walia , Parallel Big Bang - Big Crunch Global Optimization Algorithm: Performance and its Applications to routing in WMNs , Wireless Personal Communications, Springer, SCI Indexed Journa,l, pp. 1601- 1618, 2018.
[36] Sharma, K., Bala, S., Bansal, H., & Shrivastava, G., Introduction to the special issue on secure solutions for network in scalable computing. Scalable Computing, 18(3), 2017, https://doi.org/10.12694/scpe.v18i3.1299
[37] Amar Singh,Shakti Kumar, Sukhbir Singh Walia, P3PGA: Multi-Population 3 Parent Genetic Algorithm and its Application to Routing in WMNs, International Journal of Advanced Research in Computer Science, Volume, 8(5), pp. 968-975, 2017.
[38] Mittal, N., Singh, U., & Sohi, B. S, Harmony Search Algorithm Based Threshold-sensitive Energy-efficient Clustering Protocols For WSNs, Ad Hoc & Sensor Wireless Networks, 36 (1-4), pp 149-174, 2017

Edited by: Khaleel Ahmad
Received: Aug 2, 2018
Accepted: Jan 18, 2019

DOI 10.12694/scpe.v20i1.1464
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 71–82. http://www.scpe.org © 2019 SCPE

A PARTICLE SWARM OPTIMIZATION BASED LOAD SCHEDULING ALGORITHM IN CLOUD PLATFORM FOR WIRELESS SENSOR NETWORKS

ARVINDA KUSHWAHA* AND MOHD AMJAD†

Abstract. Integration of wireless sensor network into cloud computing is a growing paradigm that supports a massive amount of applications in cloud computing, optimization of resources required in the machines. This integration requires the optimization of resources to efficiently complete the different tasks in the devices at cloud platform. This optimization can be done using load scheduling algorithms. These algorithms reduce overload and achieve higher throughput by maximizing the machine utilization concerning cost stabilization. There are lots of methods like First Come First Serve, Min-Min, Particle Swarm Optimization (PSO) for optimizing the load but we use Particle Swarm Optimization as it obtains the motivation from the social behavior of the flock of birds and analyses various approaches for load scheduling. In this paper, we propose the load scheduling algorithm based on PSO in wireless sensor networks for cloud computing to minimize total transfer time and cost stabilization. The proposed method is compared with the existing approaches used for load scheduling in Cloudlets. It is clear from the simulation results that the proposed method is more efficient because it minimizes the transfer time and cost than the conventional algorithms thereby making a system for cost stable.

Key words: Wireless Sensor Networks, Particle Swarm Optimization, Load Scheduling, Cloudlets, Cloud Computing.

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction.

The Wireless Sensor Networks (WSNs) have seen significant growth in academia as well as the industry in the field of WSNs due to the advancement in the sensor's abilities like sensing power, computation, and communication capabilities, in last decade and so. WSNs possess different characteristics like energy constraints, fault tolerance, heterogeneity and homogeneity of nodes, deployment scalability, ease of use, and adaptability to sustain in extreme environmental conditions [1, 2]. In WSN, various sensors are spread in the target area for supervising and logging the physical conditions of the environment that have a small size, low processing power, less storage, and low energy abilities. These sensor nodes can sense the target and make an infrastructure-less wireless communication among them and base station (BS). Sensors accumulate the data from the target area and forward it to the BS directly or with the assistance of other sensors. The BS is connected to the cloud server with the help of wired/wireless links. Thus, the information is accessed by the users from the cloud servers [3]. The BS is supposed to be reliable and is capable of performing any operation. WSNs are being used in different applications like battlefield surveillance, data centre monitoring and data logging, health care supervision, forest fire detection, landslide detection, natural disaster prevention, water quality monitoring, structural health monitoring, environmental conditions such as extreme temperature variation, sound, pollution levels, humidity, wind, etc., industrial and consumer applications like monitoring of computer system health, industrial process supervision and control etc., and so on [4]. On the bases of the above-discussed applications, the wireless sensor node deployment can be categorized into two categories like deterministic and non-deterministic. In deterministic deployments, sensor nodes are placed into controlled manner or manually at the selected locations where the deployment area is physically accessible such as city sense monitoring, soil monitoring, etc. On the other hand, in non-deterministic deployments sensor nodes are deployed into physically inaccessible areas using other sources like sensors are dropped from an aircraft, e.g., battlefield surveillance and landslide detection, etc. The non-deterministic deployment is also called random deployment [5, 6].

Due to the widespread advent of the Internet of Things (IoT) which connects daily use objects such as mobile devices, Smart TVs, washing machines, Air conditioners, etc. to the internet so that an intelligent linking can be done. Therefore, the user of the devices can communicate with the tools as per his/her convenience being either at home or office. One of the essential parts of the Internet of Things paradigm is wireless sensor networks (WSNs). To expand the wireless services with the online user base, there is a need of efficient hosting of the aggregated data from WSN on the cloud which is very flexible and cost-effective solution, so that online user

*Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, ID (arvindakush@gmail.com)
†Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, ID ((amjad2k8@gmail.com)

can intelligently and efficiently communicate [7].

So, in this paper, we introduce a cost-efficient architecture for hosting the WSN data on the cloud platform. The architecture uses PSO based scheduling algorithm for load balancing among the servers of cloud so that the cost and time can be minimized.

This paper is framed in the following manner. The literature review of the integration techniques of wireless sensor networks and cloud computing is discussed in 2. A system model that includes the assumption for wireless sensor networks and cloud, network model is presented in 3. The proposed algorithmic method including flowchart is discussed in 4. The experimental setup and results are presented in 5. Finally,the paper provides the conclusions in 6.

2. Literature Review. In this section, we will discuss the literature review based on the integration of cloud platform and wireless sensor networks.

In paper [14], authors discuss an integrating framework for cloud paradigm and WSNs, which use data processability and cloud service model entirely. In the frame, data are efficiently utilized and managed to form the WSN by considering different information services provided to the users. In paper [15], an author discusses different types of scheduling algorithms and compares their results with various parameters. It also shows the many limitations of the existing algorithms such as they have not considered the execution time for calculating the performance of the method.

In paper [16], a trust assisted sensor cloud system is discussed which focuses on improving the quality of service of sensor cloud for users to obtain sensory data from the cloud. In faith supported sensor cloud system, trusted sensors which have trust values more significant than a threshold, collect and transmit sensory data to the cloud. Then the cloud selects the trusted data centers, which have trust values more significant than a threshold, to store, process the sensory data and further transmit the processed sensory data to users on demand. They show the trust assisted sensor cloud system can substantially improve the response time for users to obtain sensory data from the cloud system and compared to without faith supported sensor cloud system. In paper [17], the authors discussed wireless sensor network packet simulator, server application and web service, developed at the Institute Mihajlo Pupin for storing and viewing data collected by the network. The paper [18] discusses a novel architecture based on cloud computing for improving the performance of WSNs. In this architecture, a cloud platform acts as a virtual sink with the base station that collects data from sensors and processed in the distributed manner in the cloud system.

The paper [19] discusses an extensible and flexible architecture for integrating WSNs with the Cloud called REST. The REST-based Web services are used as an interoperable application layer that can be directly integrated into other application domains for remote monitoring. The paper [20] discusses a data processing framework for transmitting desired data to the mobile users. It decreases the storage requirements for sensor nodes and networks gateway and minimizes the traffic overhead and bandwidth requirement. However, the framework can able to predict the future trends of the sensory data and provides security for this sensory data.

The paper [23] analyzes the characteristics of job scheduling concerning integration cloud computing and wireless sensor networks and then discusses two popular job scheduling algorithms namely Min-Min and Max-Min. Two scheduling algorithms are proposed with the integration of cloud computing and wireless sensor networks by considering the Min-Min, and Max-Min and results of the proposed algorithms show shorter expected completion time than Min-Min and Max-Min, for cloud computing integrated with WSN. The paper [24] shows the optimal computing node searched in a limited time by proposing an ant colony algorithm. Three timers were used in setting back time for three role ants such as information and, positioning ant and task ant. When timer started, the general node of WSN was in power saving state. It takes a shorter time for the optimal computing resource. The paper [25] discusses location sleep scheduling algorithm with load balancing (LCLSS) scheme. It considers both awake and asleep status of sensors dynamically. It performs scheduling purely by mobile users current location. It reduces the power consumption and applies to remove areas applications where replacement of the battery is quite difficult. In the next section, we will discuss the system model and the assumptions for the integration of cloud computing and wireless sensor networks.

3. System Model. In this section, we discuss our system model by considering the following assumptions for the cloud networks:

- All nodes have similar capabilities, energy heterogeneity, location unaware, memory and computation constraints; a unique ID recognizes each node.
- The base station is stationarily located at the center in the network, and the sensor battery recharge or replacement is not possible.
- Each node has the capability to aggregate data, and the received signal strength can calculate the distance between nodes.
- The radio link consumes same energy in transmission and receiving information thus these links called symmetric connections.
- WSNs transmit gathered data to the cloud and end users require sensory data from the cloud.
- Multiple data centers are considered which consist of numerous data server in the cloud.

This network model consists of 3-types of classes based on their initial energy. In this network, WSN has n nodes out of which $\phi * n$ nodes have minimum energy, where a range of ϕ is $0 \leq \phi \geq 1$. We call these nodes Class-1 nodes and their energy is denoted as E_1. The $\phi^2 * n$ nodes have more energy than the Class-1 nodes, call these nodes Class-2 nodes and their energy is denoted as E_2. The remaining nodes have maximum energy called Class-3 nodes, i.e., (n-(ϕ*n+ϕ^2*n)) and E_3 denotes their energy. The sensor nodes have maximum energy considered to be minimum in numbers. The total energy of the network ($Total_{energy}$) is given below:

$$Total_{energy} = \phi * n * E_1 + \phi^2 * n * E_2 + (1 - \phi - \phi^2) * n * E_3 \tag{3.1}$$

This model describes a WSN, i.e., consisting of Class-1, Class-2, and Class-3 heterogeneity by considering the value of model parameter ϕ. The range of ϕ is between 0 and 1. When we put ϕ=0 in (3.1), it gives one non-zero term which contains only Class-3 nodes. It indicates the 1-level of heterogeneity, but these nodes have E_3 energy. In the case of 1-level heterogeneity, appropriate constraints are imposed to haveClass-1 nodes instead of the Class-3 nodes. It may be calculated by defining the following equation:

$$\phi = \frac{E_3 - E_1}{N * f(E_2, E_3)} \tag{3.2}$$

where N and f are the positive integer greater than 1 and the function of E_2 and E_3, respectively. The function has either (E_3+E_2) or (E_3-E_2). In (3.2), the value of ϕ should be in the consonant by considering the constraint: $E_1 < E_2 < E_3$.

The 2-level of heterogeneity contains two type of nodes, i.e., Class-1 and Class-2 for this we need to find the value of ϕ by considering the following equation:

$$1 - \phi - \phi^2 = 0 \tag{3.3}$$

Equation (3.3) is generated by (3.1) by considering the third term which generates the two level of heterogeneity. We call these nodes Class-1 and Class-2 nodes. Equation (3.3) has two solutions: ($(\sqrt{5})$-1)/2 and (($\sqrt{5}$)+1)/2. Since ϕ is upper-bounded by 1 and (($\sqrt{5}$)+1)/2>1, the valid solution of (3) is (($\sqrt{5}$)-1)/2. For ϕ =(($\sqrt{5}$)-1)/2, the model (1) consists of two types of nodes with energies E_1 and E_2.

In the case of 3-level heterogeneity, the range of ϕ is firstly determined where the upper limit is (($\sqrt{5}$)-1)/2. Let the lower limit of ϕ be ϕ_L that is to find out. The range of ϕ for 3-level heterogeneity is ϕ_L<ϕ<(($\sqrt{5}$)-1)/2. Taking f as (E_3-E_2) and ϕ from (3.2), we have

$$\phi_L < \frac{E_3 - E_1}{N * (E_3 - E_2)} < ((\sqrt{5}) - 1)/2 \tag{3.4}$$

Let E_2=α_1 +E_1 and E_3=α_2+E_2. From (3.4), we have

$$\frac{\alpha_2}{\alpha_1} < \frac{1}{n * \theta_L - 1} Or \frac{-\alpha_2}{\alpha_1} \geq \frac{1}{1 - n * \theta_L} \tag{3.5}$$

Since L.H.S. of inequality (5) is negative, we should have 1-N*ϕ_L < 0. This gives

$$\frac{1}{N} < \phi_L \tag{3.6}$$

The (3.4) can be written as

$$(E_3 - E_1) \leq N * \frac{(\sqrt{5}) - 1}{2} * (E_3 - E_2) \tag{3.7}$$

This inequality may be written as

$$N * ((\sqrt{5}) - 1) * E_2 - 2 * E_1 \leq (N * ((\sqrt{5}) - 1) - 2) * E_3 \tag{3.8}$$

This model describes WSNs which consists of three types of nodes, i.e., Class-1, Class-2, and Class-3. The total energy of the network (1) also describes the 1-level, 2-level, and 3-level of heterogeneity. In the next section, we will discuss the proposed clustering method for the heterogeneous network model.

4. Proposed Method. In this section, we discuss the proposed method which is divided into two parts namely energy efficient clustering protocols for WSNs and load scheduling algorithm based on PSO for cloud computing.

4.1. Energy Efficient Clustering Protocol For Sensor Networks. In this section, we consider an energy efficient heterogeneous DEEC protocol in WSNs [1]. The 3-level heterogeneous network model consists of three types of sensor nodes which are Class-1, Class-2, and hetDEEC-1, hetDEEC-2 denote Class-3 and its implementation, and hetDEEC-3, respectively. The cluster head selection of [1] assumes N* p_{opt} as the average number of cluster heads as in every iteration, and each sensor node assumes the responsibility of a cluster head once in every r_i=1/p_{opt} iteration, where p_{opt} is an initial probability of each sensor node, and r_i is iteration. The networks average energy E (r)can be calculated as follows:

$$E(r) = \frac{1}{N} \sum_{i=1}^{N} E_i(r) \tag{4.1}$$

where $E_i(r)$ is residual energy. The average probability of i^{th} node for becoming the cluster head during r^{th} iteration can be calculated as follows:

$$p_i = p_{opt}[1 - \frac{E(r) - E_i(r)}{E(r)}] = p_{opt} * \frac{E_i(r)}{E(r)} \tag{4.2}$$

The count of cluster heads per iteration is given by

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} p_{opt} * \frac{E_i(r)}{E(r)} = p_{opt} \sum_{i=1}^{N} \frac{E_i(r)}{E(r)} = N * p_{opt} \tag{4.3}$$

The cluster head in r^{th} iteration for the i^{th} node is given by (using (4.2))

$$r_i = \frac{1}{p_i} = \frac{E(r)}{p_{opt} * E_i(r)} = r_{opt} * \frac{E(r)}{E_i(r)} \tag{4.4}$$

As discussed in (3.1), the total energy at the beginning of the sensor network is $N*(\phi*E_1+\phi^{2}*E_2+(1\text{-}\phi\text{-}\phi^{2})*E_3)$ which is increased by $\phi+\phi^{2}*E_2/E_1$ +(1-ϕ-ϕ^{2})*E_3/E_1. Here, all nodes assume the responsibility of cluster head exactly once in every $\frac{1}{p_{opt}}$ *($\phi+\phi^{2}*E_2/E_1$ +(1-ϕ-ϕ^{2})*E_3/E_1)iteration. Thus, the average number of cluster heads per iteration is($\phi+\phi^{2}*E_2/E_1$ +(1-ϕ-ϕ^{2})*E_3/E_1)*N*$p_{class-1}$. In1-level of heterogeneity, each Class-1 node becomes a cluster head once in every ($\phi+\phi^{2}*E_2/E_1$ +(1-ϕ-ϕ^{2})*E_3/E_1) iteration, each Class-2 node becomes a cluster head (1+α) times more than the Class-1 nodes in every (($\phi+\phi^{2}*\frac{E_2}{E_1}$ +(1-ϕ-ϕ^{2})*$\frac{E_3}{E_1}$) iteration, and each Class-3 node becomes a cluster head (1+β) times more than the Class-1 nodes in every (($\phi+\phi^{2}*E_2/E_1$ +(1-ϕ-ϕ^{2})*E_3/E_1)) iteration. These methods mainly associate weights to obtain the optimal probability for each class of sensor node. It divides p_i by the factor of the total increased energy in the network for clustering

[1]. The weighted probabilities of the Class-1, Class-2, and Class-3 nodes denoted by $p_{class-1}$, $p_{class-2}$, and $p_{class-3}$, respectively, are given by

$$p_{class-1} = \frac{p_{opt} * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.5}$$

$$p_{class-2} = \frac{p_{opt} * (1 + \alpha) * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.6}$$

$$p_{class-3} = \frac{p_{opt} * (1 + \beta) * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.7}$$

The cluster head selection probability is calculated as follows.

$$\begin{aligned} T(s) &= \frac{p_{opt}}{1 - p_{opt}.(rmod * 1/p_{opt})} \;\; if s \in G \\ &= 0 \; otherwise \end{aligned} \tag{4.8a}$$

The thresholds T(s_i) Class-1, Class-2, and Class-3 nodes are given by

$$\begin{aligned} T(s_i) &= \frac{p_{class-1}}{1 - p_{class-1} * (rmod * 1/p_{class-1})} \;\; if p_{nrm} \in G^{'} \\ &= \frac{p_{class-2}}{1 - p_{class-2} * (rmod * 1/p_{class-2})} \;\; if p_{adv} \in G^{''} \\ &= \frac{p_{class-3}}{1 - p_{class-3} * (rmod * 1/p_{class-3})} \;\; if p_{sup} \in G^{'''} \\ &= 0 \; otherwise \end{aligned} \tag{4.9a}$$

where $G^{'}$, $G^{''}$ and $G^{'''}$ are set of Class-1, Class-2, and Class-3 nodes that have not become cluster heads within last $\frac{1}{p_{Class-1}}$,$\frac{1}{p_{Class-2}}$,and $\frac{1}{p_{Class-3}}$ rounds, respectively.

4.2. Load Scheduling Algorithm Based On Particle Swarm Optimization For Cloud Computing. In this subsection, we discuss the proposed scheme for load scheduling which is based on simple and basic particle swarm optimization [8, 9]. A new fitness function is incorporating in the proposed method. The proposed scheme defines the fitness function by the transfer time and costs with the higher exploitation of the space of the particles in the base station (BS). The data packets are initialized to the cloudlets and the virtual machines from the BS. The cloudlets are passed to the particles (N_p) to be allocated to the optimal virtual machines. The Cost(T) is specified as the total cost of all the tasks are assigned to calculate the resources available. The Cost(T) is calculated as follows:

$$C_{ex}(T)_j = \sum_k W_{kj} \tag{4.10}$$

$$\forall T(k) = j \tag{4.11}$$

$$C_{total}(T)_j = C_{ex}(T)_j + C_{tr}(T)_j \tag{4.12}$$

$$Cost(T) = \max(C_{total}(T)_j) \; \forall j \in P \tag{4.13}$$

$$\min(Cost(T) \; \forall T) \tag{4.14}$$

Here $C_{ex}(T)_j$, $C_{tr}(T)_j$, and $C_{total}(T)_j$ are the execution time, the transfer time employed in passing the particles from one resource to another, and gives the sum of the execution cost and the transfer cost between the tasks and the resources for the particle j, respectively. The transfer time is the maximum time taken by all the VMs to complete a particular task. The transfer time is calculated as:

$$E_{vmtime}(T)_k = \sum_k W_k \tag{4.15}$$

$$TransTime(T) = \max\left(E_{vmtime}(T)_k\right) \forall j \in N \tag{4.16}$$

$$k = 1, .., N \tag{4.17}$$

The new fitness function for both the cost and transfer time parameters is minimized by considering the weights in the system. The new fitness function is calculated as follows.

$$\alpha * (Cost(T)) + (1 - \alpha) * TransTime(T) \tag{4.18}$$

$$\min(\alpha(Cost(T)) + (1 - \alpha)TransTime(T)) \tag{4.19}$$

where α is the random parameter range from 0 to 1. It is used to shift the load scheduler to both the parameters, i.e., cost and transfer time giving the weightage to both the parameters of the packets in the base station.

The proposed fitness function minimizes weighted sum of transfer time and cost. If the value of $\alpha < 0.5$, higher weightage to the cost function otherwise, higher weightage to the transfer time. The ring-shaped clusters are employed in the neighborhood of the particle to obtain a new global best. The cumulative best values of the clusters help in moving the particles further to the next position.

These motivate in finding the best particles among all the particles, i.e., pbest(i,t) and global best among all the particles in the current iteration, i.e., gbest(t) values.The pbest value is computed as follows:

$$pbest(i,t) = arg \min_{k=1,....t} [f(P_i(k))], i \in 1, ...N_p \tag{4.20}$$

and gbest is known as the best position:

$$gbest(t) = arg \min_{i=1,....N_p;k=1,....t} [f(P_i(k))] \tag{4.21}$$

Here, i, N_p, f, P, t represent the index of the particle, a total number of particles, symbolically-represent the fitness function, position, and current round/iteration, respectively. The velocity and position of the particles are computed in the following manner.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1(pbest(i,t) - P_i(t)) + c_2 r_2(gbest(t) - P_i(t)) \tag{4.22}$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \tag{4.23}$$

where the velocity of the particle i at iteration t is denoted by $V_i(t)$, the velocity of the particle i at (t+1)round is denoted as $V_i(t+1)$. The c_1 and c_2 represent the acceleration coefficients of the system & r_1 and r_2 denote the random values in the range 0 to 1with ω denoting the inertia weight. The best location of the particle i is denoted by pbest(i,t) and the best position among all pbest(i,t) values is generated by gbest(t). The $P_i(t)$ specifies the current position of the particle i at iteration t and $P_i(t+1)$ denotes the position of the particle i at (t+1) iteration.On the basis of these values, the particles migrate to the next location $P_i(t+1)$. The positions of the VMs assigned to the particles are returned for further execution to the cloudlets in the cloud over the datacenter which further schedule the packets in the base station. These values are returned back to the base station.

The complete flow diagram of the proposed method is shown in Fig. 4.2.

Proposed Algorithm

1. **Assign** the packets from the base station to the cloud.
2. ***Assign*** the dimension of the particles in the search space.
3. ***Divide*** the region into ring-shaped clusters.
4. ***Randomly set*** position $P_i(t)$ and velocity $V_i(t)$ of particles from the set of resources (1 to j).
5. ***Compute the*** fitness function of the particles using the new fitness function *f*.
6. ***Substitute*** α with the random values [0, 1].
7. ***If*** $(\alpha < 0.5)$ then
8. Cost function dominates the transfer time
9. ***else***
10. Transfer time function dominates the cost
11. ***endif***
12. ***if*** *(current fitness value of the particle is better than the existing stored value),* then
13. ***Replace*** $pbest(i,t)$ by the new value.
14. ***endif***
15. ***Calculate*** $pbest(i,t)$ for all the particles.
16. ***Compute*** $gbest(t)$ as the best value from $pbest(i,t)$.
17. ***Generate*** velocity of the next particle $V_i(t+1)$ and next particle position $P_i(t+1)$.
18. ***Repeat process*** until the maximum numbers of iterations is met.
19. ***Return the scheduled values of the data packets to the base station.***

FIG. 4.1. *Proposed Algorithm.*

TABLE 5.1
Simulation parameters for the radio dissipation model and network model for wireless sensor networks [1,3,4]

Description	Value
Battery consumed to transmit at a shorter distance	10nJ/bit/m2
Battery consumed to transmit at a longer distance	0.0013pJ/bit/m4
Battery consumed to transmit or receive the signal	50nJ/bit
The battery used in data aggregation	5nJ/bit/signal
Threshold distance	70 m
Message Size	4000 bits
Network Size	100M X 100M
Base station Position	(50,50)
Maximum no. of Sensor Nodes	100
Cluster Radius	25M
Initial battery	0.50J
Constant N	10
1-level of heterogeneity: no of nodes and initial battery	100 and 0.5J
2-level of heterogeneity: no of nodes and initial battery of nodes	80 & 20 and 0.5J & 1.5J
3-level of heterogeneity: no of nodes and initial battery of nodes	51, 26, & 23 and 0.5J, 1.38J & 1.67J

5. Experimental Results And Discussions. The simulation considers the random deployment of 100 wireless sensor nodes in a field of size 100M X100M. It locates the base station at the center of the area, and we have borrowed a radio dissipation model from [1, 3, 4]. The model and associated input parameters are given in Table 5.1.

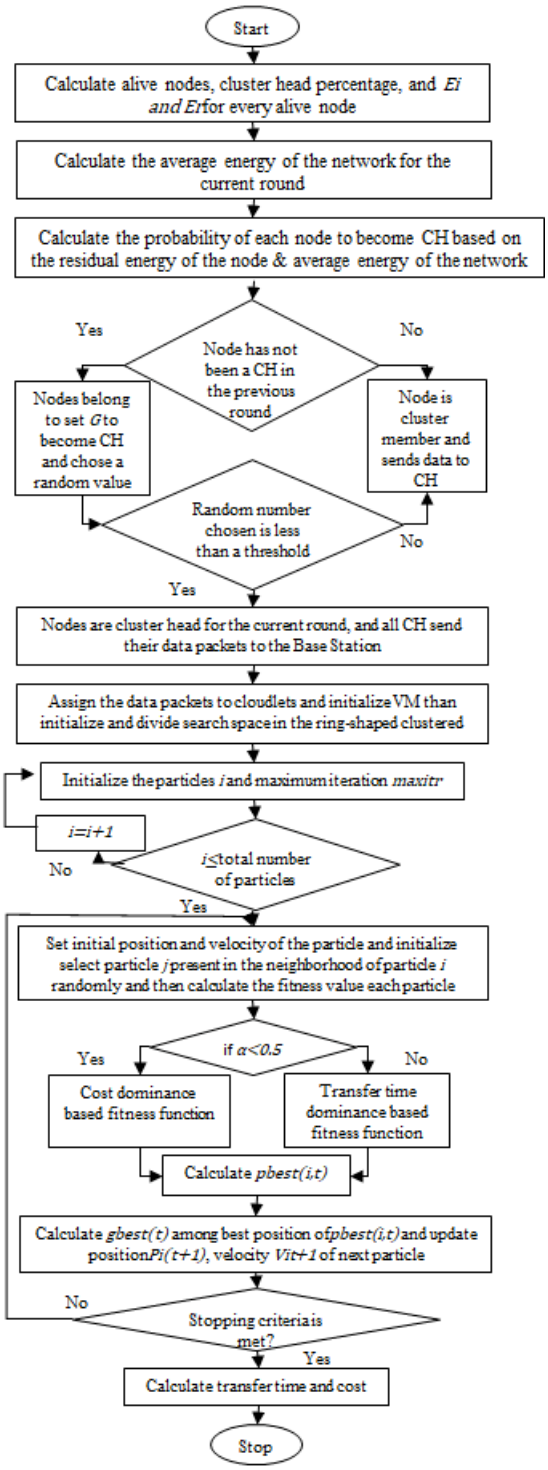


FIG. 4.2. *Flowchart of the proposed method.*

The proposed method and existing PSO algorithm [8, 9, 10, 11] are implemented in the CloudSim tool package. The CloudSim is used to perform, designing and analyzing the results and provides users built-in environment. It was given by Garg et al. in 2011, and Network CloudSim Simulator is simulated on top of the CloudSim tool package [7]. We consider two important matrices namely transfer time (seconds) and cost (dollars) for comparing the results of the proposed and existing approaches. The proposed method uses a JSwarm package to perform the simulation. The work in consideration takes ten particles referring to VMs. Each particle's resources include the following parameters like inertia, maximum & minimum position, and velocity. The dynamic allocation of resources depends on the number of iterations. The iterations taken into consideration are 10, 50, 100, 200, 500, and 1000 in figures. The values are provided before running to the system. The cloudlets and VMs possess the same capability given by the system such as MIPS, transfer cost, execution cost (which are used for the computation of the execution cost and transfer cost), and bandwidth. The parameters supplied to the simulator or the workload characteristics are MIPS (1000), ram (2048), bandwidth (10000), storage (10000) along with the number of iterations.

The results of the proposed method and existing approaches are analyzed by considering two matrices, i.e., transfer time and cost and compared. Six values of the proposed method are considered for the categorization namely, minimum value ($\alpha = 0$), average value ($\alpha = 0.3$), best value ($\alpha = 0.4$), mid or half value ($\alpha = 0.5$), random value (α= 0.7) and maximum value ($\alpha = 1$) values. These results are shown for a large set of iterations regarding transfer time and cost concerning the number of iterations.

The results are analyzed graphically as given in Fig. 5.1 to 5.4. Fig. 5.1 provides a precise depiction of the transfer time of the proposed approach at various α values, viz., $\alpha = 0$, $\alpha = 0.3$, $\alpha = 0.4$ (best value), respectively with existing ones PSO. Fig. 5.2 shows the current approaches with proposed method along with the various interval values regarding the total transfer time incurred over the number of iterations at $\alpha = 0.5$ (average value), $\alpha = 0.7$, $\alpha = 1$. Finally, it balances the system and decreases the cost of computation as well as the finish time.

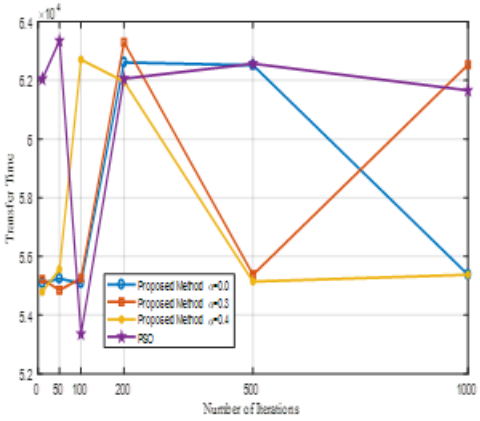


FIG. 5.1. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed a method by considering the different value of α=0.0,0.3,and 0.4.*

Fig. 5.3 provides a representation of the cost of the proposed method concerning many iterations at different α values, viz., $\alpha = 0$, $\alpha = 0.3$, $\alpha = 0.4$ (best value), respectively with existing ones PSO. Fig. 5.4 illustrates cost over the number of iterations at $\alpha = 0.5$ (Average Value), $\alpha = 0.7$, $\alpha = 1$. The cost analysis provides much better results for a large number of iterations. The average cost at all the iterations is quite similar. The proposed method gives a stable view of the cost incurred in the system thereby creating a stabilized system. Thus, the proposed approach produces better results regarding transfer time with a stable cost, i.e., no increase in cost. Fig. 5.3 and 5.4 give the total cost differentiation for a larger value of iterations showing maximum optimization. It explains that cost differentiation generates optimal results at various values and best result at $\alpha = 0.4$ (Best Value) and $\alpha = 0.5$ (Average Value). It shows that less cost is incurred in the proposed method than existing approaches.

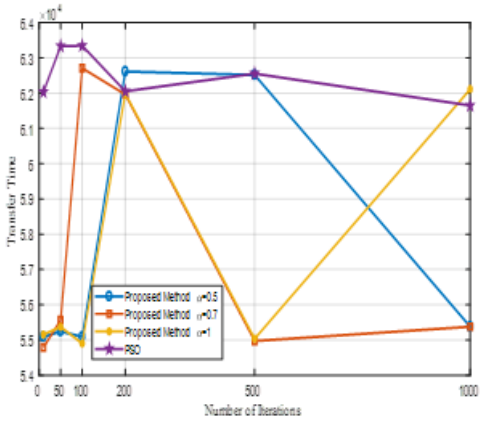


FIG. 5.2. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed the method by considering the different value of* α*=0.5,0.7,and 1*

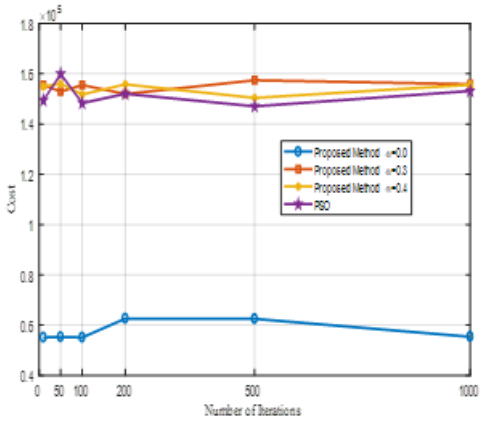


FIG. 5.3. *Shows the cost for the number of iteration for the existing PSO and proposed a method by considering the different value of* $\alpha = 0.0, 0.3$ *and 0.4*

It is evident from the Figs. 5.1 to 5.3 that the proposed method takes less transfer time concerning many iteration in the computation of the tasks. It leads to a further reduction in the total computational cost of the system. It produces much higher exploitation of the resources as compared to the existing approaches as evident from the figure. The figure signify the reduction in the total transfer time and total cost based on the proposed method. The cost and transfer time are the prime metrics to justify the storage, processing, and retrieval of data. Lesser transfer time leads to faster processing of user requests. Less cost specifies higher bandwidth and ram for more data processing. This proposed method provides more considerable significance in the real world scenario by optimal allocation, scheduling, and execution of the user requests on the virtual machines. Based on this the total cost incurred in the storage and processing of the data or information in the cloud is further reduced considerably by more considerable search space exploitation.

Table 5.2 shows the network lifetime regarding rounds by taking an equal number of nodes (i.e., 100) and the same amount of total network energy (i.e., 100J) 1-level, 2-level, 3-level heterogeneity. Categorization of nodes and their respective energies are given in details in Table 5.1. The number of rounds for 1-level (Class-1), 2-level (Class-2), 3-level (Class-3) heterogeneity are 2961, 3323, and 4404 in hours, respectively.

6. Conclusions. This paper discusses a load scheduling algorithm based on particle swarm optimization (PSO) in wireless sensor networks for cloud platform among the tasks. The primary objective of this method is to minimize transfer time and decrease the system cost. The total transfer time and the total cost are evaluated

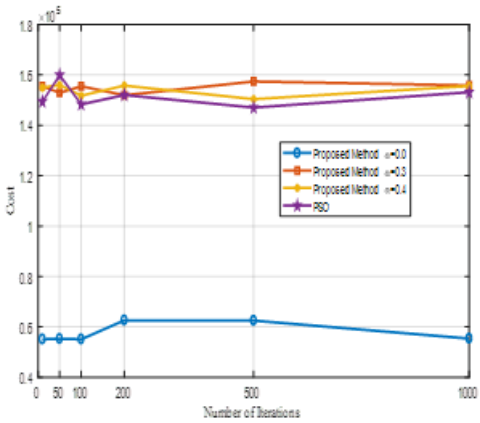


FIG. 5.4. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed a method by considering the different value of* $\alpha = 0.5, 0.7$ *and 1*

TABLE 5.2
Number of rounds for Class-1, Class-2, Class-3 using 100 number of nodes and 100 J network energy for 1-level, 2-level, and 3-level heterogeneity.

Classes	Nature of Networks	No. of Rounds (hrs)
Class-1	1-level heterogeneity	2961
Class-2	2-level heterogeneity	3323
Class-3	3-level heterogeneity	4404

by the computational values viz. the fitness value of the method. The proposed method produces better results and also provides additional benefits by making the system more stable regarding cost. The proposed method achieves higher cost optimization than the existing contemporary PSO based approaching terms of transfer time and thereby making the system more stable on the cost basis. The transfer time and the cost are reduced comparatively at $\alpha = 0.4$ (Best Value) as shown in the Figures. The future work aims at minimizing the price further to obtain greater feasibility of the results in a cloud computing environment.

REFERENCES

[1] S. Singh and A. Malik "hetDEEC: Heterogeneous DEEC Protocol for Prolonging Lifetime in Wireless Sensor Networks," Journal of Information and Optimization Sciences, vol. 38, no. 5, pp. 699-720, 2017.
[2] S. Singh, Energy Efficient Multilevel Network Model for Heterogeneous WSNs Int. Journal Engineering Science and Technology, vol. 20, no. 1, pp. 105-115, 2017.
[3] A. Malik and S. Singh "hetSEP: Heterogeneous SEP Protocol for Increasing Lifetime in WSNs," Journal of Information and Optimization Sciences, vol. 38, no. 5, pp. 721-743, 2017.
[4] S. Singh and A. Malik, Energy Efficient Scheduling Protocols for Heterogeneous WSNs International Journal of Forensic Computer Science, vol. 11, no. 1, pp. 8-29, 2016.
[5] A. Malik and S.Singh, "HeterogeneousEnergy Efficient Protocol for Enhancing the Lifetime in WSNs I.J. Information Technology and Computer Science, vol.8, no.9, pp. 62-72, 2016.
[6] S. Chand, S. Singh, and B. Kumar, "Multilevel Heterogeneous Network Model for Wireless Sensor Networks," Telecommunication Systems, vol. 64(2), pp. 259277, 2016.
[7] S.K. Garg, and R. Buyya, "Network CloudSim: Modelling Parallel Applications in Cloud Simulations," 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA), Melbourne, Australia, 2011.
[8] J Kennedy and R Eberhart, Particle swarm optimization. IEEE International Conference on Neural Networks, 4, 19421948, 1995.
[9] S. Pandey, and R Buyya et al., A Particle Swarm Optimization based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments.24th IEEE International Conference on Advanced Information Networking and Applications, 400-407, 2010.

[10] R.Buyya, S Pandey and Vecchiola, Cloudbus toolkit for market-oriented cloud computing.CloudCom 09: Proceedings of the 1st International Conference on Cloud Computing, 2009.
[11] P.Y. Yin, S.S. Yu, and Y.T. Wang, "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems" Computer Standards and Interfaces, 28(4), 441450, 2006.
[12] S. Singh, S. Chand, and B. Kumar, "Performance Evaluation of Distributed Protocols Using Different Levels of Heterogeneity Models in Wireless Sensor Networks," Int. Journal of Computer Network and Information Security, 7(1), pp.38-45, 2015.
[13] A K Sharma, and S. Singh, Distributed Algorithms for Maximizing Lifetime of WSN with Heterogeneity and Adjustable Range for Different Deployment Strategies I. J. Information Technology and Computer Science, 5(8), pp.101-108, 2013.
[14] P. You; Y. Peng; and H. GaoProviding Information Services for Wireless Sensor Networks through Cloud Computing IEEE Asia-PacificServices Computing Conference (APSCC), pp. 362 - 364, 6-8 Dec. 2012
[15] A.VenumadhavA Survey of Various Workflow Scheduling Algorithms in Cloud Environment International Journal of Scientific and Research Publications, Vol. 3, Issue 10, Oct. 2013.
[16] C. Zhu, V. C. M. Leung, L. T. Yang, L. Shu, J. J. P. C. Rodrigues, X. Li, "Trust Assistance in Sensor-Cloud" IEEE INFOCOM 2015, pp. 342-347, 2015.
[17] L. Kraus, M. Starcevic, M. Oklobdija and . Stojkovic, "Integrated Software Tools for Support of Wireless Sensor Network Applications", 19th Telecommunications forum TELFOR 2011 Serbia, Belgrade, pp. 1289-1292, November 22-24, 2011.
[18] P. Zhang, Z. Yan, and H. Sun, "A Novel Architecture Based on Cloud Computing for Wireless Sensor Network" Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), pp. 472-475, 2013.
[19] R. Piyare, S. Park, S. Y. Maeng, S. H. Park, S. C. Oh, Sang G. Choi, H. S. Choi, and S. R. Lee, "Integrating Wireless Sensor Network into Cloud Services for Real-time Data Collection" ICTC 2013, pp. 752-756, 2013.
[20] C.M Sukanya et al., " Integration of Wireless Sensor Networks and Mobile Cloud- a Survey" International Journal of Computer Science and Information Technologies, vol. 6, no. 1, 159-163, 2015.
[21] Y. Singh, S. Singh, and R. Kumar A Distributed Energy-Efficient Target Tracking Protocol for Three Level Heterogeneous Sensor Networks, International Journal of Computer Applications, 51(11), pp-31-36, Aug. 2012.
[22] A K Sharma, and S. Singh Distributed Energy-Efficient Algorithm for Wireless Sensor Networks, International Journal of Advanced Research in Computer Science, 2(3), pp-548-550, June 2011.
[23] C. Zhu, X. Li, V. C. M. Leung, X. Hu, and L. T. Yang, "Job Scheduling for Cloud Computing Integrated with Wireless Sensor Network," IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 62-69, 2014
[24] H. Yuan, C. Li, and M. Du, "Resource Scheduling of Cloud Computing for Node of Wireless Sensor Network Based on Ant Colony Algorithm". Journal Information Technology, vol. 11, pp.1638-1643, 2012.
[25] R. Priya, "collaborative location-based sleep scheduling with load balancing in sensor-cloud," International Journal of Electrical and Electronics Research, vol. 5, Issue 2, pp: (63-72), 2017.

Edited by: Khaleel Ahmad
Received: Nov 25, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1481
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 83–92. http://www.scpe.org © 2019 SCPE

NODE AUTHENTICATION USING NTRU ALGORITHM IN OPPORTUNISTIC NETWORK

MUSAEED ABOUAROEK* AND KHALEEL AHMAD †

Abstract. The demand for using wireless paradigms for performing various information and communication operations has been exploded. The opportunistic networks is a special type of delay tolerant networks proposed to operate in an emergency manner to facilitate mobile connectivity between the nodes when there is no connectivity. These emergencies are caused either by human-made or natural disasters. Opportunistic Networks depend on mobile phones and other mobile devices that carry wireless technology. This paper is an attempt to expand the opportunistic network through the authentication nodes. We propose an NTRU algorithm for node authentication in opportunistic networks .NTRU algorithm is an asymmetric post-quantum cryptosystem. This algorithm is unbreakable and robust compared to RSA and ECC cryptosystem.

Key words: Node Authentication, Packet Integrity, Sybil Attack, NTRU Algorithm, Post-Quantum Cryptography

AMS subject classifications. 68M12

1. Introduction. The opportunistic network is a subclass of mobile ad hoc network (MANET) that has been proposed to operate in an emergency manner where no network exists. OppNets are used to communicate between the nodes which may be mobiles or other devices having Wi-Fi or Bluetooth using personal laptops, cameras, sensors. OppNets may connect to other heterogeneous networks among the cellular base station, sensor networks, IoT devices, and other networks which connected through WI-FI [1, 2, 3, 4]. In this network the nodes act to each other as a router that causes Opportunistic networks to be more adaptable than Delay Tolerant Network (DTN) as shown in Fig. 1.1. OppNets use store-carry and forward mechanism to connect and extend the network because the path between the source and destination does not exist [5, 6, 7].

An OppNets grows from heterogeneous nodes and extend through authenticated nodes . The helper nodes need to join to opportunistic networks will check and verify to prevent and secure the network from the malicious node[7]. These nodes have self-configure and free to join/leave the OppNets. The use of OppNets is very suitable for disasters and emergency scenarios as they are infrastructure less or unavailable while the nodes can store - carry and forward the messages and the routes from the source to the destination are built dynamically [8, 9, 10]. The nodes in OppNets usually have high mobility, low density, limited power, short radio range, and often subject to different kinds of attacks by malicious nodes. Due to these characteristics, OppNets have gained significant research attention due to the security and authentication challenges that have emerged [11]. OppNets are very challenging network and the main challenging is to develop security solutions or discovering new algorithms to improve the connectivity between the nodes in a secure manner. Authentication is an important feature which must be implemented in each every network. Different researchers have done various efforts to explain authentication goals, node authentication, and packet authentication features in opportunistic networks [12, 13, 14].

2. Related Work. In 2018, Ahmed et al.presented a technique that allows nodes to authenticate packets as they receive them by constructing hash trees, also referred to as Merkle trees. Merkle trees are used to check and authenticate all the packets. As a result the direct trust is formed. Direct trust is updated based on the authenticity of the packets and the encounter rate of the node. As nodes come into contact with each other during the packet transmission period, they share feedback on how much they trust other nodes [11].

In 2018, Mehra et al. proposed codeword authenticated scheme in which the authentication between the entities takes place using three factor namely codeword, password and OTP. This protocol efficiently validates the entities preserving perfect forward secrecy and thwarting reply attack, DoS attack and man in the middle attack in hostile environment [15].

In 2017, Kumar et al. researchers proposed a security algorithm for opportunistic networks. The proposed algorithm utilizes dynamic IDs for key exchange mechanism and RSA for the message encryption

*Department of Computer Science and IT, Maulana Azad National Urdu University, India (musaeednaji@gmail.com)
†Department of Computer Science and IT, Maulana Azad National Urdu University, India (khaleelamna@gmail.com)

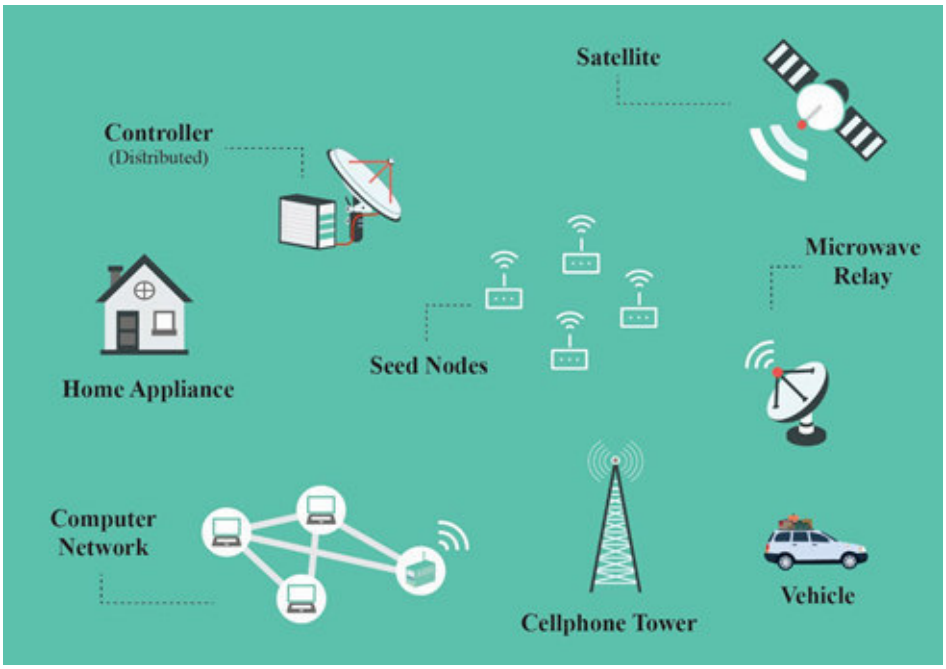


FIG. 1.1. *Heterogeneous nodes in an OppNet*

purposes and maintains the privacy of data as well as user privacy. In addition to this message integrity is also preserved. Results showed the proposed algorithm that fulfills the security requirements of opportunistic networks and perform well under various performance metrics [5] .

In 2017, Singh et al. proposed an authentication mechanism that uses a trust framework for opportunistic networks. They proposed some trust framework which provides trust of authenticated nodes to supernode, so that it can be an authentic node to register new node,as same as super node does. This mechanism is applied for the short-term and limited wireless network environment. They used a separate node to manage node registration and provide authorization to other authenticated nodes to register unauthentic nodes [16].

Wu et al. 2015 discussed a security architecture of opportunistic network where it is divided into five modules as authentication, access control, secure routing, trust management, cooperation and application user privacy [17].

In 2015, Guo et al. researchers proposed an authentication mechanism with privacy protection for opportunistic networks. It is applied for the short-term and limited-time wireless network environment, and a supernode is also set to manage node registration. The proposal implements some encryption and security technologies against security threats and attacks. In this analysis, the proposed mechanism finishes the authentication with less data, and provides anonymity and user privacy in the network. The present study proposes an OppNet-specific authentication scheme for preventing malicious node attacks and protecting personal privacy [18].

In 2014, Kuo et al. proposed an efficient and secure anonymous roaming authentication scheme for mobility networks. In order to maintain secure advantages, researchers proposed a scheme that utilizes hash functions and point operations instead of the asymmetric or symmetric system. Comparing security and performance, researchers proposed a scheme which not only has more security properties in comparison with previous schemes, but also enhances performance during the roaming authentication phase for mobility networks [19].

In 2010 Ma et al. proposed threshold secret sharing and identity- based cryptography were employed to facilitate opportunistic node authentication to secure data communication over intermittently connected mobile ad hoc networks(ICMANs). To avoid the key escrow problem and the single point of failure problem, the master private key of IBC was cooperatively generated and shared by n distributed PKGs, while each authenticating node has to encounter t-out-of-n PKGs to recover its own private key [20].

In 2014 Soleimani and Kahvand presented a dynamic trust model that used a trust to defend ad hoc networks from packet dropping attacks. At the initial stage of the network, a node trusted its surrounding nodes and updated the trust value according to their behavior. Behavior that decreased the trust value of a node included: dropping a packet, not forwarding a packet to the destination, not starting a route discovery phase. The trust value of a node increased when it forwarded the packets in a route targeting the destination. The nodes in the model do not propagate the trust values of the nodes in the network [21].

In 2015, Niaz and Saake present a deterministic method to protect the integrity of outsourced data using Merkle Hash Trees authentication. The method aims to deal with saving and loading the authenticated data from and to cloud service providers. The authors are aware of the difficulties faced with traditional databases to increased communication and computation overhead that result from adding and removing records, and aim to addresses in their ongoing work [22].

In 2007, Asokan et al. analyzed the applicability of IBC in this context and conclude that for authentication and integrity, IBC has no significant advantage over traditional cryptography, but it can indeed enable better ways of providing confidentiality. Additionally, the author showed a way of bootstrapping the needed security associations for IBC use from an existing authentication infrastructure [23].

3. Preliminaries. NTRU stands for Nth Degree Truncated Ring Units.NTRU encrypt is a public-key cryptosystem developed by three mathematicians named Hoffstein, Pipher and Silverman in 1998. NTRU is a ring-based cryptosystem. It has a ring R that consists of all truncated polynomials of degree N-1 having integer coefficients: a = $a_0 + a_1X + a_2X^2 + a_3X^3 + \ldots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}$. Polynomials are added in the usual way. They are also multiplied more-or-less as usual, except that X^N is replaced by 1, X^{N+1} is replaced by X, X^{N+2} is replaced by X^2, and so on [24, 25, 26, 27].

Notations:. P : Defines the highest degree of polynomial to be truncated.
q : q takes as a large modulus, usually, the coefficients of the truncated polynomials will be reduced mod q.
p : p is a small modulus. As the final step in decryption, the coefficients of the message are reduced mod p.
f(x): f(x) is a small polynomial and it is the part of the private key.
f_p : fp is the inverse of f(x) mod p and it is the part of the private key.
f_q : fq is the inverse of f(x) mod q.
g(x) : g(x) is a small polynomial, it is used to generate the public key.
P_U : P_U denotes the public key;P_k= p * (fq * g(x)) (mod q)
P_R : P_R denotes the private key.
m(x) : m(x) denotes the message, it takes as a small polynomial.
r(x) : r(x) denotes the random blind value, it is used during the encryption.
C : C denotes the ciphertext (encrypted message).
P_1 : The partially decrypted message P_1 = (f(x) * C) mod q.
P_2 : The partially decrypted message $P_2 = P_1$(mod p).
N_s : Seed Nodes (Authenticated Nodes)
N_h : Helper Node
N_{source} : Source Node (Sender)
N_d : Destination Node (Receiver)
D_R : Delivery Ratio
$M_{created}$: Messages created or generated by the sender
$M_{broadcast}$: Messages broadcast
$M_{received}$: Messages received by the receiver
$T_{created}$: Messages created or generated time
$T_{received}$: Messages received time

Set up of NTRU Algorithm.

i. **Key Generation**

- Choose two random polynomial f(x) and g(x) that is invertible by (mod p) and (mod q)
- Choose two random polynomial f(x) and g(x) that is invertible by (mod p) and (mod q)
- Choose two random polynomial f(x) and g(x) that is invertible by (mod p) and (mod q)

- Compute the inverse of f(x) mod p = fpand f(x) mod q= fq such that f(x) * f(x)-1 = 1 (mod p, q)
- The private key (PR) is the pair (f(x), fp)
- The public key (PU)= p * (fq * g(x)) (mod q)

ii. Encryption

- Choose a random polynomial r(x)
- Choose a message m(x).
- Compute the ciphertext (C)= r(x) * P_U + m(x) (mod q).

iii. Decryption

- Compute a polynomial using private key f(x) as P1 = f(x) * C(mod q)
- Compute a polynomial as P2 = P1 (mod p) and lessen the coefficients between p/2 and p/2.

4. System Model. In OppNets, the malicious node may modify or change the content of the messages [28, 29, 30]. To secure an opportunistic network from such type of attacks, we proposed an NTRU algorithm that shall eradicate the malicious nodes and useful in secure message communication. To perform this work, we purposed the NTRU algorithm which will provide an authentication to the node and the packet. The authenticated nodes (seed nodes) in an OppNet will generate the unique ID for each node to recognize them and generates the key to encrypt and decrypt messages to protect against advisory during transmission. Authenticated nodes will check the helper node which needs to join the network. If the node has an assigned ID then the authenticated node (seed node) will verify the ID but if the helper node is new then the seed node will generate the ID and authenticate it to join the network. In addition with, the seed node also broadcast the public key of helper node to update the public key list of each node. After ID allocation and verification the node is ready to carry and forward the messages as shown in Fig. 4.1.

4.1. Authenticated nodes. These are the main nodes on which opportunistic network based on and to ensure the security of the network where the authenticated nodes (seed nodes) register the new node to become an authenticated node. Its the backbone of the opportunistic network which generates ID to new node to authorize and connects to the network. After receiving the authorized ID for granted to enter the opportunistic networks and only authenticated nodes can carry and forward the messages to other nodes. Authenticated nodes can be generated and verify ID for new nodes which need to join the opportunistic network as a helper as shown in Fig. 4.2.

4.2. Unauthenticated nodes. These are the nodes which want to join to the opportunistic network without getting the registration ID or complete registration process, so these nodes are not authorized by authenticated nodes (seed nodes).

4.3. Mutual Node Identification and Authentication. When a new node needs to join the opportunistic network, it has to get the permission from the seed nodes or authenticated nodes. First, the authenticated node will receive a request from the node which needs to join, then will check and verify to know the node id is authorized or not, if node id is available in the public list of ids which is store in every authenticated node then will allow the node to join the opportunistic network. Second, if the node does not have any id so will send ACK to the authenticated node to register and response with new ID to join OppNets and broadcast new id to all authenticated nodes (seed nodes) to update the id list periodically. Now, this node is ready to join and communicate with any authenticated nodes and can generate an ID for new nodes. This is the main advantage to extend and grow the Opportunistic network.

4.4. Packet Authentication. The message, authenticated node, encrypts function and helper node must be in IDs list which must be authenticated, so these are the requirements to authorize the packet. When node n has the message for transmission, so it has to follow these steps to send the packet:

- Authenticated node or seed node will use the encrypt function by NTRU algorithm and generate two keys public key and private key to secure the packet
- Next node also called helper node will receive packet contents of destination id, keys of packet and message to carry and forward in an authenticated situation.
- Receiving node will receive the encrypted message and if its the destination node then will use the decryption function and forward the message to the next node till reach the destination.

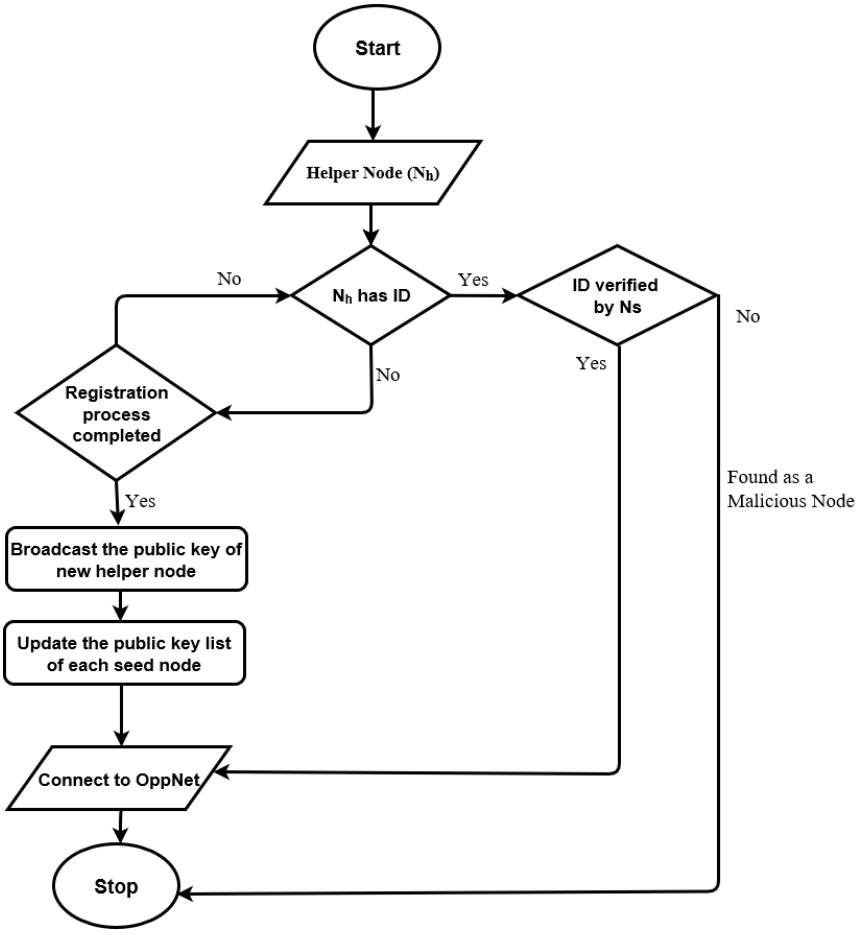


FIG. 4.1. *Flow Chart of Node Authentication Process*

4.5. Algorithm for Node Authentication Process.

```
BEGIN
  Nh is interested to join in OppNet, it sends a registration request to Ns
  If Nh has already an ID
        // If helper node is already registered in OppNet
        // then helper node has ID
       If ID is verified as authenticated by Ns // ID is verified by seed nodes.
           Connect to OppNets
       Else
           Authentication failed// Node found as a malicious.
       End If
  Else
      If the registration process is successfully completed by Ns
         Ns broadcast Public key (PU) of  Nh
         Update the public key list of each Ns
         Connect to  OppNets
```

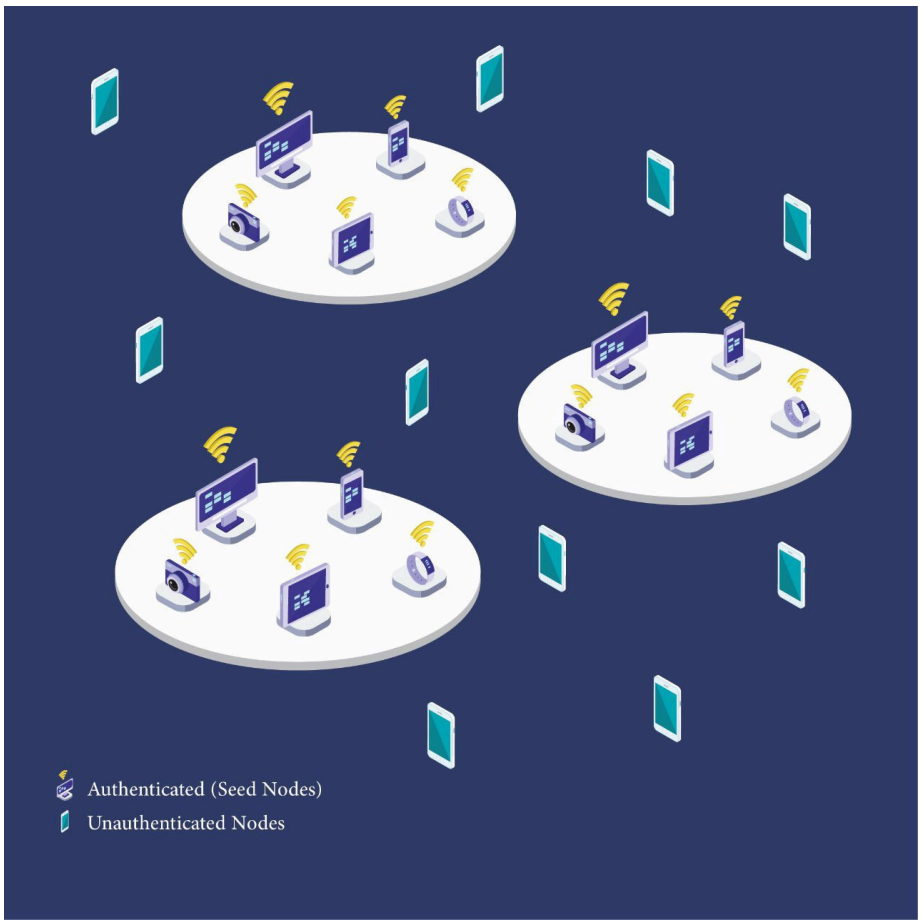


FIG. 4.2. *Node Authentication in Opportunistic Networks*

```
14       Else
15           Go to step 2 // re-initiate the process until the registration complete
16       End If
17   End If
17 END
```

4.6. Simulation of NTRU Algorithm. Suppose a node (N_{source}) wishes to send a message to a destination (Nd),N_{source} will encrypts the messages using receivers public key (P_U).

Consider the following Public Key Parameters values: $N = 11, p = 3$ and $q = 32$.

Choose the two polynomials randomly:

$f(x) = -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}$, $g(x) = -1 + x^2 + x^3 + x^5 - x^8 - x^{10}$

Consider the message which is in binary form 01001011 which is equivalent to $x^6 + x^3 + x^1 + 1$ in polynomial form: message $m(x) = 1 + x^1 + x^3 + x^6$

I. **Key Generation Process** Compute the inverse f_p of (f mod p)and the inverse $f_q$of (f mod q).

- $f_p = f(x)^{-1}(\text{mod p}) = (-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10})^{-1}(\text{mod } 3)$

```
In[7]:= fp = -1 + x + x^2 - x^4 + x^6 + x^9 - x^10;
        PolynomialMod[Algebra`PolynomialPowerMod`PolynomialPowerMod[fp, -1, x, x^11 - 1], 3]

Out[8]= 1 + 2 x + 2 x^3 + 2 x^4 + x^5 + 2 x^7 + x^8 + 2 x^9
```

$f_p = 1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + 2x^9$

- $f_q = f(x)^{-1} (\text{mod q}) = (-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10})^{-1} \ (\text{mod } 32)$

```
In[9]:= fq = -1 + x + x^2 - x^4 + x^6 + x^9 - x^10;
        PolynomialMod[Algebra`PolynomialPowerMod`PolynomialPowerMod[fq, -1, x, x^11 - 1], 32]

Out[10]= 5 + 9 x + 6 x^2 + 16 x^3 + 4 x^4 + 15 x^5 + 16 x^6 + 22 x^7 + 20 x^8 + 18 x^9 + 30 x^10
```

$f_q = 5 + 9x + 6x^2 + 16x^3 + 4x^4 + 15x^5 + 16x^6 + 22x^7 + 20x^8 + 18x^9 + 30x^{10}$

- Public Key (P_U)= p * (fq * g(x)) (mod q) Now, compute f_q * g(x) and apply the truncated concept:
  f_q* g(x)= $(5 + 9x + 6x^2 + 16x^3 + 4x^4 + 15x^5 + 16x^6 + 22x^7 + 20x^8 + 18x^9 + 30x^{10}) * (-1 + x^2 + x^3 + x^5 - x^8 - x^{10})$
  $= -5 - 9x - x^2 - 2x^3 + 11x^4 + 12x^5 + 13x^6 + 3x^7 + 22x^8 + 15x^9 + 16x^{10} + 29x^{11} + 60x^{12} + 19x^{13} - 2x^{14} - 7x^{15} - 36x^{16} - 40x^{17} - 50x^{18} - 18x^{19} - 30x^{20}$
  $= -5 - 9x - x^2 - 2x^3 + 11x^4 + 12x^5 + 13x^6 + 3x^7 + 22x^8 + 15x^9 + 16x^{10} + x^{11}(29 + 60x + 19x^2 - 2x^3 - 7x^4 - 36x^5 - 40x^6 - 50x^7 - 18x^8 - 30x^9)$
  $= 24 + 51x + 18x^2 - 4x^3 + 4x^4 - 24x^5 - 27x^6 - 47x^7 + 4x^8 - 15x^9 + 16x^{10}$
  $p * (f_q * g(x))$
  $= 3 * (24 + 51x + 18x^2 - 4x^3 + 4x^4 - 24x^5 - 27x^6 - 47x^7 + 4x^8 - 15x^9 + 16x^{10})$
  $= 72 + 153x + 54x^2 - 12x72x^5 - 81x^6 - 141x^7 + 12x^8 - 45x^9 + 48x^{10}$

```
In[24]:= PolynomialMod[72 + 153 x + 54 x^2 - 12 x^3 + 12 x^4 - 72 x^5 - 81 x^6 - 141 x^7 + 12 x^8 - 45 x^9 + 48 x^10, 32]

Out[24]= 8 + 25 x + 22 x^2 + 20 x^3 + 12 x^4 + 24 x^5 + 15 x^6 + 19 x^7 + 12 x^8 + 19 x^9 + 16 x^10
```

- Public Key (P_U) = $8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}$
- Private Key (P_R): f(x) = $-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}$
  f_p=$1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + 2x^9$

II. **Message Encryption Process** Now, sender computes the ciphertext (C) using receivers public key:
Message m(x) =$1 + x^1 + x^3 + x^6$
Choose Random Polynomial r(x) = $-1 + x^2 + x^3 + x^4 - x^5 - x^7$
$P_U = 8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}$
q = 32

- C= r(x) * P_U+ m(x) (mod q)
  Compute r(x) * P_U and apply the truncated concept: r(x) * P_U =$(-1 + x^2 + x^3 + x^4 - x^5 - x^7) * (8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10})$

```
>> conv([-1 0 1 1 1 -1 0 -1 0 0 0],[8 25 22 20 12 24 15 19 12 19 16])

ans =

    -8   -25   -14    13    43    35    14     7    -6     5   -14    23     4     8   -22   -28   -19   -16     0     0
```

$= -8 - 25x - 14x^2 + 13x^3 + 43x^4 + 35x^5 + 14x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10} + 23x^{11} + 4x^{12} + 8x^{13} - 22x^{14} - 28x^{15} - 19x^{16} - 16x^{17}$
$= -8 - 25x - 14x^2 + 13x^3 + 43x^4 + 35x^5 + 14x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10} + x^{11}(23 + 4x + 8x^2 - 22x^3 - 28x^4 - 19x^5 - 16x^6)$
$= 15 - 21x - 6x^2 - 9x^3 + 15x^4 + 16x^5 - 2x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10}$
r(x) *P_U+ m(x)
$= (15 - 21x - 6x^2 - 9x^3 + 15x^4 + 16x^5 - 2x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10}) + (1 + x^1 + x^3 + x^6$
$= 16 - 20x - 6x^2 - 8x^3 + 15x^4 + 16x^5 - x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10}$
r(x) * P_U+ m(x) (mod 32)
$16 - 20x - 6x^2 - 8x^3 + 15x^4 + 16x^5 - x^6 + 7x^7 - 6x^8 + 5x^9 - 14x^{10} (\text{mod } 32)$
$= 16 + 12x + 26x^2 + 24x^3 + 15x^4 + 16x^5 + 31x^6 + 7x^7 + 26x^8 + 5x^9 + 18x^{10}$

After encryption process, the ciphertext (C) is:

- $C = 16 + 12x + 26x^2 + 24x^3 + 15x^4 + 16x^5 + 31x^6 + 7x^7 + 26x^8 + 5x^9 + 18x^{10}$

III. **Message Decryption Process** The receiver (N_d) received the message (m) in an unreadable form which is known as ciphertext (C). Now, the receiver (N_d) decrypts the message using own private key (P_R):
$C = 16 + 12x + 26x^2 + 24x^3 + 15x^4 + 16x^5 + 31x^6 + 7x^7 + 26x^8 + 5x^9 + 18x^{10}$
Private Key (P_R):
f(x) $= -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}$
$f_p = 1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + 2x^9$
q = 32
p = 3

- P_1 = f(x) * C(mod q)
  Firstly computes f(x) * C and apply the truncated concept:
  f(x) * C= $(-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}) * (16 + 12x + 26x^2 + 24x^3 + 15x^4 + 16x^5 + 31x^6 + 7x^7 + 26x^8 + 5x^9 + 18x^{10})$

```
>> conv([-1 1 1 0 -1 0 1 0 0 1 -1],[16 12 26 24 15 16 31 7 26 5 18])

ans =

   -16     4     2    14    19    11   -10    28    23    52    -7    46    21    -7     9    20    -6    19   -21    13
```

$= -16 + 4x + 2x^2 + 14x^3 + 19x^4 + 11x^5 - 10x^6 + 28x^7 + 23x^8 + 52x^9 - 7x^{10} + 46x^{11} + 21x^{12} - 7x^{13} + 9x^{14} + 20x^{15} - 6x^{16} + 19x^{17} - 21x^{18} + 13x^{19} - 18x^{20}$
$= -16 + 4x + 2x^2 + 14x^3 + 19x^4 + 11x^5 - 10x^6 + 28x^7 + 23x^8 + 52x^9 - 7x^{10} + x^{11}(46 + 21x - 7x^2 + 9x^3 + 20x^4 - 6x^5 + 19x^6 - 21x^7 + 13x^8 - 18x^9)$
$= 30 + 25x - 5x^2 + 23x^3 + 39x^4 + 5x^5 + 9x^6 + 7x^7 + 36x^8 + 34x^9 - 7x^{10}$
f(x) * C(mod q), to reduce the coefficients between -q/2 and q/2:
$= (30 + 25x - 5x^2 + 23x^3 + 39x^4 + 5x^5 + 9x^6 + 7x^7 + 36x^8 + 34x^9 - 7x^{10})$ (mod 32)
$= 30 + 25x + 27x^2 + 23x^3 + 7x^4 + 5x^5 + 9x^6 + 7x^7 + 4x^8 + 2x^9 + 25x^{10}$
Choose the values lying between -q/2 and q/2 or between [-16,15].
$P_1 = -2 - 7x - 5x^2 - 9x^3 + 7x^4 + 5x^5 + 9x^6 + 7x^7 + 4x^8 + 2x^9 - 7x^{10}$

- $P_2 = P_1$(mod p)
  P_1(mod p), to reduce the coefficients between -p/2 and p/2:
  $= (-2 - 7x - 5x^2 - 9x^3 + 7x^4 + 5x^5 + 9x^6 + 7x^7 + 4x^8 + 2x^9 - 7x^{10})$(mod 3)
  $= 1 + 2x + x^2 + 0 + x^4 + 2x^5 + 0 + x^7 + x^8 + 2x^9 + 2x^{10}$
  $= 1 - x + x^2 + x^4 - x^5 + x^7 + x^8 - x^9 - x^{10}$
  $P_2 = 1 - x + x^2 + x^4 - x^5 + x^7 + x^8 - x^9 - x^{10}$

- Original Message m(x) = $f_p * P_2$(mod p)
  Now, compute$f_p * P_2$ and apply the truncated concept:
  $f_p * P_2$
  $= (1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + 2x^9) * (1 - x + x^2 + x^4 - x^5 + x^7 + x^8 - x^9 - x^{10}) = 1 + x - x^2 + 4x^3 + x^4 + 2x^5 - x^6 + 6x^7 + 2x^8 + 3x^9 - 3x^{10} + 6x^{11} + 0x^{12} - 2x^{13} - 3x^{14} + 2x^{15} + x^{16} - x^{17} - 3x^{18} - 2x^{19}$
  $= 1 + x - x^2 + 4x^3 + x^4 + 2x^5 - x^6 + 6x^7 + 2x^8 + 3x^9 - 3x^10 + x^{11}(6 - 2x^2 - 3x^3 + 2x^4 + x^5 - x^6 - 3x^7 - 2x^8)$
  $= (7 + x - 3x^2 + x^3 + 6x^4 + 3x^5 - 2x^6 + 3x^7 + 3x^9 - 3x^{10})$

m(x) = $f_p * P_2$ (modp)
$= (7 + x - 3x^2 + x^3 + 6x^4 + 3x^5 - 2x^6 + 3x^7 + 3x^9 - 3x^{10})$(mod 3)
$= 1 + x - 0 + x^3 + 0 + 0 + x^6 + 0 + 0 + 0$
$= 1 + x + x^3 + x^6$
$= x^6 + x^3 + x + 1$

Finally, the receiver (N_d) received the original Message:
$m(x) = x^6 + x^3 + x + 1 = 01001011$
$m(x) = 01001011$

5. Security Analysis. In this part of the paper, we analyze the security of our proposed algorithm, so the proposed algorithm is able to fulfill the security requirements.

Identification and Authentication. Before joining the network, any node has to check by authenticated nodes to verifying ID to join the Opportunistic Networks. Without unique I, the node is not able to join the OppNets and can't satisfy the identification. Before accepted any node to join or carried the message needs to apply the authentication technique, in which each authenticated node will check the ID before giving the task to transmit it, if ID is found in public list of IDs that mean the node is authenticated.

Confidentiality. During the registration process of each node to ensure that nodes are not a malicious node to protect the message transferred from source to destination are only based on authorized nodes. To protect the packet confidentiality would be encryption using cryptography to ensure that only authenticated nodes can know the key and carry the message.

Integrity. Before forwarding the message to the next node, the encryption techniques generates two keys, one public key and second private key to prevent any unauthorized nodes to modify or change the message. The integrity of the message refers to protect the packet from a malicious node or unauthorized nodes

Sybil Attack. In the case of a Sybil attack, the masquerade node tries to act as an authenticated node and create the multiple fake IDs which congest the network or stealing the confidential information. The proposed approach will check the ID of every helper node which comes into the network, if helper nodes have an ID then the proposed approach will check the ID that helper node is authentic or malicious. If a helper node does not have the ID then the approach will register the new helper node and allocate the unique ID. After successful registration, the seed node will broadcast the public key (PU) of a new helper node so that every seed node could recognize the helper node. The proposed approach prevents the Sybil attack.

6. Performance Analysis. To evaluate the performance of the proposed algorithm on the basis of the following three metrics:

Delivery Ratio (D_R):. It is the ratio of the number of messages received by the receiver in respect of the number of messages created (generated) at the sender side.

$$D_R = \frac{m_{received}}{m_{created}}$$

Latency. Latency is the average end-to-end delay between sender and receiver. The low latency depicts the better performance with respect to saving the network resources.

$$Latency = \frac{T_{received} - T_{created}}{Totalofm_{created}}$$

Routing Overhead. It is the ratio between the number of messages broadcast and the number of messages received. The ample routing overhead implies the more resources utilized.

$$RoutingOverhead = \frac{m_{broadcast}}{m_{received}}$$

7. Conclusion and Future Work. In this paper, we presented the security algorithm for opportunistic networks which provides node authentication to protect and prevent from Sybil attack, malicious and unauthorized nodes. This algorithm helps and improve authentication to extend opportunistic network from heterogeneous nodes, when new node needs to join it has to contact the seed nodes, then will check the node, if it has ID then will check and compare with public list which keep all nodes ID, if found then node become authorized, if ID not found then the seed nodes will generate new ID and broadcast it to update the public list periodically, so node can join after authorized. This algorithm features enhanced authentication in Opportunistic network environments, the proposed algorithm can prevent the Sybil attack and other nodes which are not authenticated. In the future, the authors will implement the proposed algorithm in the opportunistic environment on ONE simulator.

REFERENCES

[1] M. Goyal and M. Chaudhary, *Ensuring Privacy in opportunistic Network,* IOSR J. Comput. Eng. 13 (2), 74–82, 2013.
[2] J. Solis, P. Ginzboorg, and N. Asokan,*Best-Effort Authentication for Opportunistic Networks,* 2011.
[3] Y. Ma and A. Jamalipour, *Opportunistic node authentication in intermittently connected mobile ad hoc networks,* in 2010 16th Asia-Pacific Conference on Communications, APCC 2010, 2010, 453–457
[4] A. Seth and S. Keshav, *Practical security for disconnected nodes,* 2005 First Work. Secur. Netw. Protoc. NPSec, held conjunction with ICNP 2005 13th IEEE Int. Conf. Netw. Protoc., vol. 2005, 31–36, 2005.
[5] P. Kumar, N. Chauhan, and N. Chand, *Authentication with Privacy Preservation in Opportunistic Networks,* in International Conference on Inventive Communication and Computational Technologies, 2017, no. Icicct, 183–188
[6] J. L. Tsai and N. W. Lo, *Provably secure anonymous authentication with batch verification for mobile roaming services,* Ad Hoc Networks, vol. 44, 19–31, 2016.
[7] A. Albeshri, S. A. Chaudhry, and S. Kumari, *Cryptanalysis and improvement of a Multi-server Authentication protocol by,* vol. 12, no. 1, 523–549, 2018.
[8] P. Kumar, N. Chauhan, and N. Chand, *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications,* vol. 519, 465–471, 2018.
[9] R. Miller and W. Trappe, *ACE - Authenticating the Channel Estimation Process in Wireless Communication Systems,* 91–96.
[10] M. A. Shah, Sijing Zhang, C. Maple, and O. Shah,*A novel symmetric key cryptographic authentication for cooperative communication in cognitive radio networks,* 19th Int. Conf. Autom. Comput. (ICAC), 2013, no. September, 1–5, 2013.
[11] A. Ahmad, R. Doss, M. Alajeely, S. F. Al Rubeaai, and D. Ahmad, *Packet integrity defense mechanism in OppNets,* Comput. Secur., vol. 74, 71–93, 2018
[12] P. Raveneau, E. Chaput, R. Dhaou, and A. L. Beylot, *A multi-level FREAK DTN: Taking care of disconnected nodes in the IoT,* in 2016 7th International Conference on the Network of the Future, NOF 2016, 2017.
[13] A. T. A. Forwarding and S. M. Traces,*Data Forwarding In Opportunistic Network Using Mobile Traces,* In Data Forwarding In Opportunistic Network Using Mobile Traces, 2012, 425–430.
[14] S. Aluvala, K. Raja Sekhar, and D. Vodnala,*A novel technique for node authentication in mobile ad hoc networks,* Perspect. Sci., vol. 8, 680–682, 2016.
[15] P. S. Mehra, M. N. Doja, and B. Alam, *Codeword Authenticated Key Exchange (CAKE) light weight secure routing protocol for WSN*, International Journal of Communication Systems, 32 (2018).
[16] U. P. Singh and N. Chauhan, *Authentication using Trust Framework in Opportunistic Networks,* 2017.
[17] Y. Wu, Y. Zhao, M. Riguidel, G. Wang, and P. Yi, *Security and trust management in opportunistic networks: a survey,* Security and Communication Networks, vol. 8, no. 9, 1812–1827, 2015.
[18] M. Guo, H. Liaw, M. Chiu, and L. Tsai, *Authenticating with Privacy Protection in Opportunistic Networks Ming-Huang,* in EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE 2015), 2015, no. Qshine, 375–380.
[19] W. C. Kuo, H. J. Wei, and J. C. Cheng, *An efficient and secure anonymous mobility network authentication scheme,* J. Inf. Secur. Appl., vol. 19, no. 1, 18–24, 2014.
[20] Y. Ma and A. Jamalipour,*Opportunistic node authentication in intermittently connected mobile ad hoc networks,* in 16th Asia-Pacific Conference on Communications (APCC). IEEE, 2010, 453–457.
[21] Soleimani M, Kahvand M.*Defending packet dropping attacks based on dynamic trust model in wireless ad hoc networks.* In: Seventeenth IEEE Mediterranean Electrotechnical Conference (MELECON). 2014. p. 362–6.
[22] Niaz M, Saake G. Merkle*hash tree based techniques for data integrity of outsourced data.* In: The twenty seventh GI-workshop on foundations of databases. 2015. p. 66–71
[23] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, *Applicability of identity-based cryptography for disruption-tolerant networking,* in Procs. 1st international MobiSys workshop on Mobile opportunistic networking. ACM, 2007, 52–56.
[24] S. Bu and H. Zhang *"Research on the method of choosing parameters for NTRU,* 1st Int. Conf. Multimed. Inf. Netw. Secur. MINES 2009, vol. 2, no. 2, 334–337, 2009.
[25] R. Jha and A. Saini *A Comparative Analysis & Enhancement of NTRU Algorithm for Network Security and Performance Improvement* Commun. Syst. Netw. , 80–84, 2011.
[26] N. Zhao and S. Su,*An improvement and a new design of algorithms for seeking the inverse of an NTRU polynomial,* Proc. - 2011 7th Int. Conf. Comput. Intell. Secur. CIS 2011, 891–895, 2011.
[27] B. N. Hien,*An Overview of the NTRU Cryptographic System. 2014.*
[28] C. M. Huang, K. C. Lan, and C. Z. Tsai,*A survey of opportunistic networks,* in Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 2008, 1672–1677.
[29] M. Alajeely, R. Doss, and A. Ahmad,*Security and Trust in Opportunistic Networks – A Survey,* IETE Tech. Rev., vol. 33, no. 3, 256–268, 2016.
[30] R. M. Chaudhari, *A Survey on Attack Detection Techniques In Delay,* pp. 101–109.

Edited by: Rosilah Hassan
Received: Nov 25, 2018
Accepted: Jan 27, 2019

DOI 10.12694/scpe.v20i1.1440
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 93–99. http://www.scpe.org © 2019 SCPE

ON THE SECURITY OF AUTHENTICATED GROUP KEY AGREEMENT PROTOCOLS

SUMAN BALA*, GAURAV SHARMA†, HIMANI BANSAL‡, AND TARUNPREET BHATIA§

Abstract. The group key agreement protocol enables to derive a shared session key for the remote members to communicate securely. Recently, several attempts are made to utilize group key agreement protocols for s ecure multicasting in Internet of Things. This paper contributes to identify the security vulnerabilities in the existing protocols, to avoid them in future constructions. The protocols presented by Gupta and Biswas have been found insecure to ephemeral secret key leakage (ESL) attack and also, malicious insiders can impersonate an honest participant. Additionally, the protocol presented by Tan is also ESL-insecure. We also present a fix to the Tan's protocol to make it secure.

Key words: Group key agreement, Authentication, Insider security, Mutual authentication

AMS subject classifications. 68M12

1. Introduction. The recent security concerns are prevailing when multiple devices over a wireless communication interact among them leaking sensitive information to a non-participating entity [20, 19]. A common concern in these real world applications is to establish a secure session among the interested remote participants. It is always challenging to derive a shared secret key which can prevail over the advantages of adversary. A group key agreement (GKA) protocol enables the participating members to establish such a symmetric session key, usually following an asymmetric procedure. This symmetric key is further used for encryption decryption purpose. Various real life applications of GKA includes distributed computations, video conferencing and multi-user games. A variety of key establishment approaches have been presented in literature based on the network characteristics, agreement strategy, communication rounds and contributiveness. The two major classes representing the protocols are either transportation of session key or agreement via participant's contribution. In the key transport protocols, the session key is derived by one of the powerful nodes and then transferred securely to all the members of the group. The common session key, derived by all the members following an interactive protocol, is known as key agreement protocols.

The hybridization of above two categories can originate another variety of protocols namely, balanced and imbalanced protocols. The balanced protocols are equally contributive protocols while in imbalanced, all the participants contribute but the major part of computations, such as signature verification, is performed by some powerful node. Other than preserving the basic attributes such as known key security and forward secrecy, *contributiveness* is an important aspect of a GKA protocol. By contributiveness, we mean that all the member's contributions are involved so that none of the member can predetermine the session key without incorporation of other members.

Following the Diffie and Hellman [8] work on two-party key exchange, there has been extensive efforts to convert their two-party key exchange protocol to multi-party key exchange protocol [6, 11, 21]. Among the most notable works, Joux's one round three-party key agreement protocol [13] is considered as a significant contribution for practical GKA protocol due to the functionality of pairing. Based on Joux's work [13], Barua et al. [1] have presented protocols of multi-party key agreement in two flavours *unauthenticated-* based on ternary trees and *authenticated-* from bilinear maps. Unfortunately their protocols are secure against passive adversaries only. As established by Bellare and Rogaway (Crypto'93) [2], to avoid man in the middle (MITM) attack, *authentication* is an essential security requirement for key exchange protocols.

The first contribution towards modeling provable security for authenticated key exchange (AKE), was commenced by Bresson et al. [3, 4, 5] but their protocol accounts $O(n)$ rounds, which is very expensive. Later in 2003, Katz and Yung [15] presented a scalable compiler to transform any unauthenticated GKA into an authenticated GKA with the additional cost of one round. Since the GKA study involves multiple participants, the consideration of malicious insider is a realistic scenario. Katz and Shin [14] firstly modeled the *insider*

*Université Libre de Bruxelles, Belgium (suman1005@gmail.com)
†Université Libre de Bruxelles, Belgium
‡Jaypee Institute of Information Technology, Noida, India
§Thapar University, India

security in GKA protocols. Gorantla et al. [9] studied that the compromise of long-term key of one participant should not enable the impersonation of any other participant. The improved security model which addresses the *forward secrecy* and *key compromise impersonation resilience* (KCIR) for GKA protocols to take into account authenticated key exchange (AKE) security and mutual authentication (MA) security. In 2011, their model was revisited and enhanced by Zhao et al. [30] where they addressed the ephemeral secret key leakage (ESL) attack. The extended model is the strongest model, as it takes into account both the leakage of secret key as well as the leakage of ephemeral key independently. However, Tseng et al. [23] argued about the insufficiency of UF-ACMA secure signature scheme and proposed a UF-ACM-ESL secure signature based on Schnorr [17].

In *identity based* setting, the first authenticated ID-based GKA protocol was formalized by Choi et al.[7] in 2004, but their scheme was found vulnerable to insider colluding attack [29]. In 2007, Shim [16] claimed that scheme in [7] is vulnerable to another insider colluding attack and improved the protocol. Unfortunately, none of these AGKA protocols could achieve the perfect forward secrecy. Perfect forward secrecy allows the compromise of long term secret keys of all participants maintaining all earlier shared secrets unrevealed. In 2011, Wu et al. [28] presented a provably secure ID-AGKE protocol from pairings, providing forward secrecy and security against the insider attacks. Later, Wu et al. [27] presented their first revocable ID-based AGKE (RID-AGKE) protocol, which is provably secure and can resist malicious participants as well. The main attraction of this protocol was efficient revocation of group members. However, the protocol takes three rounds but unable to identify malicious participants. In a subsequent improvement, Wu et al. [26] proposed an ID-based AGKE protocol, which can passively detect malicious participants and also proved its security against insider attacks. Although, the protocol was later found insecure against an insider colluding attack by [24]. Afterwards, a two round revocable ID-AGKE protocol was presented by Wu et al. [25] which can identify malicious participants. Another work on authenticated group key agreement protocol without pairing is presented by Sharma et al. [18]. Recently in 2017, Gupta and Biswas [10] presented an ECCbased AGKA protocol and claimed it computationally efficient. However, in this paper, we present security flaws in their construction and proved it insecure. All the above discussed protocols are balanced GKA protocols where all the participants contribute equally and derive a shared session key.

On the other hand, some imbalanced GKA protocols are also presented where one of the powerful node contribute more in the computational sense. A recent contribution to improve the computational efficiency by Islam et al. [12] is presented. This is an ECC-based ID-AGKA protocol for imbalanced mobile networks. The best feature of this protocol is pairing-free property. However, Tan [22] found the Islam et al. [12] construction insecure and improved it. We present an ESL attack on their improved work and attempt to fix it.

Rest of the paper is organized as follows: in Section 2, we introduce necessary definitions, corresponding hardness assumption for AGKA protocol and standard security model for AGKA. Section 3 and Section 4 describes the AGKA protocols and our attacks on their construction, followed by the conclusion Section 5.

2. Preliminaries and Definitions. In this section, we introduce mathematical definitions, hardness assumptions, the notion of AGKA protocol and security model for it. If X is a set, then $y \xleftarrow{\$} X$ denotes the operation of choosing an element y of X according to the uniform random distribution on X.

2.1. Notations Used. This section describes the preliminaries used for AGKA protocol. Table 2.1 shows the notations used throughout the paper.

2.2. Definitions and assumptions. Definition 2.1 (Computational Diffie-Hellman Problem (CDHP)). Let $\mathbb{G}$ be an additive cyclic group (precisely an elliptic curve group) of order q with generator P. Let $CDH : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ be a map defined by

$$CDH(X, Y) = Z, \text{ where } X = aP, Y = bP \text{ and } Z = abP\,.$$

The *computational Diffie-Hellman problem* (CDHP) is to evaluate $CDH(X, Y)$ given $X, Y \xleftarrow{\$} \mathbb{G}$ without the knowledge of $a, b \in \mathbb{Z}_q^*$. (Note that obtaining $a \in \mathbb{Z}_q^*$, given $P, X \in \mathbb{G}$ is solving the elliptic curve discrete logarithm problem (ECDLP).)

Definition 2.2 (Computational Diffie-Hellman Assumption). Given a security parameter λ, let $\langle q, \mathbb{G}, P, X, Y, \rangle \leftarrow \mathfrak{G}(\lambda)$. The *computational Diffie-Hellman assumption* (CDHA) states that for any PPT algorithm $\mathcal{A}$ which

TABLE 2.1
Notations Used

Notation	Description
q	a large prime number
$\mathbb{F}_q$	finite field
$\mathbb{E}/\mathbb{F}_q$	elliptic curve defined on $\mathbb{F}_q$
$\mathbb{G}, \mathbb{G}_1$	cyclic additive group composed of the points on $\mathbb{E}/\mathbb{F}_q$
$\mathbb{G}_2$	cyclic multiplicative group composed of the points on $\mathbb{E}/\mathbb{F}_q$
P	generator of $\mathbb{G}$
$\hat{e}$	admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$
$H_i(\cdot)(1 \leq i \leq n)$	secure one-way hash functions
k	security parameter
$param$	system parameters
U_i	A mobile node
U_n	The powerful node
ID_i	Identity of node $U_i(1 \leq i \leq n)$
n	number of participants
$\mathcal{C}$	Challenger, who is authoritative to respond adversary's query
$\mathcal{A}$	Adversary

attempts to solve CDHP, its *advantage*

$$\mathbf{Adv}_{\mathfrak{G}}(\mathcal{A}) := Prob[\mathcal{A}(q, \mathbb{G}, P, X, Y) = CDH(X, Y)]$$

is negligible in λ. We say that the (t, ϵ)-*CDH assumption* holds in group $\mathbb{G}$ if there is no algorithm which takes at most t running time and can solve CDHP with at least a non-negligible advantage ϵ.

2.3. AGKA Protocol. Let there are total n participants $U_1, U_2, \cdots, U_n$ and any subset with $(n \geq 2)$ can run the protocol (π). Each participant is provided a (public, private) key pair. In a protocol, we refer by *session* a running instance. Each participant is allowed to run multiple sessions concurrently. An i^{th} instance of the protocol is represented as Π_U^i, where U is the corresponding user or participant. We define two identities - the session identity sid_U^i which is the session dependent information computed by user U at it's i^{th} instance using the shared information in that session, and the partner identity pid_U^i which is a set of identities of the participants who are involved in generation of the session key with Π_U^i . We say an instance Π_U^i *accepts* when it computes a valid session key sk. We say instances Π_U^i and $\Pi_{U'}^j$(for $\Pi_U^i \neq \Pi_{U'}^j$) are partnered iff (i) they have both accepted (ii) $sid_U^i = sid_{U'}^j$ (iii) $pid_U^i = pid_{U'}^j$. We further define the term *freshness*.

DEFINITION 2.3 (Freshness). *An instance Π_U^i is referred to be fresh if it satisfies the following conditions:*

1. *If the instance Π_U^i is accepted, neither U_i nor any of its partnered instances, can query Reveal key oracle.*
2. *No participant is allowed to query Corrupt and Reveal Ephemeral Key simultaneously.*
3. *In a partnered instance between U_i and U_j, if an adversary $\mathcal{A}$ corrupts U_j, any message sent from U_j to U_i must actually come from U_j.*

2.4. Security Model for AGKA Protocol. We analyze the security of proposed protocol within the standard security frame of indistinguishability. For the purpose we define the following experiment between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$:

Setup: On input a security parameter 1^λ, the challenger $\mathcal{C}$ runs $\mathbf{KeyGen}(1^\lambda)$ to generate the public parameter *Params* and the system key pair (pk, msk) and gives the adversary $\mathcal{A}$ the public key pk. msk is the master secret of the system.

Queries: $\mathcal{A}$ can adaptively make the following queries:

- Execute(Π_U^i): Any time the adversary $\mathcal{A}$ can query for the complete transcripts of an honest execution among the users selected by himself.
- Send(Π_U^i, m): During the normal execution of the protocol, this query returns the reply generated by instance Π_U^i.
- Reveal Key (Π_U^i): When the oracle is accepted, this query outputs the group session key.

- Corrupt(U_i): This query models the reveal of long-term secret key. The participant is honest iff adversary $\mathcal{A}$ has not made any *Corrupt* query.
- Ephemeral Key Reveal(Π_U^i): This query models the reveal of ephemeral key of participant U_i for instance Π_U^i.
- Test(Π_U^i): This query can be made only once during the execution of protocol π. The challenger responds with a session key.

Challenge: During the Test query, the challenger randomly selects a bit $b \xleftarrow{\$} \{0,1\}$ and returns the real session key if $b = 0$ or a random value if $b = 1$.

Guess: $\mathcal{A}$ outputs its guess b' for b.

The adversary succeeds in breaking the security if $b' = b$. We denote this event by $Succ_{\mathcal{A}}$ and define $\mathcal{A}$'s advantage as $Adv_{\mathcal{A}}(1^k) \stackrel{\text{def}}{=} |2Pr[Succ_{\mathcal{A}}] - 1|$.

DEFINITION 2.4 (AKE-Security). *Let $\mathcal{A}_{ake}$ be an adversary against AKE-security. It is allowed to make queries to the Execute, Send, RevealKey, Ephemeral Key Reveal, Corrupt oracles. It is allowed to make a single Test query to the instance Π_U^i at the end of the phase and given the challenge session key $sk_{ch,b}$ (depending on bit b). Finally $\mathcal{A}_{ake}$ outputs a bit b' and wins the game if (1)$b = b'$ and (2) the instance Π_U^i is fresh till the end of the game. The advantage of $\mathcal{A}_{ake}$ is $Adv_{\mathcal{A}_{ake}} = |2Pr[Succ_{\mathcal{A}_{ake}}] - 1|$. The protocol is called AKE-secure if the adversary's advantage $Adv_{\mathcal{A}_{ake}}$ is negligible.* Below we recall the MA-security considering both types of adversaries, outsiders and insiders.

DEFINITION 2.5 (MA-security with outsider KCIR). *Let $\mathcal{A}_{ma,out}$ be an outsider adversary against MA-security. Let pid_U^i be a set of identities of participant in the group with whom Π_U^i wishes to establish a session key and sid_U^i denotes a session id of an instance Π_U^i. $\mathcal{A}_{ma,out}$ is allowed to make queries to the Execute, Send, RevealKey, EphemeralKey Reveal, Corrupt oracles. $\mathcal{A}_{ma,out}$ breaks the MA-security with outsider KCIR notion if at some point there is an uncorrupted instance Π_U^i with the key sk_U^i and another party U' which is uncorrupted when Π_U^i accepts such that there are no other insiders in pid_U^i and the following conditions hold:*

- *there is no instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ or,*
- *there is an instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ which has accepted with $sk_{U'}^{i'} \neq sk_U^i$.*

DEFINITION 2.6 (MA-security with insider KCIR). *Let $\mathcal{A}_{ma,in}$ be an insider adversary against MA-security. It is allowed to query Execute, Send, RevealKey, EmphemeralKey Reveal and Corrupt oracles. It breaks the MA-security with insider KCIR if at some point there is an uncorrupted instance Π_U^i which has accepted with the secret key sk_U^i and another party U' which is uncorrupted when Π_U^i accepts and*

- *there is no instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ or,*
- *there is an instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ which has accepted with $sk_{U'}^{i'} \neq sk_U^i$.*

3. Review of Tan's Identity-based Authenticated Group Key Agreement Protocol. This section reviews the Tan's pairing-free ID-AGKA protocol for imbanced mobile networks. Tan's pairing-free ID-AGKA protocol consists of five phases namely, Setup phase, Key extraction phase, Key agreement phase, Remove phase and Join phase. The notions used throughout the paper are listed in Table.....

Setup: For a given security parameter k, PKG does the following:

- Choose a k-bit prime q and generate a group $\mathbb{G}$ over the elliptic curve, where P is the generator of the group of prime order q.
- Choose the master key $x \in \mathbb{Z}_q^*$ and compute the system public key $P_{pub} = xP$.
- Choose cryptographic hash functions as follows:
  - $H_0 : \{0,1\}^* \times \mathbb{G} \times \cdots \times \mathbb{G} \times Z_q \to \{0,1\}^k$
  - $H_1 : \{0,1\}^* \times \mathbb{G} \to Z_q$
  - $H_2 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \to Z_q$
  - $H_3 : \{0,1\}^* \to Z_q$
  - $H_4 : \{0,1\}^* \times \mathbb{G} \times \cdots \times \mathbb{G} \to Z_q$
- Publish the system parameters $\mathbb{G}, F_q, q, P, H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot), P_{pub}$.

Key Extraction: Public key generator extracts the secret key of U_i with identifier ID_i as follows:

- Choose a number $r_i \in \mathbb{Z}_q^*$ and compute $R_i = r_i P$.
- Compute a Schnorr signature [17] about the identity ID_i as $x_i = r_i + xH_1(ID_i, R_i)$.

U_i checks if $x_iP = R_i + H_1(ID_i, R_i)P_{pub}$. If the equality holds, U_i takes (x_i, R_i) as the private long-term key.

Key Agreement: Each node $U_i(1 \leq i \leq n-1)$ randomly selects two numbers $a_i, b_i \in \mathbb{Z}_q^*$ and computes $T_i = a_i x_i P, V_i = b_i P, s_i = b_i + x_i H_2(ID_i, T_i, V_i) \mod q$. Next U_i sends the message $(ID_i, T_i, V_i, R_i, s_i)$ to the powerful node U_n.

Upon receiving the message $(ID_i, T_i, V_i, R_i, s_i)$, U_n executes the following operations:

- Compute $P_i = R_i + H_1(ID_i, R_i)P_{pub}$ and check if $s_iP = V_i + H_2(ID_i, T_i, V_i)P_i$ holds. If it holds, U_n authenticates U_i.
- Choose two random numbers $a_n, b_n \in \mathbb{Z}_q^*$ and compute the following:

$$T_n = H_3(a_n||x_n)P, V_n = b_nP, Z_i = H_3(a_n||x_n)T_i (1 \leq i \leq n-1),$$
$$s_n = b_n + x_n H_4(ID_n||Z_1||Z_2||\cdots||Z_{n-1}||V_n||T_n) \mod q$$

- Broadcast the message $(ID_n, V_n, R_n, s_n, Z_1, Z_2, \cdots, Z_{n-1})$ to the group U.
- Compute the session key $SK = H_0(ID, Z, T_n, s_n)$, where $ID = ID_1||ID_2||\cdots||ID_{n-1}||ID_n$, $Z = Z_1||Z_2||\cdots ...|_{n-1}$.

Each U_i computes $P_n = R_n + H_1(ID_n, R_n)P_{pub}, T = (a_i x_i)^1 Z_i$, and checks if $s_nP = V_n + H_4(ID_n, Z, V_n, T)P_n \mod q$. If it is valid, U_i computes the group session key $SK = H_0(ID, Z, T, s_n)$.

3.1. Our Attack and Fix. Recall that, when the leakage of ephemeral secret is included in the security model, the leakage of these short term secrets should not allow the adversary to compute the session key. In Tan's protocol, the leakage of a_i and b_i will allow the adversary to find long term secret key from the signature. The adversary can compute x_i as $x_i = (H_2(ID_i, T_i, V_i))^{-1}(s_i - b_i) mod q$.

The adversary computes $T = (a_i x_i)^{-1} Z_i$ where Z_i can be easily eavesdropped from the transcript. The session key can be computed as $SK = H_0(ID, Z, T, s_n)$, where $ID = ID_1||ID_2||\cdots||ID_{n-1}||ID_n$ and $Z = Z_1||Z_2||\cdots||Z_{n-1}$. To fix the above attack, one solution is to use a signature such that the leakage of private key can be avoided on the leakage of ephemeral secrets while other solution suggests to mask the ephemeral secret. The masking can be done by a simple substitution $\tilde{b}_i = H_5(b_i, x_i)$, where $H_5 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \to \mathbb{Z}_q^*$. Now, the leakage of ephemeral secrets a_i and b_i will not allow the adversary to compute x_i from the signature.

4. Gupta and Biswas ECC-based AGKA Protocol. In this section, we first present the AGKA protocol by Gupta and Biswas and then, we discuss about the security vulnerabilities in their proposal. The protocol is insecure against insider colluding attack and ephemeral key leakage attack. The algorithm steps are as follows:

Setup(1^λ) : On input security parameter 1^λ, this phase outputs the system parameters *Params* in the following steps:

- Chooses an elliptic curve group $\mathbb{G}_1$ of prime order q . Let P be a generator of group $\mathbb{G}$. Let $\hat{e}$ be an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_2$ is a multiplicative group.
- Chooses cryptographic hash functions $H_0 : \{0,1\}^* \to \mathbb{G}_2$, $H_1 : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{Z}_q^*$.
- Finally publishes the system parameters $Params = \{\mathbb{G}_1, \mathbb{G}_2, q, P, H_0(\cdot), H_1(\cdot), \hat{e}\}$.

KeyGen$(params, ID_i)$: The phase performs the following for all the group members:

- Each party P_i publishes her public key $PU_i = s_iP$ and computes her private key as $PR_i = \frac{s_i}{s_i + QU_i}P$, where $QU_i = H_0(ID_i)$ and $s_i \in \mathbb{Z}_q^*$ is a randomly selected master key of each party P_i.

Key Agreement(x_i, pid) : In this phase, all the participating group members $P_1, P_2, \cdots, P_n$ have their indexes in cyclic form and also, all the members have already received their public/private key pairs. The steps for key agreement protocol are as follows:

Round 1:

- Each party $P_i(1 \leq i \leq n)$ chooses a random $r_i \in \mathbb{Z}_q^*$ and computes $\psi_i = r_iP$ and $h_i = H_1(\psi_i, r_i)$.
- Each Party computes the signature on h_i as $\sigma_i = h_i PR_i$ and broadcasts $\langle \psi_i, h_i^{-1}, \sigma_i \rangle$.
- On receiving these signatures $\langle \psi_j, h_j^{-1}, \sigma_j \rangle$, all participants verify as $\hat{e}(\sigma_j, h_j^{-1} Q_{ID\,j}) = \hat{e}(PU_j, P)$, where $Q_{ID\,j} = H_0(ID_j)P + PU_j$.

Round 2: On successful verification, each P_i computes $X_i = r_i(\psi_{i+1} - \psi_{i-1})$ and broadcast to all other participants.

Key Computation : Each party computes the shared group key as
$K_i = nr_i\psi_{i-1} + Y_i = (r_1r_2 + r_2r_3 + ... + r_nr_1)P$
where $Y_i = (n-1)X_i + (n-2)X_{i+1} + ... + X_{i-2}$.

4.1. Our Attacks. Here, we list some attacks on the above protocol and prove that the protocol is not secure. The attack points are as follows:

Key Generation Flaw: In this protocol, it is ambiguous to derive the public/private key pair by the participant itself. Usually, there are three cryptosystems in practice, Public Key Infrastructure (PKI), Identity based Cryptosystem (IBC) and Certificateless Cryptosystem (CL-PKC). In all the cryptosystems, the private key is partially or fully generated by the trusted third party. If the user can derive the public/private key pair by themselves, there will be no authentication because an adversary can do the same and hence, anyone becomes a valid member in any communication.

Insider Colluding Attack: The presented protocol is also vulnerable to insider colluding attack. Two insiders P_{i-1} and P_{i+1} can collude together to impersonate the participant P_i in any other group. The malicious participants eavesdrop the transcript $\langle \psi_i, h_i^{-1}, \sigma_i \rangle$ from the previous session and replay this in a new group. Further, note that the computation of X_i in **Round** 2 can be easily performed by P_{i+1} and P_{i-1}.

$$\begin{aligned} X_i &= r_i(\psi_{i+1} - \psi_{i-1}) \\ &= r_i\psi_{i+1} - r_i\psi_{i-1} \\ &= r_{i+1}\psi_i - r_{i-1}\psi_i \end{aligned}$$

The common group key can be computed as $K_i = nr_{i-1}\psi_i + Y_i$
where $Y_i = (n-1)X_i + (n-2)X_{i+1} + ... + X_{i-2}$.
Therefore, any two malicious insiders can impersonate a participant without his consensus and agree upon some session key. The adversary in our attack must be an active adversary which has the privilege to call *Send* oracle in standard security model.

Ephemeral Key Leakage Attack: Another drawback of the scheme is, the leakage of ephemeral key directly compromises the group session key. However, the authors claim in Theorem 7.9 [10], the session key resistance against the leakage of session specific temporary information but the given session key formula $K_i = nr_i\psi_{i-1} + Y_i$ is completely dependent on r_i.

5. Conclusion. In this paper, we analyze two AGKA protocols against the claimed security notions and we found them insecure. The Gupta and Biswas protocol is vulnerable to ESL attack as well as insider colluding attack while the Tan's AGKA protocol is ESL insecure. We also present a fix to the Tan's protocol.

REFERENCES

[1] R. Barua, R. Dutta, and P. Sarkar, *Extending joux's protocol to multi party key agreement*, in Indocrypt, vol. 2904, Springer, 2003, pp. 205–217.
[2] M. Bellare and P. Rogaway, *Entity authentication and key distribution.*, in Crypto, vol. 93, Springer, 1993, pp. 232–249.
[3] E. Bresson, O. Chevassut, and D. Pointcheval, *Provably authenticated group diffie-hellman key exchange-the dynamic case*, in Asiacrypt 2001, vol. 2248, Springer, 2001, pp. 290–309.
[4] ———, *Dynamic group diffie-hellman key exchange under standard assumptions*, in Advances in CryptologyEUROCRYPT 2002, Springer, 2002, pp. 321–336.
[5] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, *Provably authenticated group diffie-hellman key exchange*, in Proceedings of the 8th ACM conference on Computer and Communications Security, ACM, 2001, pp. 255–264.
[6] M. Burmester and Y. Desmedt, *A secure and efficient conference key distribution system*, in Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1994, pp. 275–286.

[7] K. Y. Choi, J. Y. Hwang, and D. H. Lee, *Efficient id-based group key agreement with bilinear maps*, in PKC 2004, Springer, 2004, pp. 130–144.
[8] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE transactions on Information Theory, 22 (1976), pp. 644–654.
[9] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, *Modeling key compromise impersonation attacks on group key exchange protocols*, in PKC 2009, Springer, 2009, pp. 105–123.
[10] D. S. Gupta and G. Biswas, *An ecc-based authenticated group key exchange protocol in ibe framework*, International Journal of Communication Systems.
[11] I. Ingemarsson, D. Tang, and C. Wong, *A conference key distribution system*, IEEE Transactions on Information theory, 28 (1982), pp. 714–720.
[12] S. H. Islam and G. Biswas, *A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks*, Annals of télécommunications-annales des telecommunications, 67 (2012), pp. 547–558.
[13] A. Joux, *A one round protocol for tripartite diffie–hellman*, in International algorithmic number theory symposium, Springer, 2000, pp. 385–393.
[14] J. Katz and J. S. Shin, *Modeling insider attacks on group key-exchange protocols*, in Proceedings of the 12th ACM conference on Computer and communications security, ACM, 2005, pp. 180–189.
[15] J. Katz and M. Yung, *Scalable protocols for authenticated group key exchange.*, in Crypto, vol. 3, Springer, 2003, pp. 110–125.
[16] S. Kyung-Ah, *Further analysis of id-based authenticated group key agreement protocol from bilinear maps*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 90 (2007), pp. 295–298.
[17] C.-P. Schnorr, *Efficient identification and signatures for smart cards*, in Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 239–252.
[18] G. Sharma, R. A. Sahu, V. Kuchta, O. Markowitch, and S. Bala, *Authenticated Group Key Agreement Protocol Without Pairing*, in International Conference on Information and Communications Security, Springer, 2017, pp. 606–618.
[19] K. Sharma and B. Gupta, *Attack in smartphone wi-fi access channel: State of the art, current issues, and challenges*, in Next-Generation Networks, Springer, 2018, pp. 555–561.
[20] ———, *Taxonomy of distributed denial of service (ddos) attacks and defense mechanisms in present era of smartphone devices*, International Journal of E-Services and Mobile Applications (IJESMA), 10 (2018), pp. 58–74.
[21] M. Steiner, G. Tsudik, and M. Waidner, *Key agreement in dynamic peer groups*, IEEE Transactions on Parallel and Distributed Systems, 11 (2000), pp. 769–780.
[22] Z. Tan, *An efficient pairing-free identity-based authenticated group key agreement protocol*, International Journal of Communication Systems, 28 (2015), pp. 534–545.
[23] Y.-M. Tseng, T.-T. Tsai, and S.-S. Huang, *Enhancement on strongly secure group key agreement*, Security and Communication Networks, 8 (2015), pp. 126–135.
[24] F. Wei, Y. Wei, and C. Ma, *Attack on an id-based authenticated group key exchange protocol with identifying malicious participants.*, IJ Network Security, 18 (2016), pp. 393–396.
[25] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, *A provably secure revocable id-based authenticated group key exchange protocol with identifying malicious participants*, The Scientific World Journal, 2014 (2014).
[26] T.-Y. Wu and Y.-M. Tseng, *Towards id-based authenticated group key exchange protocol with identifying malicious participants*, Informatica, 23 (2012), pp. 315–334.
[27] T.-Y. Wu, Y.-M. Tseng, and T.-T. Tsai, *A revocable id-based authenticated group key exchange protocol with resistant to malicious participants*, Computer Networks, 56 (2012), pp. 2994–3006.
[28] T.-Y. Wu, Y.-M. Tseng, and C.-W. Yu, *A secure id-based authenticated group key exchange protocol resistant to insider attacks.*, J. Inf. Sci. Eng., 27 (2011), pp. 915–932.
[29] F. Zhang and X. Chen, *Attack on an id-based authenticated group key agreement scheme from pkc 2004*, Information Processing Letters, 91 (2004), pp. 191–193.
[30] J. Zhao, D. Gu, and M. C. Gorantla, *Stronger security model of group key agreement*, in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ACM, 2011, pp. 435–440.

Edited by: Khaleel Ahmad
Received: Sep 13, 2018
Accepted: Feb 20, 2019

DOI 10.12694/scpe.v20i1.1460
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 101–112. http://www.scpe.org © 2019 SCPE

A PATTERN-BASED MULTI-FACTOR AUTHENTICATION SYSTEM

PANKHURI* AKASH SINHA† GULSHAN SHRIVASTAVA‡ AND PRABHAT KUMAR§

Abstract. User authentication is an indispensable part of a secure system. The traditional authentication methods have been proved to be vulnerable to different types of security attacks. Artificial intelligence is being applied to crack textual passwords and even CAPTCHAs are being dismantled within few attempts. The use of graphical password as an alternate to the textual passwords for user authentication can be an efficient strategy. However, they have been proved to be susceptible to shoulder surfing like attacks. Advanced authentication systems such as biometrics are secure but require additional infrastructure for efficient implementation. This paper proposes a novel pattern-based multi-factor authentication scheme that uses a combination of text and images resulting for identifying the legitimate users. The proposed system has been mathematically analyzed and has been found to provide much larger password space as compared to simple text based passwords. This renders the proposed system secure against brute force and other dictionary based attacks. Moreover, the use of text along with the images also mitigates the risk of shoulder surfing.

Key words: Security, Password, User authentication, Multi-factor, Pattern-based

AMS subject classifications. 68M12

1. Introduction.

Recent advances in the technology have resulted in the development of complex IT based systems for delivering value added services to the users. These systems may store users personal data with aim of providing personalized services to the users. The rapid growth in the demand for personalized services will eventual transform these systems into the storehouses of various types of personal information of the users. This urges for the requirement of having more robust and secure access mechanisms in order to mitigate the various security risks associated with the unauthorized access to these IT systems [1, 2, 3]. This makes the user authentication the most essential and indispensable component of such systems. There are three basic authentication methods which are based on token, biometric and knowledge [4]. Smart cards are token-based authentication system implementing knowledge-based techniques to enhance security as in case with ATM cards having a PIN number. Biometric based authentication techniques, such as fingerprints, iris scan or facial recognition provide highest level of security but are still expensive, slow, unreliable and hence, not yet widely adopted [5]. Knowledge based techniques are the most widely used authentication techniques which include both text and picture-based passwords [6].

The most basic mechanism for authenticating users is by the use of passwords [7]. The concept of using passwords is an efficient and cost effective solution for user authentication. The fundamental requirement for any password is that it should be easy to remember and must be secure enough. In other words, authentication process must be efficient and password must be tough to guess. Text based password continues to be the most widely used form of authentication methods owing to a number of factors such as easy to remember, tough to guess, and small time is required to finish the process. Studies have revealed that users often have a tendency of picking a short password so that it could be easily remembered but unfortunately, these passwords are easy to compromise. This can be attributed to the fact that textual passwords are just a series of characters (numeric, alphanumeric, and special characters) and are usually based on Latin or other well-known scripts supported by the input devices. This renders the textual passwords susceptible to various security attacks. It has been observed that 4.66% of accounts on rockyou.com have been compromised with the help of social engineering, dictionary and brute force attack [8]. As per Open Security Foundation, millions of credit card records has been compromised by the hackers from the big organizations like TRW, Sears Roebuck, Sony Corporation etc. [9]. According to a computer world news article, the security team at a large company ran a network password cracker and within 30 seconds and identified about 80% of the passwords [10].

In light of the above mentioned requirements, this paper proposes a novel pattern-based multi-factor recognition mechanism for user authentication. The proposed mechanism requires a user to enter textual key along

*Computer Science and Engineering Department, National Institute of Technology Patna, India (pankhuri.sai@gmail.com)
†Computer Science and Engineering Department, National Institute of Technology Patna, India (akash.cse15@nitp.ac.in)
‡Computer Science and Engineering Department, National Institute of Technology Patna, India (gulshanstv@gmail.com)
§Computer Science and Engineering Department, National Institute of Technology Patna, India (prabhat@nitp.ac.in)

with the clicks at specific areas on multiple graphical images. The combination of text and graphics increases the password space thereby making the authentication mechanism more robust and secure against various types of security threats. This paper further analyzes the storage requirements and the tolerance of the proposed password scheme with respect to different possible combinations of images and text in the proposed mechanism.

The rest of the paper is organized as follows: Section 2 provides the highlights of the existing literature related to the proposed work; Section 3 discusses the proposed system; Section 4 presents a mathematical analysis of the proposed system; and finally, Section 5 concludes the paper along with a brief discussion of the future works.

2. Literature Review. It is well established fact that humans can remember pictures better than text which makes graphical password schemes better alternative to text-based schemes [11]. Moreover, if the number of images is sufficiently large, the possible password space of a graphical password scheme exceeds than that of text-based schemes thereby, offering better resistance to dictionary attacks. In pattern-based recognition techniques, a user is being presented with a set of images and authenticated by recognizing the images he or she selected during the registration stage but in recall-based techniques, a user is asked to reproduce something that he or she created or selected earlier during the registration stage. Password retention was measured longitudinally three times: at the end of the first session (R1), one week later (R2) and four weeks later (R3), which revealed the following statistics presented in Table 2.1 [12].

TABLE 2.1
Response towards textual and graphical password scheme [12]

	Mode	**Mean R1**	**Mean R2**	**Mean R3**
No. of incorrect submission	Alphanumeric	1.61	2.82	1.43
	Graphical	0.28	2.44	1.20
Time of correct submission(sec)	Alphanumeric	9.01	22.53	20.76
	Graphical	5.28	9.87	8.99

Table 2.1 clearly depicts that graphical passwords are easy and efficient to implement, and therefore, the chance of incorrect submission and time taken for correct submission is always less than that of alphanumeric passwords. In contrast to graphical passwords, alphanumeric passwords require more effort to remember but their implementation for secure system is simpler. Table 2.2 and 2.3 lists few important statistics related to textual passwords.

TABLE 2.2
Most common textual passwords [13]

Top 10 Passwords	**Number of users**	**Percentage of use**
123456	1666	0.38
Password	780	0.18
Welcome	436	0.1
Ninja	333	0.08
abc123	250	0.06
123456789	222	0.05
12345678	208	0.05
sunshine	205	0.05
princes	202	0.05
Qwerty	172	0.04

From Table 2.2 and 2.3, it can be inferred that dictionary and brute force attack may decode alphanumeric passwords easily. Graphical password has been employed for implementing the users personal handheld device as

TABLE 2.3
Most popular structure of textual passwords [14]

Prevalent password	Number of users	Percentage of use
One to six characters	88164	19.91
One to eight character	272885	61.63
More than eight characters	169888	38.37
Only lowercase alpha	146486	38.37
Only uppercase alpha	1778	0.4
Only alpha	148264	33.49
Only numeric	26077	5.89
First capital last symbol	1259	0.28
First capital last number	17464	3.94

the password decoder and the user is being challenged with an image password with few hints [15][7]. Humans have exceptional ability to recognize image, therefore, PassFace Scheme has been implemented with cogno metric or search metric systems requiring user to remember a set of images both during password creation and authentication phase [16][17]. Figure 2.1 depicts the PassFaces grid scheme.

FIG. 2.1. *PassFaces grid [18]*

Hollingworth et al. [19] revealed that people may retain accurate, detailed, visual memories of objects which they attended previously. They suggested that user may remember specific parts of an image more accurately as their password if they are focused upon as shown in Figure 2.2.

In an ideal design, the cue is always helpful for legitimate user during authentication. Cued-recall system requires users to remember particular locations within image which is easier than that of pure recall. These systems may also be called as locimetric due to their reliance on identifying specific location. Sobrado et al.[20] developed a graphical password technique that deals with shoulder surfing problem in which user needs to identify his pre-selected pass objects among objects. User authenticates himself by clicking inside convex hull formed by these objects 2.3.

Sabzevar and Stavrou [15] proposed a methodology in which users need to move a frame along with objects until the pass object in this frame lines up with the other two pass objects. This process may be repeated to reduce the likelihood of getting authenticated randomly but this makes the entire process slow. Jansen et al. [6] proposed a graphical password mechanism for mobile devices in which user enrolls himself by selecting a theme consisting of thumbnail images and then registers a sequence of images as a password. During authentication, the user must enter the registered images in the correct sequence. The basic drawback of this technique is the number of thumbnail images is limited to 30 and hence, the password space is small.Figure 2.4 shows an instance of their second algorithm, where the user is required to move a frame until pass objects line up with

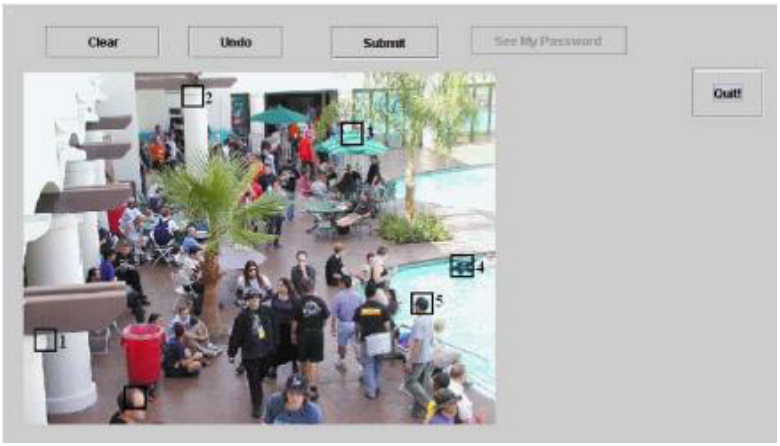


FIG. 2.2. *Cued Recall System [12]*

FIG. 2.3. *Convex hull formed with pre-selected objects [20]*

the other two pass objects. The main disadvantage of this is the slow authentication process.

Jansen et. al [6] proposed a graphical password mechanism for mobile devices. In enrollment stage, a user selects a theme which consists of thumbnail images and then registers a sequence of images as a password. During authentication, the user must enter the registered images in the correct sequence. One weak point with this technique is number of thumbnail images are limited to 30 as shown in Figure 2.5. This leads to a smaller password space.

3. Proposed Methodology. This paper implements a pattern-based multi-factor authentication scheme in which a number of images are displayed for a fixed time interval. The user has to click within a predefined area on a particular image for authentication. As the number of clicks increases, the security of the system also increases but at the same time it becomes more complex, therefore, the number of clicks required for authentication may vary as per the requirement of the system. The proposed system also requires a user to enter a key along with the click at specific areas on respective images in the slide show to enhance the level of security. The images considered in the proposed system are of 1360 x 660 pixels, each of which is displayed for an interval of 2.5 seconds during the slide show. Images selected for the authentication are preferably gray scale images to reduce the storage space and to make the process fast. During authentication, user has to input

FIG. 2.4. *Alignment of pass object across line [20]*

FIG. 2.5. *Grid of 30 thumbnail images [6]*

a key along with the click within a region of size 30 x 30 pixels around particular points in the corresponding images. If the region is too small, it will be more secure but at the same time, false rejection will be high for a genuine user. If this region is too large then it will be easier for the attackers to guess the clickable region. Graphical passwords are easy to remember, however, it is also prone to shoulder surfing, therefore, user has to input some keystroke along with the mouse click which would be difficult for the attacker to notice. The time duration for which each image is being displayed is another crucial factor. Longer the time interval of display, it will be easier for the user to authenticate. But, the authentication process will be much time-consuming and at the same time, attacker will get more time to guess the password. The basic working of the proposed system is as shown in Figure 3.1.

Studies on the Passface technique have shown that people often choose weak and predictable graphical passwords [21]. More research efforts are needed to understand the nature of graphical passwords created by real world users. Passface scheme has a shorter password-space than that of the system discussed in this paper. Davis et al. implemented Passface technique and discovered some obvious patterns among passwords [21]. For example, most users tend to choose faces of people from the same race which makes the Passface password predictable. This is not the case with this system, as different people select different click points in the same image containing many objects. Except for a few exceptions and mouse tracking spywares, it is very difficult for key logging or key listening spywares to crack graphical passwords. Mouse motion co-related with window position, size and timing has to be managed to compromise the graphical password system.

Like text based passwords, most of the graphical passwords including this system, are vulnerable to shoulder surfing [21]. However, adding the input keys in this system may help in preventing it. Recall-based password

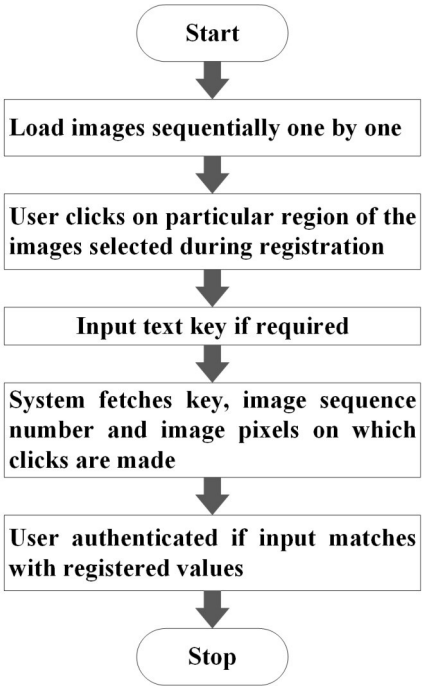


FIG. 3.1. *Flow Diagram for the Proposed system*

system is more prone to shoulder surfing than graphical password system. As drawing is being entered on the screen, an attacker needs to see the login process just once for getting the password and recall is not always a difficult task depending on memory prompts or cues. Passwords based on recognition-based techniques are remembered over a longer period of time. The system discussed in this paper provides more resistant to shoulder surfing and efficient than Jansen et. al algorithm [6] which is based on the correct sequence of clicks on the thumbnail images. The proposed system introduces a key, which would be difficult for an attacker to notice along with the correct click. The system discussed here is less confusing than the system used by Sobrado and Brdget for avoiding shoulder surfing as it contains thousands of pass-objects on the screen, out of which user had to select some objects which is being selected during the registration phase [6]. Therefore, introduction of key stroke along with click provide better protection against shoulder-surfing as compared with other algorithms. The formal specification regarding the working of the proposed system has been shown in Algorithm 1. The algorithm considers that the user has to click on *5* images (image1, image2, image3, image4 and image7) out of n images. Moreover, the user also enters an additional textual key along with the click on *image1*.

*The considered password string is : $\{a, 001, (615, 335), (555, 320), (1052, 335), (115, 160), (327, 695), 1111001\}$. Here a is the key during the first click and (615, 335),(553, 320),(1052, 335),(115, 160), (327, 695) are the co-ordinate positions of the clicks in the first, second, third, fourth and up to seventh respective images during the enrollment stage. The user does not have much difficulty in remembering the click points as these are well-defined locations inside an image displayed on the screen.

Algorithm 1: Authenticating a user with graphical passwords

Input:
1. mouseX=x-coordinate of mouse when it is being clicked
2. mouseY=y-coordinate of mouse when it is being clicked
3. ch=the key entered by the user
4. mouseClick=boolean variable which returns 1 if mouse is clicked, else 0

```
Create an array img of n images;
while all images not loaded do
    track status;
end
i ← 1
while i <= n do
    display img[i] for 2.5 sec;
    if i=1 and mouseClick=yes and mouseX ≥ (615-15) and mouseX≤(615+15) and mouseY≥(335-15)
    and mouseY≤(335+15) and ch=a then
        click1 ← 1
    end
    if i=2 and mouseClick=yes and click1=1 and mouseX≥(555-15) and mouseX≤(555+15) and
    mouseY≥(320-15) and mouseY≥(320+15) then
        click2 ← 1
    end
    if i=3 and mouseClick=yes and click2=1 and mouseX≥(1052-15) and mouseX≤(1052+15) and
    mouseY≥(335-15) and mouseY≤(335+15) then
        click3 ← 1
    end
    if i=4 and mouseClick=yes and click3=1 and mouseX≥(115-15) and mouseX≤(115+15) and
    mouseY≥(160-15) and mouseY≥(160+15) then
        click4 ← 1
    end
    if i=7 and mouseClick=yes and click4=1 and mouseX≥(327-15) and mouseX≤(327+15) and
    mouseY≥(695-15) and mouseY≤695+15 then
        print "USER AUTHENTICATED";
    end
    i ← i + 1
end
```

4. Discussion and Analysis. This section provides a mathematical analysis of the proposed system in terms of password space, storage requirement and tolerance level. The discussion is supported using a number of case studies in order to justify the efficiency of the proposed work.

Password Space

User needs to input an ASCII character as a key. Each key can take 2^7 different values.
Size of key = 1 Character
Password space for the keys= 2^7
Size of each image= 1360 x 660 pixels
Tolerance= 30 x 30 pixels
Hence, number of square grids for clicking = (1360 x 660) / (30 x 30) = 997

Consider the total number of images = 7
However, only 5 images have to be selected out of 7.

Number of possible ways in which this can be done is ${}^7C_5 = 21$
Total number of clicks required = 5
Password space for this = 21 x $(997)^5 = 2^{52.61}$

If key is input with the first click, password space of the system = (2^7) x $(2^{52.61})$ =$2^{59.61}$
However, the key can be input with any of the 5 clicks.
So, total Password space of the system = 5 x $(2^{59.61}) = 2^{61.93}$
This is greater than that of password space of an 8-character (ASCII) alphanumeric password, which is $(2^7)^8 = 2^{56}$.

A super computer may test 100 million passwords every second. Therefore, time required to test 10^7 passwords = 1s
Time required to test $2^{23.25}$ passwords = 1s
Time required to test $2^{61.93}$ passwords = $(2^{61.93})$ / $(2^{23.25}) = 2^{38.68}$s = 13964.76 years = 14 thousand years (approx.)
Time required to test 2^{56} passwords = (2^{56}) / $(2^{23.25}) = 2^{32.75}$ seconds = 229 years
Hence, it can be clearly deduced that it would be infeasible to use even a supercomputer for a brute force attack on the proposed system. Table 4.1 shows that there is no much difference in password space, however, the time required to brute force graphical password is 60 times greater than that of textual password system. Hence, graphical password with 7 images and key of length 1 byte is 60 times more secure than textual password.

TABLE 4.1
Comparison of textual and graphical password system

Password system	**Password space**	**Time required to brute force**
Textual password of length 8 characters	2^{56}	229 years
Graphical password with 7 images in slide show and key of length 1 byte	$2^{61.93}$	13964 years (almost 14 thousand years)

4.1. Case Study. The following sub-section presents different case studies with respect to the size of input key, number of images and other crucial parameters of the proposed system.

Case 1.
Size of input key = 4 characters
Number of images = 5
Number of input keys required = 5
Password space for keys = $(2^7)^4 = 2^{28}$
Password space for the clicks = ${}^7C_5$ x $(997)^5 = 2^{52.61}$
Total Password space of the system = 2^{28} x $2^{52.61} = 2^{80.61}$
Time required to test $2^{85.78}$ passwords = $(2^{80.61})/(2^{23.25}) = 2^{57.36}$ s = $5.86 * 10^9$ years = $5.86 * 10^3$ million years (approx.)

Case 2. Number of images in slide show = 10
Size of input key = 4 Characters
Number of input keys required = 5
Password space for clicks = $({}^{10}C_5)$ x $(997)^5 = 2^{57.78}$
Total password space of the system = 2^{28} x $2^{57.78} = 2^{85.78}$
Time required to test $2^{85.78}$ passwords = $(2^{85.78})/(2^{23.25}) = 2^{62.53}$ s = $2.11 * 10^{11}$ years = $2.11 * 10^5$ million years (approx.)

Case 3.
Size of input key = 1 Character

Number of images = 6
Total number of clicks required = 5
Password space for the clicks = ${}^6C_5$ x $(997)^5 = 2^{52.39}$
The key can be with any of the 5 clicks.
Total password space of the system = $5x(2^7)$ x $(2^{52.39}) = 2^{61.71}$
Time required to test $2^{61.71}$ passwords = $(2^{61.71})/(2^{23.25})$ s = $2^{38.46}$ s = 11989 years

Memory space is still greater than that of an 8-character long textual ASCII password but it may be reduced if 8-bit gray scale image is being displayed. Hence, total space required to store the slideshow = 6 x 1360 x 660 x 8 bits = 5.14 MB

Case 4.
Let the number of images = 5 Total number of clicks required = 5 Therefore, password space for the clicks = $(997)^5 = 2^{49.8}$
The key can be input with any of the 5 clicks. Therefore, total password space of the system = 5 x (2^7) x $(2^{49.8}) = 2^{59.12}$
Total space required to store the slide show with gray scale image = 5 x 1360 x 660 x 8 bits = 4.28 MB
Time required to test $2^{59.12}$ passwords = $(2^{59.12})$ / $(2^{23.25})$s = $2^{35.87}$s = 1991 years
Table 4.2 depicts that even if memory storage required for graphical password is much higher than that of textual password but still graphical password is more secure than textual password.

Table 4.2
Comparison of textual and graphical password system with respect of storage memory and time required to brute force

Password system	**Storage memory required**	**Time required to brute force**
Textual password of length 8 characters	56 bits	229 years
Graphical password with 8-bit 6 gray scale images	5.14 MB	12 thousand years
Graphical password with 8-bit 5 gray scale images	4.28 MB	2 thousand years

Case 5.
Let the number of images = 4
Number of clicks required = 4
Password space for clicks = $(997)^4$
Total password space of the system = 4 x 2^7 x $997^4 = 2^{48}$
Time required to test $2^{29.4}$ passwords = $(2^{48})/(2^{23.25})$s = $2^{24.75}$s = 326 days

Table 4.3
Impact of images and size of key on password space and time required to brute force

Number of images in slide show	**Size of keys in characters**	**Image selected**	**Password space**	**Time required to brute force**
7	1	5	$2^{61.93}$	14 thousand years
5	4	5	$2^{80.61}$	5.86 x 10^3 million years
10	4	5	$2^{85.78}$	2.11 x 10^5 million years
6	1	5	$2^{61.71}$	11989 years
5	1	5	$2^{59.21}$	1991 years
4	1	4	2^{48}	326 days

From Table 4.3, it may be concluded that even if the size of textual key is kept same but the number of

images during slide show are being doubled then time required to brute force also gets doubled. Increase in size of keys is not as influential as increase in number of images during slide show. If we reduce the number of images to 4, the password space will be less than that of an 8-bit ASCII password (textual). Hence, this case is the optimal case with minimum password space.

Probability of Identifying the correct password by an attacker
Probability of identifying the correct key = $1/(2^7)$
Probability of identifying the correct click with which the key has to be input = 1/5
Probability of identifying the correct region of first click in an image = 1 / ((1360 x 660)px / (30 x 30)px) = 3/2992
Probability of choosing the correct 5 images on which to click out of the 7 images = $1/ {}^7C_5 = 1/21$
Probability of identifying the correct password of the user = (1/5) x (1/128) x (1/21) x $(3/2992)^5$ = 7.54 x $(10)^{-20}$
This is too small. Hence, our graphical authentication system is very less vulnerable to password guessing.
It is clear from Table 4.4 that chances of guessing graphical password will always be less than that of alphanumeric password.

TABLE 4.4
Identification probability of different password systems

Password System	Identification probability
Alphanumeric password	1.3 x 10^{-17}
Graphical password	7.54 x 10^{-20}

Storage memory space
Graphical passwords require more storage space as compared with textual password but in this era of technological advancement, the storage space for enhancing the security may not be an issue.
Size of each image = 1360 x 660 px
Space taken to store 1 px = 32-bit in a 32-bit display
Total no. of images in our system = 7
Therefore, total space required = 7 x 1360 x 660 x 32 bits = 23.96 MB which is very large.

Case 1: Binary image
Space required to store each pixel = 1 bit
Total space required to store the slideshow = 7 x 1360 x 660 x 1 bits = 767 KB
This is much less than space required in a 32-bit display. Although the number of colors which can be displayed in the image does not effect the proposed system, however, some users might not prefer using binary images as these can cause some inconvenience to them.

Case 2: 8-bit gray scale images
Total space required to store the slideshow = 7 x 1360 x 660 x 8 bits = 6 MB

Table 4.5 depicts the storage requirements for different password systems.

Storage Space for the Proposed System
The password string consists of one key and coordinates (x,y) of each of the 5 clicks.
Space required to store 1 key character = 1 byte
Space required to store 5(click points) x 2(coordinates of each click), i.e. 10 integers = 10 x 2 bytes = 20 bytes
Number of bits required to store input key associated with one of the 7 image = 3 bits
(010 indicate the key has to be entered with 2^{nd} image.) Number of bits required to represent the selected image = 7 bits
(1111001 indicates clicks in 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 7^{th} images are required respectively)

TABLE 4.5
Total memory space of different password system

Password system	Storage space required
Textual password length = 8 characters	56 bits
Graphical password Binary image	767 KB
Graphical password 8-Bit gray scale image	6 MB
Graphical password Colored image	23.96 MB

Therefore, total memory space required to store the password string of each user = 3 bits + 1 byte + 20 bytes + 7 bits = 178 bits
The password string will be of the form: key, click, (X1, Y1), (X2, Y2), (X3, Y3), (X4, Y4), (X5, Y5), select
Here, click refers to the sequence number of the click with which key is being associated and select refers to the sequence number of images(in 7 bits) which have been selected by the user. Table 4.6 shows that single password string of length 8 character requires 56 bits of memory while proposed scheme for password requires 178 bits of memory.

TABLE 4.6
Comparison of memory space requirements

Password system	Space required to store password string
Textual password	56 bits
Proposed scheme of password	178 bits

5. Conclusions and Future Works. User authentication is one of the most important component of a secure system. Even after the development of advanced authentication mechanisms such as biometrics, the traditional concept of passwords still continues to be the most widely adopted means for user authentication. Owing to the limitations and weaknesses of text-based passwords such as smaller password space, susceptibility to brute force and shoulder surfing attacks, etc., this paper proposes a novel pattern-based multi-factor authentication scheme that involves the use of a combination of textual and graphical passwords. The proposed system has a larger password space and is secure against dictionary attacks since it involves additional mouse input along with keyboard input. Moreover, a brute force attack would require automatic generation of all possible mouse-click and text combination in order to crack the actual password. This renders the bruce force attack infeasible for the proposed system.

Traversing through multiple graphical images during the login process can be tedious and time taking process. This also requires maintaining tens of thousands of pictures in a centralized database and as such optimal storage space is also a matter of concern. Future research can be made regarding the storage of the images in an optimal manner in conjunction with minimization of network latency. Further, one of the major design challenge for the proposed system is regarding the accuracy and reliability of the user inputs. A high error tolerance may lead to many false positives while low tolerances may lead to many false negatives. This requires an optimal error tolerance strategy in order to enhance the accuracy of the system.

REFERENCES

[1] HUNT, H. C., & SHEA, A. (2018). Enhanced user authentication. U.S. Patent Application No. 10/078,783.
[2] KHARI, M., SHRIVASTAVA, G., GUPTA, S., AND GUPTA, R. (2017). Role of Cyber Security in Today's Scenario. In R. Kumar, P. Pattnaik, and P. Pandey (Eds.), Detecting and Mitigating Robotic Cyber Security Risks (pp. 177-191). Hershey, PA: IGI Global.
[3] SAXENA, A., SHRIVASTAVA, G., & SHARMA, K. (2012). Forensic investigation in cloud computing environment. The International Journal of forensic computer science, 2, 64-74.
[4] VELSQUEZ, I., CARO, A., & RODRGUEZ, A. (2018). Authentication schemes and methods. Information and Software Technology, 94(C), 30-37.
[5] AWAD, A., & LIU, Y. (2019). Cognitive Biometrics for User Authentication. In Biometric-Based Physical and Cybersecurity Systems (pp. 387-399). Springer, Cham.

[6] Jansen, W., Gavrila, S. I., Korolev, V., Ayers, R. P., & Swanstrom, R. (2003). Picture password: a visual login technique for mobile devices. UMBC Student Collection.
[7] Abhishek, K., Roshan, S., Kumar, P., & Ranjan, R. (2013). A comprehensive study on multifactor authentication schemes. In Advances in Computing and Information Technology (pp. 561-568). Springer, Berlin, Heidelberg.
[8] Franchi, E., Poggi, A., & Tomaiuolo, M. (2015). Information and Password Attacks on Social Networks: An Argument for Cryptography. Journal of Information Technology Research (JITR), 8(1), 25-42.
[9] CNN Business (2013). 5 of the biggest-ever credit card hacks. https://money.cnn.com/gallery/technology/security/2013/12/19/biggest-credit-card-hacks [Accessed on Jan. 31, 2019]
[10] Gilhooly, K. (2005). Biometrics: Getting back to business. Computerworld, May, 9, 2005
[11] Amit, E., Rim, S., Halbeisen, G., Priva, U. C., Stephan, E., & Trope, Y. (2019). Distance-dependent memory for pictures and words. Journal of Memory and Language, 105, 119-130.
[12] Agarwal, G., Singh, S., & Shukla, R. S. (2010). Security analysis of graphical passwords over the alphanumeric passwords. International Journal of Pure and Applied Sciences and Technology, 1(2), 60-66.
[13] Emil Protalinski (2012). The top 10 passwords from the Yahoo hack: Is yours one of them?. https://www.zdnet.com/article/the-top-10-passwords-from-the-yahoo-hack-is-yours-one-of-them [Accessed on Jan. 31, 2019]
[14] Ahitagni (2012). 453,000 Yahoo voice, username and password leaked. http://www.ahitagni.com/?p=422 [Accessed on Jan. 31, 2019]
[15] Sabzevar, A. P., & Stavrou, A. (2008). Universal multi-factor authentication using graphical passwords. In Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference on (pp. 625-632). IEEE.
[16] De Angeli, A., Coventry, L., Johnson, G., & Renaud, K. (2005). Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. International Journal of Human-Computer Studies, 63(1-2), 128-152.
[17] Renaud, K. V. (2009). Guidelines for designing graphical authentication mechanism interfaces. International Journal of Information and Computer Security, 3(1), 60-85.
[18] Passfaces: Two Factor Authentication for the Enterprise. http://www.passfaces.com/ [Accessed on Jan. 31, 2019]
[19] Hollingworth, A., & Henderson, J. M. (2002). Accurate visual memory for previously attended objects in natural scenes. Journal of Experimental Psychology: Human Perception and Performance, 28(1), 113-136.
[20] Sobrado, L., & Birget, J. (2002). Graphical passwords, The Rutgers Scholar, An electronic bulletin of undergraduate research. Rutgers University, Camden New Jersey, 4,12-18.
[21] Zheng, Z., Liu, X., Yin, L., & Liu, Z. (2010). A Hybrid Password Authentication Scheme Based on Shape and Text. Journal of Computers, 5(5), 765-772.

Edited by: Khaleel Ahmad
Received: Nov 15, 2018
Accepted: Feb 17, 2019

DOI 10.12694/scpe.v20i1.1465
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 113–160. http://www.scpe.org © 2019 SCPE

A DETAILED DESCRIPTION ON UNSUPERVISED HETEROGENEOUS ANOMALY BASED INTRUSION DETECTION FRAMEWORK

ASIF IQBAL HAJAMYDEEN* AND NUR IZURA UDZIR†

Abstract. Observing network traffic flow for anomalies is a common method in Intrusion Detection. More effort has been taken in utilizing the data mining and machine learning algorithms to construct anomaly based intrusion detection systems, but the dependency on the learned models that were built based on earlier network behaviour still exists, which restricts those methods in detecting new or unknown intrusions. Consequently, this investigation proposes a structure to identify an extensive variety of abnormalities by analysing heterogeneous logs, without utilizing either a prepared model of system transactions or the attributes of anomalies. To accomplish this, a current segment (clustering) has been used and a few new parts (filtering, aggregating and feature analysis) have been presented. Several logs from multiple sources are used as input and this data are processed by all the modules of the framework. As each segment is instrumented for a particular undertaking towards a definitive objective, the commitment of each segment towards abnormality recognition is estimated with various execution measurements. Ultimately, the framework is able to detect a broad range of intrusions exist in the logs without using either the attack knowledge or the traffic behavioural models. The result achieved shows the direction or pathway to design anomaly detectors that can utilize raw traffic logs collected from heterogeneous sources on the network monitored and correlate the events across the logs to detect intrusions.

Key words: Anomaly detection, Clustering, Heterogeneous logs, Filtering, Feature analysis

AMS subject classifications. 68T10, 68T05

1. Introduction. The contemporary IDS are incompetent in taking advantage on the benefit of heterogeneous data sources for investigation to identify intrusions [1]. The significance of using several logs for intrusion detection was presented and emphasized by Abad et al. [2], and the outcome of attacks on various logs was demonstrated. As a proof of concept, the existence of an intrusion was determined by correlating the system calls with network logs and the experiments were conducted to detect a particular anomaly. Artificial data were used to train and test the model projected. Detection accuracy improved with log correlation, but then predict the next system call method underperformed for this problem. Apart from that, the description of attack traces and the way such information were extracted from various log sources used in the study were not described.

UCLog, a unified logging architecture [3] correlates events from various logs for intrusion detection. The correlations between the activities were utilized to achieve better accuracy and also trace the origin of intrusion exempting the administrator to examine such information. This was further extended as UCLog+ [4] to parse and store alerts and incident records. Cross-Layer Based Intrusion Detection and Prevention [5] detects intrusions by manipulating the data obtainable through multiple layers of the protocol stack. Most importantly, Denning (1987) [6] suggested to construct a framework for a general-purpose intrusion-detection expert system (IDES) which is independent of any system, application environment, system vulnerability, or type of intrusion needs to be extensively explored and unsupervised heterogeneous anomaly detection framework (UHAD) [7] is a step towards this directive.

2. Dataset Description. The sensitivity of the data urges to remain proprietary affecting the availability of datasets regularly [8]. This creates a bigger challenge in gathering the appropriate data for the experiments. The most accepted log source utilized for intrusion detection nowadays is network traffic and the widespread use of network traffic was due to its availability and standardization [9]. Selecting useful data is a significant task in the pre-processing stage [10] and the most essential concern is the form of data sources to consider [4] as the selection of right data sources helps to identify various kinds of intrusions or attacks. To assess the capability of the framework proposed in detecting anomalies, the logs captured by heterogeneous sources are needed. One of the challenges of the Honeynet project i.e., Scan of the Month #34 (SOTM#34) [11], was used as input data that contains both intrusive and non-intrusive events.

*Faculty of Information & Sciences Engineering, Management & Science University, 40100 Shah Alam, Selangor, Malaysia (asif@msu.edu.my).

†Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia .

The dataset used for the experiments described in this paper has been chosen for the following reasons: appropriateness of the data and real traffic data.

Appropriateness of the data. Testing the proposed framework requires logs collected directly by heterogeneous sources, i.e., operating systems, applications and network devices. SOTM#34 dataset consists of logs collected from heterogeneous sources and hence appropriate. The KDDCup1999 and DARPA IDS evaluation datasets were commonly used for testing intrusion detectors consist of data sniffed from a particular location on the network (Homogeneous data) and not collected directly from the heterogeneous sources where the actions are destined to and therefore the usage of this data to test the framework is not applicable.

Real traffic data. The framework intends to detect a wide variety of attacks in the absence of a learned traffic model and therefore requires raw data (naturally unlabelled) recorded directly by multiple sources. While both KDDCup1999 and DARPA IDS evaluation dataset are synthetic, SOTM#34 contains real log data from heterogeneous sources and hence suitable. This dataset has been examined by several participants [12, 13, 14, 15] of the challenge, and also scrutinised by Panichprecha [16] and Herreras and Gomez [17], hence providing better confidence on the analysis results to assess the capability of the proposed framework. The dataset comprises of several logs i.e., Apache Server, Linux syslogs, Snort NIDS, and IPTables firewall, which were captured by dissimilar sources in a Honeynet system.

The Honeynet constitutes three key systems:

- Bridge, a multi-homed host operating an unknown distribution of Unix/Linux and performs routing/filtering using Netfilters IPTables kernel module.
- Bastion is a Network Intrusion Detection System running Snort with the Bleeding Snort rule sets in conjunction with those provided with the basic Snort package.
- Combo is the victim system running Red Hat Linux with a 2.4.20-8 kernel, an Intel Pentium III 740 MHz processor, and 128 MB RAM. It runs multiple virtual IP addresses on the 11.11.79.0/24 network.

The framework makes two assumptions about the data used:

Assumption 1: The data from the logging sources has logged every action and is free from unauthorised alteration(s) or fabrication(s).

Assumption 2: The majority of the logged events are usual traffic with a minority percentage of malicious traffic.

2.1. Data Pre-processing. Logs were written in a proprietary fashion, since there were no establishments to standardize log formats [18]. Therefore, the relevant features in a log needed for processing has to be extracted beforehand. Parsing is the process of extracting meaningful data [19], and translating the data into a format to ease and enhance further processing. Existing log parsing tools introduce new features derived from existing features in the resulting parsed log; and this log have to be parsed again to remove the additional features in order to retain the integrity of the logs. To avoid the overhead of double parsing and also to maintain logs integrity, we wrote our own parsers rather than using existing tools.

The significance of the custom written parser is as follows:

1. Able to extract all the features as exactly recorded in the log, irrespective of the difference in the delimiters.
2. Able to extract features precisely without losing any part of the value for the feature.
3. Does not use any kind of log format specifications to extract the features of a specific log.
4. Does not introduces new features derived from the existing features into the parsed log.
5. Able to adjust the timestamp of the events in a log for the time difference between the logs to support correlation, if the time difference is provided.
6. Able to remove unwanted feature names stored together with the values and also removes unnecessary punctuations, i.e., white space, comma, equal (=), appearing in between feature values. This is to avoid misinterpretations of treating a single feature as multiple features by the subsequent process.

However, due to difficulties in precisely extracting the features from Linux syslogs, Sawmill [20] was used for initial parsing. The parsed log contained '(empty)', symbolizing the absence of value for a particular feature; and this was parsed again to replace '(empty)' with '-' to maintain uniformity across the logs. The timestamp in all the logs was separated as date and time while parsing to ease synchronization. The parser is a part of our

extractor which is integrated with supplementary functions, i.e., Isolator and Timestamp Synchronizer. The feature selection and clustering methods necessitate the number of features in an event should be the same throughout the log in order to be processed. Therefore, the isolator separates the events of IPTables firewall log in separate files, as there were differences in the number of features for different connections (TCP, UDP, ICMP) having 24, 22 and 21 features, respectively. Time is one of the important factors to correlate events between logs, hence the synchronizer adjusts the timestamp for the time difference between the logs.

Feature selection was used in intrusion detection for recognizing and eliminating insignificant features [21], and was applied on logs to assist the progressing process to enhance the accuracy in predicting and categorising abnormal events. Knowing the predictive ability of every feature in relation to every other feature in a log will assist in precisely identifying the important features. Therefore, feature selection was accomplished by setting all feature in the log as class attributes to recognise the connection of the set feature with other features. Since each subset contains events of different durations, feature selection was applied on all datasets to verify, whether the features selected for a particular log was uniform for all subsets. The selected features of all class attribute settings were summarized to find the unselected features, and every such unselected feature was removed from the respective parsed log before further processing. The same strategy was followed for various logs used for the experiments.

Keeping the evidence in a single common setup [18] by transporting all events from various logs together facilitates the analysis process in identifying most of the anomalies. The ability to distinguish normal behaviour from an attack can be better accomplished by analysing more features, but not every feature is relevant to the detection task [22]. Including all features in the logs will make the schema size bigger leading to difficulty in managing. Every log records features that they deem as important. Due to this nature, there were variations in the number of features recorded by a particular log and the values contained in these features. Especially with the logs from heterogeneous sources, the variations between the features were more diverse. Therefore, the process of framing a generic schema for a given set of logs becomes critical. To analyse the events from various logs together, a generic format (GF) was outlined, with common and significant features existing in the considered logs.

This features, i.e., timestamp, source IP, destination IP, source port, destination port, protocol, were also used by other researchers [23, 24, 25, 26] for intrusion detection. Moreover, Message was also incorporated in GF as it states the action performed by the event and their patterns were fully dissimilar for benign and malicious events. No additional features were introduced in GF to represent the log source an event belongs to, as it may mislead the clustering method. The framed GF not only states the features for GFL, but also the relevant features from various logs, that can fit in.

2.2. Data Treatment. This section describes the details on the treatment of data to make it viable for processing by the framework components. SOTM#34 log data was subjected to two processes namely feature extraction and feature selection.

2.2.1. Feature Extraction. Five operations were performed on the logs in this phase namely parsing, synchronizing, relocating, isolating and dividing and are defined as follows:

1. Parse: Extracting the relevant feature(s) from the log events and storing the extracted features using a common delimiter.
2. Synchronize: Adjusting the time stamp of a particular log to match the time difference with other logs, enabling the discovery of relationship between events across the logs.
3. Relocating: Moving the events from a subset to another after synchronizing to follow the duration covered by subsets.
4. Isolate: Separating the events in IPTables firewall log that differs in the number of features it contains.
5. Divide: Separating the IPTables firewall and Snort IDS events in several files to match the time duration covered by each subset.

Sample events from various logs are provided in Figure 2.1. The followings are the difficulties faced, while parsing the logs to extract the features precisely:

1. White space or blank space separates the features in an event by all the logs, while it also appears inside a feature value, e.g., Apache SSL-Error Log.

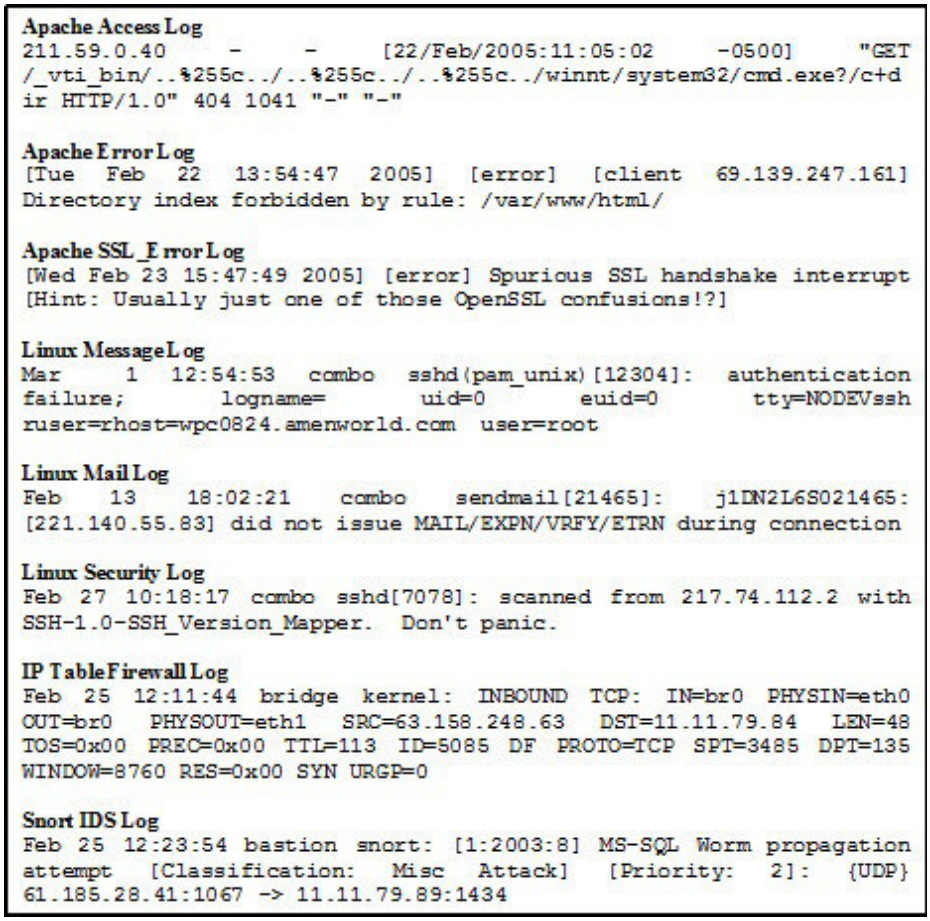


FIG. 2.1. *Samples of Raw Logs*

2. Some of the features have delimiters to mark the start and end of a value, while others do not have such delimiters, e.g., Timestamp and Status Code in the Apache Access Log.
3. Delimiters are dissimilar with different features in the same log, e.g., [ ] for Timestamp and for ClientRequestLine in Apache Access log, [ ] for Classification and for Protocol in Snort IDS Log.
4. Delimiter used for a particular feature in the logs from a particular source was also not even, e.g., ClientIP in Apache access log is delimited by white spaces whereas Apache error log is delimited by [ ].
5. The number of features in an event for a particular log is not similar for all the events in that log, e.g., TCP, UDP and ICMP connections in IPTables firewall Log.
6. The name of the feature precedes most of the values for all the events in the log, e.g., IPTables firewall and Linux Message Log.
7. Time stamp format across the logs are not even, e.g., Apache logs includes year in time stamp while others do not, and the time stamp format of Apache access log differs from the Apache error and Apache SSL-error log.

There are no establishment to standardize log formats [18], and therefore every logging source writes log in its own format with the features based on what they consider as important. So, every log was parsed separately to extract the relevant features. Even though there are five operations in this phase, not all the logs are subject to all these operations. Variations in the package of data and the lack of time synchronization between the log events recorded by multiple sources which is very common in production environments necessitates separate treatment. Hence, this module provides three different apparatus to handle the logs as illustrated in Figures 2.2,

2.3 and 2.4. The common operation available in all the setup is parsing and the application of synchronization, relocation, isolation and division may vary with logs depending on the package and time difference of the collected data.

The first step in feature extraction is to identify the boundary of a feature and consequently extracting the feature values for every log event. To rectify the differences in the delimiters used with various features in the raw logs, the features in an event has to be given a common delimiter which in our case is a comma (,). Therefore, all the logs in text files has been parsed to transform the events with a common punctuation and was recorded in a Comma Separated Value (CSV) file that will ease and enhance further processing. Existing log parsing tools introduce new features derived from existing features in the resulting parsed log, and this log has to be parsed again to remove the additional features in order to retain the integrity of the logs. Moreover these tools do not offer facilities to synchronize the time difference across logs and to segregate events within a log, if there is a difference in the number of features recorded by the events. Therefore, custom written Perl scripts for different logs were used to implement this step. But, due to difficulties in parsing Linux Syslog, Sawmill [20] was used to parse this log.

Apache Server Logs and Linux Syslog. Apache server provides three logs which are access, error and SSL-error whereas Linux provides three logs that are message, mail and security. All these logs undergo three operations that are parsing, synchronizing and relocating (Figure 2.2). The number of features and the content

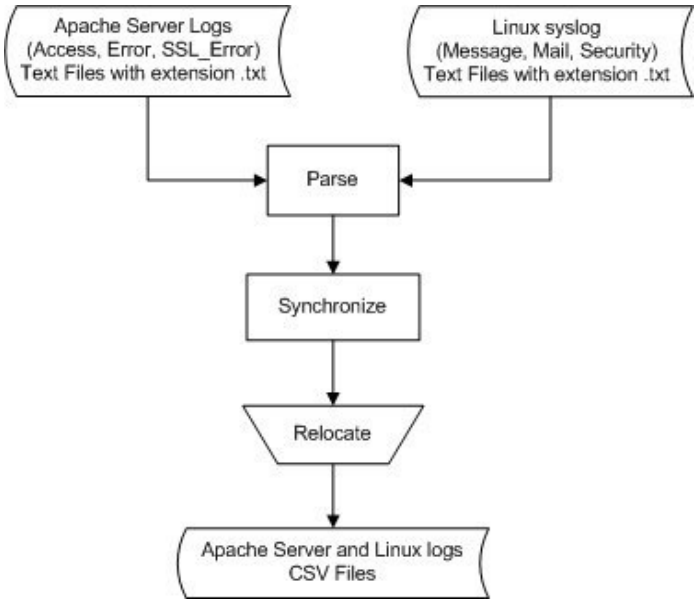


FIG. 2.2. *Apache and Linux Logs Parsing Apparatus*

of features varies with logs, therefore separate Perl scripts were written to parse each log. The logs provided in text file were parsed to extract the features and stored in a CSV file. During parsing, the timestamp was separated as date and time to facilitate synchronization. The year in which the events are recorded was not available in Linux Syslog and therefore introduced during parsing. The time between the logging sources i.e., bridge and bastion, was synchronized and thus IPTables firewall and Snort IDS events were in sync. But then, the time between the combo (victim machine) and the bridge was not synchronized.

Subsequently the Apache server and Linux syslog events contrasted by 4 hours 47 minutes against IPTables firewall log [15, 14, 13]. This was taken as the standard to synchronize and subsequently, the synchronizer balanced Apache server logs and Linux syslogs for the distinction towards the IPTables firewall log. Because of the adjustment in date and time after synchronization, a portion of the log events have been shifted to the particular subsets to maintain the time duration covered for every subset.

IPTables Firewall Log. The events of IPTables firewall log contains the name of the feature for most of the features in all the events recorded. Only the value for every feature in an event is needed for processing and on the other hand, keeping the feature name for every event increases the size of this log. Moreover, it causes additional overhead while processing events in the subsequent phases. Hence, the feature name preceding the value and the = that separates the feature name and feature value were removed during parsing. Moreover the timestamp was separated as date and time to match the format in other logs. After parsing, the events were isolated according to connections, i.e., TCP, UDP and ICMP, having 24, 22 and 21 features, respectively. The isolated events were maintained in three separate files and every such file is subdivided into four separate log files matching the duration of the other log subsets. This will allow the following process to handle different connections separately with subsets of moderate size. Since the Apache server log and Linux Syslog were synchronized towards IPTables firewall log, the timestamp of IPTables firewall log does not need synchronization. The pre-processing flow of IPTables firewall log is illustrated in Figure 2.3.

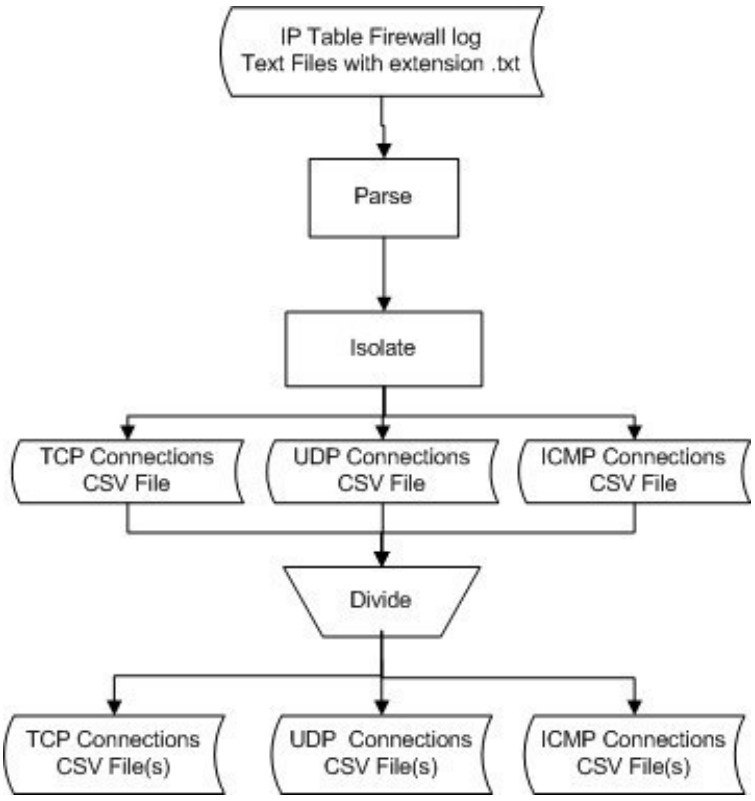


FIG. 2.3. *IPTables Firewall Log Parsing Apparatus*

Snort IDS Log. Several Snort IDS log events did not have a value for the features classification and priority and those events that has the value was preceded by the feature name. Hence the name of these features in these log events were removed during parsing and a hyphen (-) was introduced for those events which do not have a value for this feature. Moreover, the timestamp was separated as Date and Time to match the format followed by other logs. The parsed log events were divided into four separate files to match the duration of the respective subsets tested. Synchronization and isolation were not needed for this log, as there was no time difference with IPTables firewall log and no difference in the number of features in between the events. The pre-processed logs were stored in CSV files, and the flow of process is illustrated in Figure 2.4.

Outcomes of Feature Extraction. Irrespective of the variations in the setup to handle different logs, the output is provided in CSV files that are ready to be processed by the following modules. The extracted features

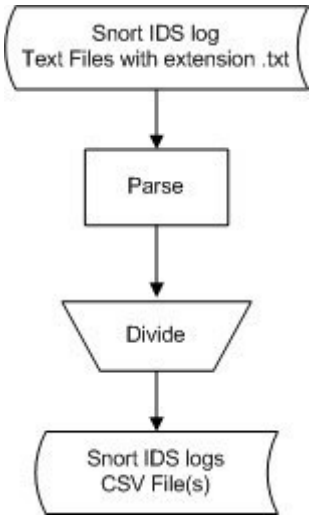


FIG. 2.4. *Snort IDS Log Parsing Apparatus*

of the logs belonging to particular subset is maintained separately. The time taken to parse and synchronize was less than a second for smaller sized logs, i.e., less than 1000 events, and a maximum of 27 seconds to 50 parse and isolate 179752 events in the IPTables firewall log. The pre-processed logs were maintained in CSV files and the volume of events in every log by subset is provided in Table 2.1.

TABLE 2.1
Details of the Log Subsets

Log Type	Subset-1	Subset-2	Subset-3	Subset-4
Access	256	539	1280	464
Error	433	604	1270	484
SSL Error	74	64	29	48
TCP	9632	32646	34748	13210
UDP	613	1996	2387	1546
ICMP	171	789	1303	474
Message	112	149	208	103
Mail	42	62	54	31
Security	138	263	232	122
Snort	4423	17049	5013	10293
Total Events	15894	54161	46524	26775

2.2.2. Feature Selection and Retention. Selection and retention of features for every log is accomplished in two steps as illustrated in Figure 2.5 which includes:

1. Selecting features and summarizing the selected features identified for a particular log.
2. Removing unselected features from the respective log.

Weka's CfsSubsetEval estimates features by considering the specific predictive capacity of each feature and BestFirst scans for intricate dealings between features [27]. Since the target was to increase the precision by using only the features assisting in predicting the anomalous events, Weka's [28] CfsSubsetEval and BestFirst methods were used to implement this step. The log for which the features need to be selected is loaded into Weka Explorer. The attribute evaluator is set to CfsSubsetEval and the search method is set to Best-First. To perform the selection process, a feature in the log is set as class attribute to recognise the association of the fixed feature with further features and the features selected are noted. Knowing the predictive ability of every

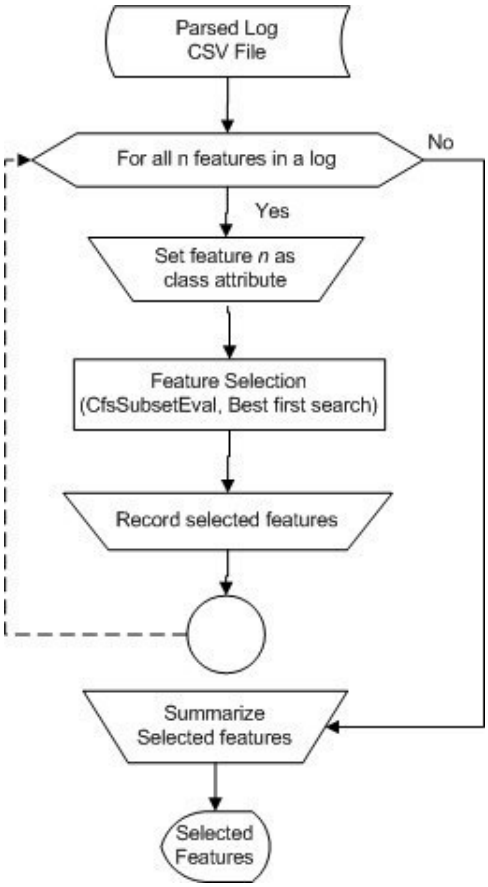


FIG. 2.5. *Feature Selection Strategy*

feature in relation to every other features in a log will assist in precisely identifying the important features and therefore every feature in the log is set as class attribute and the selected features were recorded. The number of times the features are selected for a log depends on the number of features in the log. The selected features of every class attribute setting were summarized to find the unselected features of a particular log. The same apparatus is used for different logs and every log used for the experiment is subject to this process to find the optimal set of features for a particular log. Subsequently, the features selected for a particular log across different subsets were summarized to find the unselected features. Every such unselected feature was removed from the respective parsed log before further processing. The selected features used for further processing is provided in Table 2.2.

Most importantly, this final set of features selected for every log consists of all significant features planned to be extracted for GFL except IPTables firewall log. The feature protocol in the IPTables firewall log was not selected for any class attribute setting. This is because the events of this log were previously isolated according to connections (TCP, UDP, ICMP), due to the difference in the number of features recorded for each of these connections. Therefore, the isolated events recorded in separate logs contained the same value for the feature protocol. Moreover the feature selection algorithm also requires the number of features in all the events to be the same. Eventually, applying feature selection on this isolated log events have not selected this feature for any class attribute setting, due to the similarity of the values for this feature, i.e. protocol.

2.2.3. Framing Generic Format. The features for GFL were identified by examining the features available in the logs. The only feature available in all the logs was the Timestamp (Date f_d, Time f_t). Features like Source IP Address (f_{sip}) existed in all logs except Apache SSLError and Linux mail log, whereas Destination

TABLE 2.2
Features Selected from Parsed Logs

Log		Features
Apache Server	Access	Date, Time, IPclient, Clid, Userid, clientRequestLine, Status Code, Object-Size, Referrer, Agent
	Error	Date, Time, clientIP, Msg
	SSL-Error	Date, Time, Severity, Msg
IPTables firewall	TCP	Date, Time, Bridge, Direction, Protocol, IN, PHYSIN, OUT, PHYSOUT, LEN, TOS, PREC, TTL, ID,DF, SPT, DPT, WINDOW, RES, STATUS, URGP, SRC, DST
	UDP	Date, Time, Bridge, Direction, Protocol, IN, PHYSIN, OUT, PHYSOUT, LEN, TOS, PREC, TTL, ID,DF, SPT, DPT, LEN, SRC, DST
	ICMP	Date, Time, Bridge, Direction, Protocol, IN, PHYSIN, OUT, PHYSOUT, LEN, TOS, PREC, TTL, ID,DF, TYPE, CODE, id, seq, SRC, DST
Linux Syslog	Message	Date, time, Logging device, Daemon, PID, Operation, User, Tty, UID, EUID, Remote host, System message
	Mail	Date, time, Logging device, From, To, Daemon, Mailer, Stat, Class, Priority, Protocol, Message ID, Relay, Control address, DSN, Queue ID, Messages queued, Messages delivered, Bytes queued, Bytes delivered, Delay, Xdelay
	Security	Date, time, Logging device, Daemon, PID, Operation, User, Source, System message, Messages
Snort IDS		Date, time, Logging device, Destination IP, Source port, Destination port, Classification, Snort priority

IP Address (f_{dip}), Source port (f_{sp}), Destination port (f_{dp}) and Protocol (f_{pr}) were available only in IPTables firewall log and Snort IDS logs. Initially GFL (Table 2.3) was constructed with these seven features from various

TABLE 2.3
Generic Format Specification - Version 1

Generic Format	Date	Time	Source IP	Destination IP	Source Port	Destination Port	Protocol	Log Source
Access	Date	Time	IPclient					Log name
Error	Date	Time	ClientIP					Log name
SSL-Error	Date	Time						Log name
TCP	Date	Time	SRC	DST	SPT	DPT	PROTO	Log name
UDP	Date	Time	SRC	DST	SPT	DPT	Protocol	Log name
ICMP	Date	Time	SRC	DST				Log name
Mail	Date	time						Log name
Message	Date	time	Remote host					Log name
Security	Date	time						Log name
Snort IDS	Date	time	SourceIP	Destination IP	Source port	Destination port	Protocol	Log name

logs. Additionally a feature by name log-source was introduced in GFL to specify the name of the log an event belongs to, since it consists of events from various logs. The aforesaid features of the filtered events from all the logs (whichever GFL features available in the respective log) were brought forward to GFL. Clustering this GFL resulted in higher false negatives affecting the accuracy of clusters, which eventually resulted in detecting only 12% of the anomalous events in the log. Message (f_{me}) is a feature available in most logs except IPTables firewall log, but it has different names in various logs. Therefore, a common name (Message) was suggested, and the respective features from various logs were extracted accordingly.

The new version of GFL (Table 2.4) includes all the features from the previous version together with the Message. Clustering the GFL with these nine features has improved the accuracy (nearly 80% in average) than

using the previous version of GFL. But, the subsequent detection of anomalies remains approximately the same as the previous. The improvement in the accuracy was due to the introduction of Message, but still the usage of log-source as a feature of GFL which does not exist as a feature in the actual logs affected the classification of events. In order to avoid the effect of log-source in classifying the events, log-source was removed from the GFL. The next version of GFL constitutes only eight features excluding log-source and is presented in Table 2.5. The features were extracted accordingly from the respective logs and recorded in GFL. Clustering this GFL improved the accuracy to nearly 90% and a substantial increase in detected anomalous events. IPTables firewall log merely watches and records the traffic through the network and it can be taken as an additional option to improve detection. Therefore, it was excluded from the GFL used for clustering and the events were maintained separately to be used for correlation during analysis. Many of the features stated in GFL, especially IP address

TABLE 2.4
Generic Format Specification - Version 2

Generic Format	**Date**	**Time**	**Source IP**	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	**Message**	**Log Source**
Access	Date	Time	IPclient					Client Request Line	Log name
Error	Date	Time	ClientIP					Msg	Log name
SSL-Error	Date	Time						Msg	Log name
TCP	Date	Time	SRC	DST	SPT	DPT	PROTO		Log name
UDP	Date	Time	SRC	DST	SPT	DPT	Protocol		Log name
ICMP	Date	Time	SRC	DST					Log name
Mail	Date	time							Log name
Message	Date	time	Remote host					System Message	Log name
Security	Date	time						System Message	Log name
Snort IDS	Date	time	SourceIP	Destination IP	Source port	Destination port	Protocol	Rule	Log name

TABLE 2.5
Generic Format Specification - Version 3

Generic Format	**Date**	**Time**	**Source IP**	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	**Message**
Access	Date	Time	IPclient					Client Request Line
Error	Date	Time	ClientIP					Msg
SSL-Error	Date	Time						Msg
TCP	Date	Time	SRC	DST	SPT	DPT	PROTO	
UDP	Date	Time	SRC	DST	SPT	DPT	Protocol	
ICMP	Date	Time	SRC	DST				
Mail	Date	time						
Message	Date	time	Remote host					System Message
Security	Date	time						System Message
Snort IDS	Date	time	SourceIP	Destination IP	Source port	Destination port	Protocol	Rule

and port numbers, were not available in Apache SSL-Error and Linux mail log. Therefore, the events belonging to these logs were also excluded from the GFL that was used for the following process, i.e., clustering, and were maintained separately as per the features stated in GFL (Table 2.6) to be used at some stage during analysis. Out of the eight features used to construct GFL, f_{me} has gained critical importance because it exhibits more about the action performed on the system than the other features. The Message (f_{me}) feature was available in all the logs except IPTables firewall log and therefore the STATUS feature available in TCP connections of this log was used as Message. The Source IP (f_{sip}) and Source port (f_{sp}) appears as a part of the SystemMessage in Linux security log. In order to facilitate the clustering and further analysis, both of these features has been extracted and replicated as a separate feature in this log for all the events where Source IP (f_{sip}) and Source

TABLE 2.6
Generic Format Specification - Proposed

Generic Format	**Date**	**Time**	**Source IP**	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	**Message**
Access	Date	Time	IPclient					client RequestLine
Error	Date	Time	ClientIP					Msg
Message	Date	time	Remote host					System message
Security	Date	time	Source		Port			System message
Snort IDS	Date	time	SourceIP	Destination IP	Source port	Destination port	Protocol	Rule
SSL-Error	Date	Time						Msg
TCP	Date	Time	SRC	DST	SPT	DPT	PROTO	STATUS
UDP	Date	Time	SRC	DST	SPT	DPT	Protocol	
ICMP	Date	Time	SRC	DST				
Mail	Date	time						

port (f_{sp}) is available in SystemMessage. The final version of GFL used for analysis was constructed with eight features as mentioned in Table 2.6. The GFL features and the corresponding features chosen from various logs that fit in GFL were tabulated (Table 2.6). The usage of GFL with these features (as stated in Table 2.6) increased the clustering accuracy to an average of 95% with various subsets which naturally improved the volume of anomalies detected.

Adding features to GFL, which is available in any one of the logs considered, decreased the clustering accuracy and also the subsequent detection of anomalies. Therefore, the proposed GFL was considered appropriate for the set of logs considered for detection. Moreover, the set of features chosen for GFL not only applies for the logs considered, but also to a wide variety of logs of similar nature. However, the reader should note that the selection of features for GFL and the appropriate features selected from different logs was based on the logs considered and it may differ with different logs.

3. UHAD Components. The main objective of the framework is to detect anomalous events by correlating the information available in various logs and the process involved to achieve that is summarised in the overall algorithm. Apart from this, every single process stated in the algorithm has been designed to achieve a specific goal that contributes to the ultimate objective. Therefore, this section describes various components of the framework used in the process of anomaly detection and also specifies the reason behind choosing these components. It also states the metrics used to evaluate the performance of each component.

3.1. Clustering Events. The objective of this step is to separate abnormal events from the normal events for various logs. As there were many patterns in the log for both normal and abnormal, the number of distinctive patterns needs to be identified. Manual analysis of log events to identify unique patterns is almost impractical as it takes longer time and prone to errors. Therefore, the usual events are to be recognized and reduced before analysis to decrease the processing overhead.

Clustering is an unsupervised learning scheme for grouping similar or identical events together. It also has the ability to decide on the existence of intrusive events in the raw logs [29] by grouping these events in separate clusters. Therefore, we make use of the clustering algorithms to group the log events according to the patterns. The traditional steps like training and testing employed in machine learning based intrusion detectors [31] and data mining based intrusion detectors [30, 32, 33, 34] by splitting or rearranging the test data to differentiate with training data to suit the requirements of a particular study was not used. Even though this method often results in better precision and accuracy, it deteriorates performance, if the tested data contain events that were not learned by the training model, making it unreliable for real time detection. To overcome these limitations of model based approaches, only the clustered events of a particular log were used for further analysis and not the cluster model which was usually used to detect anomalies in upcoming log events. Therefore, the proposed strategy makes use of the existing clustering algorithms to group the log events and the parameters required by these algorithms were manipulated based on the log examined. The clustering methods used necessitate the amount of cluster (K) to assemble the events. As several logs were used, clustering each log with different

clusters (K) to select the best cluster is time consuming. In order to isolate dissimilar event patterns in detached clusters, K was manipulated based on the patterns in the log. A procedure comprising two stages was used for isolating log events, which are:

Step 1: Predicting the best number of clusters (K_{ij}) for a given E_{ij} , and

Step 2: Clustering E_{ij} using the predicted K_{ij} .

Applying the clustering method distinctly for dissimilar services improves the detection quality [35]. Every log is an outcome of a service and therefore every single log from a source was treated separately for this step. IPTables firewall log is an additional option to improve the ways of detection as it just watches and records the traffic through it. Therefore, it was excluded for clustering and will be used for correlation during analysis to support the anomalies identified.

In step 1, the ideal number of clusters (K_{ij}), i.e., the number of clusters that is appropriate for a particular log to be grouped, was manipulated according to the experimental setting stated in Experiment 1. The manipulated K_{ij} serves as the output for step 1 and accordingly K_{ij} was used to cluster the respective logs in step 2.

There are numerous clustering methods and every method generates clusters differently for the same dataset. This poses the difficulty in deciding the appropriate algorithm for a particular context. The usage of existing clustering algorithms, especially K-Means and EM clustering for intrusion detection is very common and was used in many previous works [30, 32, 35, 36, 37, 38, 39, 40, 41, 42]. Siriporn and Benjawan [43] used Farthest First (FF) and K-Means clustering to detect intrusions [44] and used FF especially to detect rare attacks. This exposes the application of these algorithms for intrusion detection and the results achieved suggest the use of these algorithms for this purpose.

K-Means [45] treats all features equally [42] and is computationally faster with the ability to handle larger datasets. Moreover, it is order independent, i.e., it generates the same clusters of data irrespective of the sequence of the data presented. Since voluminous data received from the logging sources have to be grouped with minimum time consumption to facilitate anomaly detection, the usage of K-Means is applicable and appropriate. Farthest First (FF) [46] is a fast, simple, approximate, [28] hierarchical and distance based clustering using a distance measurement analogous to K-Means [47]. The calculation of cluster centroids in consecutive iterations by FF is contrary to that of K-Means, i.e., places each cluster centroid in turn at a point farthest from the current cluster centroid. This speeds up the cluster process due to less reassignments and adjustments with most datasets. This nature of FF helps to produce accurate clusters provided the events were qualitatively different from the other with minimum time consumption. EM [48] is an iterative clustering algorithm that groups the data in a way that is different from K-Means. It has the ability to optimize large number of features and also finds good estimates of the missing values in the features of a dataset. The log events examined contains many features and the values for some of the features were not available in many of the events in the log. In this case EM provides better judgement on the missing values in the logs thereby producing better clusters. Although, K-Means and FF are faster in clustering compared to EM, all the three algorithms were tested to know its ability in accurately grouping different patterns and volume of events.

In step 2, the events from various logs were clustered using the respective K_{ij} manipulated from step 1. Three clustering algorithms namely K-Means, EM and FF were used for clustering and its accuracy in clustering events of various logs were compared. The chosen clustering methods uses K in generating clusters and seed value to resolve on initial cluster centres. Even the same algorithm produces dissimilar clusters with different parameter initializations, led to the difficulty in selecting the suitable parameters. Because of this reason, the K value needed for clustering was predicted using EM in the previous step, i.e., step 1. No methods were available to manipulate the appropriate seed value for a given set of events. Choosing random seed values and clustering several times will increase the processing time. Therefore, the default seed value or the seed value equivalents to the number of events being clustered were used with different settings. The experiment for step 2 was conducted according to Experiment 2. Apart from that, in order to analyse the performance of the algorithm in separating normal and abnormal events in various logs, the clusters generated by various algorithms and its settings with different logs were examined.

3.2. Filtering Clustered Events. The objective of this phase is to eliminate the usual events (noise) whilst holding the anomalous events for subsequent processing. Filtering is a process of reducing events for

further analysis that are unlikely to hold information of importance [19]. Hence the clustered events were filtered to remove the unneeded events. Since anomalous events were less in number compared to benign events generated by normal usage, it falls in smaller clusters. Therefore, the smaller clusters needs to be identified based on the volume of events it contains. Since, multiple logs with different event volumes were used for detection, deciding a common cluster size to identify smaller clusters is not reasonable. Hence, a threshold (E_t) for identifying sparse clusters was calculated based on the volume of events and clusters for a particular log and the threshold is defined as:

$$E_t = \{E_{ij}/k_{ij}\} \tag{3.1}$$

Threshold may vary with logs due to the difference in the volume of events and the number of clusters generated for each log. Filtering events using this calculated threshold was to remove normal events (referred as filtered-out events) and to retain abnormal events (referred as filtered-in events) for further scrutiny. Therefore, the characteristics of anomalies were not required for filtering events. The clustered logs were filtered and evaluated according to the specification stated in Experiment 3.

3.3. Aggregating Filtered-In Events. The aim of this step is to combine the redundant events thereby reducing the events in the filtered log. Though, the filtered-in events were basically abnormal, using only the unique events will reduce processing overhead and increase the accuracy in formation of clusters. Even though much work use sampling methods for data reduction, it has been noted by Tavallaee et al. [49], that sampling methods in anomaly detection introduces a significant bias that degrades the performance. Hence, we chose aggregating instead, as both were basically data reduction techniques. Aggregation merges redundant records into a single record [19]. A group of two or more events were united, if all the features in the events were accurately analogous to the directly succeeding event(s), and the accumulated event serves as the representative. No features were introduced in the aggregated log to symbolize those aggregated events. The volume of events that gets reduced due to aggregation is based on the feature values that an event contains and therefore there are chances of zero reduction, if all the events in the filtered log were unique. Therefore the percentage of events reduced was not evaluated.

3.4. Transferring Events. The objective of this step is to extract the selected features of all the events from various aggregated logs as stated in GF and appropriately placing the respective features in GFL. This is to enable the analysis process to examine the events from various logs in a single structured format. Since all the GFL features were not existing in all the logs, absence of a specific feature in a log was substituted with a hyphen (-) during transfer. The performance of this step was verified to make sure whether the stated features in GF for every single event in the aggregated logs (E^{**}_{ij}) were completely transferred to GFL.

3.5. Clustering GFE. Despite the fact that the logs were clustered and filtered earlier, there are probabilities of having a less number of usual or irrelevant events; due to the imprecision in clustering and/or the succeeding withholding by the filtering threshold. To determine such events and also to discover the association between the events in GFL which comprises events from several logs, GFE was re-clustered. The GFE was clustered using the same two-step strategy, algorithms and parameter settings used previously for the individual logs. GFL was maintained separately according to the algorithms and settings used in the previous phases. This is to enable separate treatment of GFL accordingly in this phase and also the subsequent phases with respect to the algorithms and settings. The step 1 of the clustering strategy was carried out as per the setup provided in Experiment 4. Experiments for step 2 was conducted according to the algorithms and settings provided in Experiment 5. The clustered events (GFE) were stored in ARFF format and maintained separately with respect to algorithm and the respective settings.

3.6. Detecting Anomalous Events by Analysing Features. The aim of this step is to analyse clustered events to identify the relation between them with respect to the features it contains and thereby detecting the anomalous events from various logs. A mixed approach that uses multiple features of anomalies might be an eligible solution for different circumstances [50]. Therefore, the analysis process concentrates on features namely IP address (f_{sip}, f_{dip}) and port numbers (f_{sp}, f_{dp}). Additionally cluster number (f_k) was also used to get a clear picture of the intrusions.

3.6.1. IP Address Analysis. Normally an intrusion leaves multiple signs of its presence in various logs [2, 4]. As such, the events related to an intrusion may have been captured by various logs. To discover the presence of such intrusive events in multiple logs, this step identifies the relation between the IP address, i.e., source IP address (f_{sip}) and destination IP address (f_{dip}), of the events from various logs and the clustered GFL events (GFE) were used for analysis. Every abnormal activity that was captured by Apache server logs and Linux syslog must have raised an alert in Snort IDS log, as IDS has this capacity by design. So, the IP address that exists in Linux and Apache events that also exist in Snort IDS events were identified primarily. Not all IP address in Apache and Linux log has an matching with Snort IDS log, since some of the anomalous activities may have been missed by Snort IDS; but then, the anomalous patterns must have joined together in the same cluster during clustering. Therefore, to detect also those abnormal activities which were missed by Snort IDS log, all the events in those clusters to which the identified IP address belongs to were extracted and considered as anomalous. The experiments for this step were conducted and the detected anomalous events IPAE were evaluated as per Experiment 6. Attacks are frequently launched from an IP address or from an IP subnet [37] and therefore capturing every abnormal action originating from various IP address becomes significant. This is the reason behind evaluating the coverage of IP address in the detected anomalous events apart from the anomalous events itself.

3.6.2. Port Number Analysis. The objective of this step is to identify anomalous events based on port numbers. This is not a substitute to IP Address analysis, but to recognise and retain those anomalous events which were available in those clusters that were not identified by IP analysis. The findings of Kim et al. [51] also reveals that the usage of port numbers for classifying network traffic is still applicable and also suggest to use port based analysis methods. Most of the anomalous activities were launched from a host by exploiting the unassigned and dynamic ports, since no fixed service was running on these ports. Therefore, the source port number (f_{sp}) and the destination port number (f_{dp}) of the events were checked against the listing of the dynamic and unassigned port numbers as per Internet Assigned Numbers Authority (IANA). As most of the normal events have been filtered out in the previous phases, those events in (GFE) having port numbers matching with the IANA listing of dynamic and unassigned port numbers were detected and considered as anomalous. The experiments for this step were conducted and the detected anomalous events (PAE) were evaluated as per Experiment 7.

3.7. Consolidating Anomalous Events. The objective of this step is to consolidate the anomalous events discovered by both analysis methods to serve as the output of the framework. This was achieved by performing three operations namely combining, correlating and concentrating the events. Some of the anomalous events which were recognized during IP analysis may also have been recognized during port analysis. Therefore, these anomalous events, i.e., IPAE and PAE were compared to identify the distinct events and every event was brought together for correlation. To recognize the relation between the events of Linux Syslog, Apache logs and Snort IDS log with IPTables firewall log, the IP address (f_{sip} and f_{dip}), excluding internal IP address) of these events were compared with IPTables firewall log; and the matching events from IPTables firewall log were extracted and appended with the previously combined events. To concentrate on the most critical anomalous events, less significant events were reduced using a threshold (A_t) on the occurrence of events pertaining to an IP address. This was accomplished by identifying IP addresses of events which satisfies a specific threshold (A_t), and consequently the events pertaining to these IP address were extracted. These extracted events are deemed as anomalous and serves as the result of the framework. The experiments for this step were conducted and the consolidated anomalous events (AE) were evaluated as per Experiment 8.

4. Experimental Design for UHAD. This section illustrates the Unsupervised Heterogeneous Anomaly Detection Framework (UHAD) and the data pre-processing steps were discussed in detail followed by the abstract description of the framework. The inner details of the framework components were described together with the algorithm or strategy used to implement the components. Additionally, the flow of data, the input received and the output generated by each of the components were also presented. Finally, the contribution of the framework components towards anomaly detection was also discussed. The details of the experiments done at each step, i.e., clustering, filtering and analysis, of the framework are described in this section. This includes the respective algorithms used, together with the parameter settings or the needed parameters of the algorithm

and the parameters evaluated.

4.1. Experiment 1: Manipulating Ideal Clusters for Individual Logs. Prediction of the best number of clusters by clustering the logs using EM clustering with the default parameter values, i.e., K = -1 and seed = 100 (K is used by the clustering algorithm to classify the given set of events into K clusters and seed is used to identify the initial cluster centre for every cluster). Apache server log constitutes three logs namely access, error and SSL-error log whereas Linux Syslog constitutes three logs namely message, mail and security log. All the events recorded by Snort IDS were provided in a single log. Every log was treated separately by EM clustering to manipulate the ideal number of clusters that a log can be clustered.

4.2. Experiment 2: Clustering Individual Logs. To capture the impact of K and seed in clustering, the algorithms were tested with four different parameter settings. The default seed value of K-Means, EM and FF are 10, 100 and 1, respectively. The settings used are as follows: Setting-1 (S1): Ideal clusters (K_{ij}) manipulated in step 1.1 of the overall algorithm with the default seed of the respective clustering algorithm. Setting-2 (S2): Ideal clusters (K_{ij}) manipulated in step 1.1 of the overall algorithm with the seed value set to the total number of events in a particular log (L_{ij}). Setting-3 (S3): Doubling up ideal clusters (K_{ij}) manipulated in step 1.1 of the overall algorithm with the default seed of the respective clustering algorithm. Setting-4 (S4): Doubling up ideal clusters (K_{ij}) manipulated in step 1.1 of the overall algorithm with the seed value set to the total number of events in a particular log (L_{ij}). The experimental setup for the step 2 of the clustering strategy is provided in Table 4.1. Every log was clustered with all the three algorithms and four settings and therefore the experiments were conducted 12 times on every log. Subsequently the clustering accuracy achieved with different algorithms and settings were calculated separately. The ability of the clustering algorithm in identifying and

TABLE 4.1
Experiment 2 - Clustering Individual Logs

Algorithms	Settings	Evaluated Parameter
K-Means, EM, FF	1, 2, 3, 4	Accuracy

placing the events in the respective cluster gains significance, as it helps the following component to identify and remove those clusters which are insignificant. Hence, the quality of clusters produced by these algorithms was calculated using Weka Experimenter with 10 fold cross validation and 10 iterations to allow every part of the log to be tested. A true positive (TP) decision assigns two similar events in the same cluster whereas a true negative (TN) decision assigns two dissimilar events to different clusters. Failure to assign the events in the appropriate cluster is measured using false positive (FP) and false negatives (FN). FP decision assigns two dissimilar events to the same cluster whereas FN decision assigns two similar events to different clusters. All these four measurements decides the cluster goodness and therefore the accuracy of clustering is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN}$$

The accuracy achieved by various algorithms and settings with various logs were evaluated.

4.3. Experiment 3: Filter-in Events of Sparse Clusters. Every clustered log with the respective algorithms and settings were filtered separately according to the threshold calculated for the respective log. During the calculation of the threshold, if the calculated threshold turns to be a decimal number, e.g., 20.3, it was rounded to the next ascending integer, e.g., 21. The number of events in the cluster, i.e., cluster size, is a whole number and the cluster size must be less than the threshold to be filtered-in, the threshold was rounded to the next integer. The percentage of events reduced with different logs with the algorithms and settings due to the application of the threshold is evaluated and the volume of abnormal events retained for further examination were analysed. This is to ensure that, the application of filtering threshold is able to reduce the log events and at the same time, retains the abnormal events.

4.4. Experiment 4: Manipulating Ideal Clusters for Individual Logs. Prediction of the best number of clusters for GFL using EM clustering with the default parameter values, i.e., K = -1 and seed = 100. As GFL was maintained separately for each algorithm and setting, the clusters appropriate for that GFL were manipulated separately.

4.5. Experiment 5: Clustering GFL. Using the K manipulated in the first step (Experiment 4), GFL was clustered with the configuration provided in Table 4.2. TP, TN, FP and FP were measured to calculate the accuracy of clusters formed using the formula stated in Experiment 2.

TABLE 4.2
Experiment 5 - Clustering GFL

Algorithms	Settings	Evaluated Parameter
K-Means, EM, FF	1, 2, 3, 4	Accuracy

4.6. Experiment 6: Anomaly Detection using IP Address. Every clustered GFL, i.e., GFE, with the respective algorithms and settings were analysed separately based on the relationship between the events with respect to IP Address and Cluster Number. The volume of anomalous events (IPAE) detected and the volume of IP addresses covered by these anomalous events was evaluated. Additionally, the results of IP analysis was compared with SOTM/#34 analysis results which serves as the ground truth for the anomalies in this dataset. This is to ascertain the ability of the framework in detecting anomalous events without using a knowledge-base or traffic models.

4.7. Experiment 7: Anomaly Detection using Port Number. Every clustered GFL with the respective algorithms and settings were analysed separately based on port numbers. The volume of anomalous events detected and the volume of IP addresses covered by these anomalous events were evaluated.

4.8. Experiment 8: Consolidating Anomalous Events. The anomalous events detected by IPA and PA with the respective algorithms and settings were consolidated separately. The threshold was used to focus on the most significant anomalies that needs to be immediately addressed and the impact of the variable threshold (A_t), i.e., 3, 6, 9, in retaining the anomalous events from various IP addresses were evaluated. A three-way handshake (also called as three message handshake and/or SYN-SYN-ACK) is a method used to set up a connection over an Internet Protocol based network. In other words, there must be at least three events related to an IP address in a traffic log to signify the establishment of a connection between two machines. Hence the threshold was chosen as three, which requires at least three events from an IP address to exist in the log to be selected as consolidated. This will also show whether the victim machine has responded to the request of the attacker. The threshold is increased by six and nine to focus especially on those events which may provide enough evidence on the anomalous actions between attack source and victim.

To overcome the limitation of unsupervised anomaly detection approaches, we propose UHAD (Unsupervised Heterogeneous log-based Anomaly Detection), a knowledge independent framework that uses unsupervised clustering algorithms to detect anomalous events from heterogeneous logs. The grouped log events are further examined by several knowledge independent functions to detect anomalies. Every component receives the processed log events from the preceding component, and manipulates the needed parameters for the process based on the log events received.

The components of the framework are implemented using the following applications:

- Weka (Waikato Environment for Knowledge Analysis) is a popular collection of machine learning algorithms written in Java which can be applied directly to a dataset or called from your own Java code and is well-suited for developing novel machine learning schemes. It supports several data mining tasks and particularly data pre-processing, classification, regression, clustering, feature selection, and visualization.
- Perl is a high-level, interpreted, dynamic programming language offering dominant text processing services with no limitations on the data size enabling straightforward manipulation of text files. Feature selection tools in Weka were utilized during data pre-processing and the clustering tools were used to

implement the two-step clustering strategy proposed in the framework. The new algorithms proposed for filtering, aggregating, transferring, consolidating and analysing features were written using Perl.

5. Design of UHAD Framework. The goal of the framework is to detect anomalies in an unsupervised fashion without using any kind of knowledge on attacks or a trained model of network behaviour. This is accomplished by correlating and analysing the event features in heterogeneous logs. The framework consists of two major phases with several components in each phase. The first phase considers individual logs from multiple sources separately, starting from clustering and passes through several components before the events from various logs were transferred to a common format (i.e., Generic Format Log (GFL)), whereas the next phase considers the events in GFL as input, which was clustered and analysed to identify anomalies. UHAD primarily relies on the pattern of log events and its features to detect anomalies, making it applicable to the evolving network traffic environment. The overall framework of UHAD is illustrated in Figure 5.1, and the first and second phases are illustrated in Figure 5.2 and Figure 5.3, respectively.

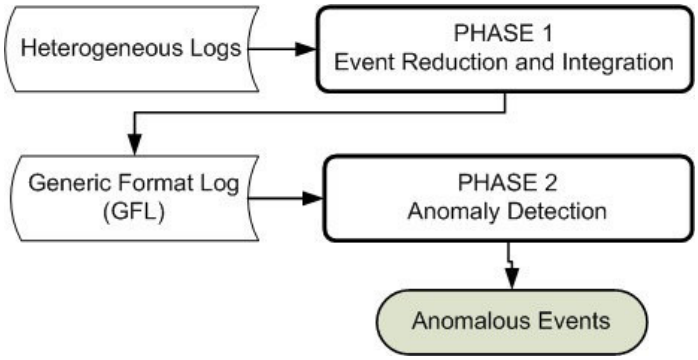


FIG. 5.1. *Overall Framework of UHAD*

Only the Cluster Events component, i.e., component (1) in Figure 5.2 and component (5) in Figure 5.3, utilize existing algorithms available in Weka, while other components are newly proposed algorithms written in Perl.

5.1. UHAD Components. The framework is composed of seven co-operating components, where every component performs a specific task in the process of anomaly detection, and the details of the tasks carried out by each component are described in the following sections.

5.1.1. Clustering Events. The log events were clustered using a two-step strategy and every log was treated separately for this step. The log is subjected to two operations as illustrated in Figure 5.4 and both the operations were implemented using Weka.

Predicting Clusters. Out of the three clustering algorithms (i.e., K-Means, EM and FF) chosen to be used in the framework, EM alone has the capacity to predict the number of cluster that is appropriate for the given dataset. This unique capacity of EM clustering is utilized to implement the first step in our clustering strategy. The Weka [28] implementation of EM algorithm manipulates the number of clusters for a given set of instances using cross validation by separating the instances into a number of partitions called folds. The steps involved in predicting the number of clusters by cross validation are as follows:

1. Number of clusters (k) is set to 1.
2. Training instances are split randomly into 10 folds.
3. EM is executed 10 times with 10 folds by the usual cross validation.
4. Log likelihood is averaged over all the 10 results.
5. If the log likelihood has increased, the number of clusters (k) is incremented by 1 and the process repeats from step 2.

For those training sets containing 10 instances or more, the fold is set to 10 otherwise the number of folds is set to the number of instances. In the first step a given set of events E_{ij} is loaded to Weka to manipulate the

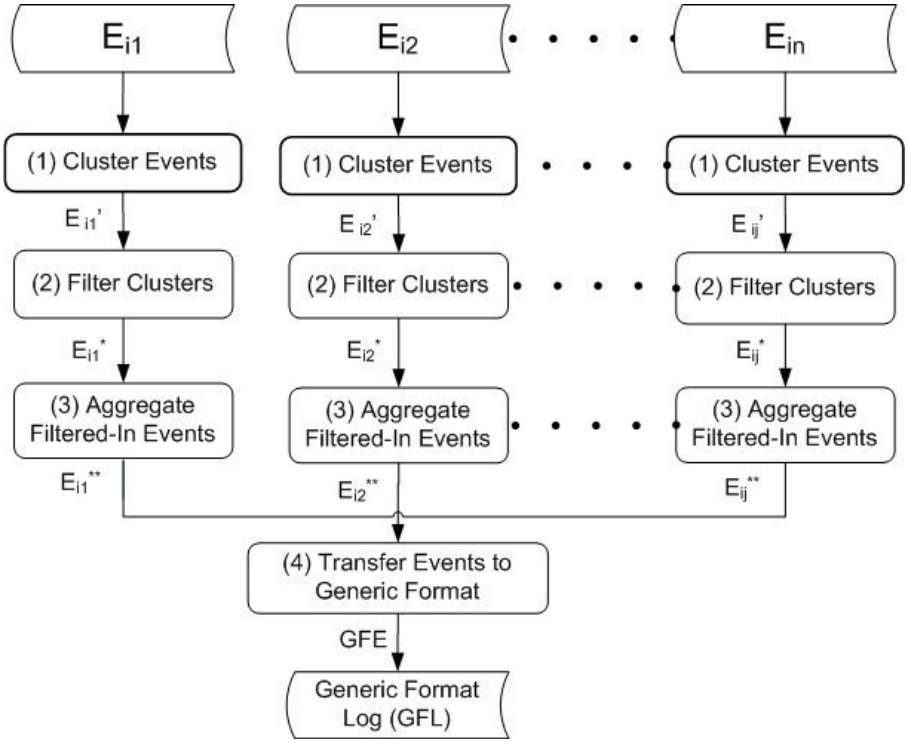


FIG. 5.2. *Event Reduction and Integration in UHAD*

best number of clusters (K_{ij}) using EM. As the intention of this step is to predict the number of clusters, the default seed value 100 was used with the number of clusters (K) set to -1 to enable EM to predict the number of clusters. The clusters (K_{ij}) thus predicted is used to cluster the events (E_{ij}) in the next step.

Generating Clusters. The details of the clustering algorithms used in the second step of the clustering strategy implemented in UHAD is as follows:

Expectation Maximization. The EM algorithm [48] comprises of two recursive steps, Expectation and Maximization, which uses a statistical model called Gaussian finite mixtures to accomplish the objective of producing the most likely set of clusters for a given dataset, given the number of clusters (K). The model includes a set of K probability distributions to provide data representation for each cluster. Each K distribution is defined by parameters like number of iterations and the difference in log likelihood between successive iterations. Initially these parameters are deduced by the algorithm based on the input data, which is subsequently determined by the probability that a particular instance belongs to specific cluster for the given data by utilizing these parameter deduced. Parameter distribution is amended again and this continues until the generated clusters have a certain level of overall cluster goodness or until the maximum number of iterations is reached.

K-Means. K-means [45] is a simple and popular clustering method that divides instances based on the attribute values into K disjoint clusters. Instances that shape the cluster have similar attribute values and K specifies the number of cluster to be generated. The steps of K-means algorithm are as follows:

1. Define the number of clusters K.
2. Initialize the K cluster centroids by randomly dividing all instances into K clusters, calculating their centroids, and verifying that all centroids differs from the other.
3. Iterate on all instances and calculate the distances of centroids for all clusters. Assign each object to the cluster with the nearest centroid.
4. Recalculate the centroids of both modified clusters.
5. Repeat step 3 until the centroids change.

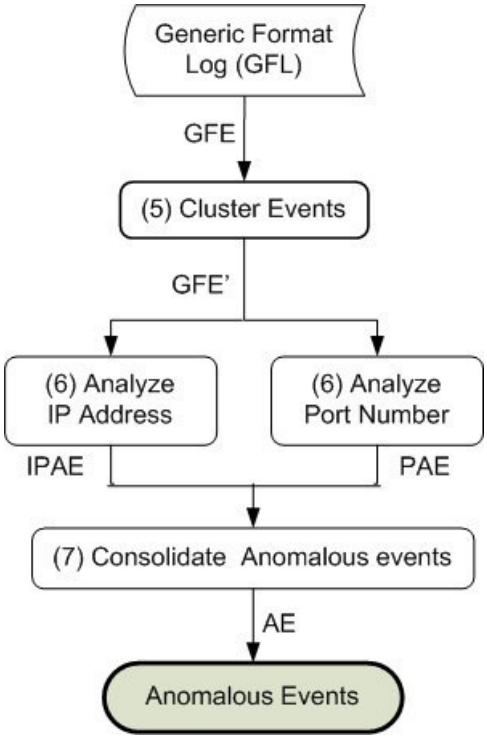


FIG. 5.3. *Anomaly Detection in UHAD*

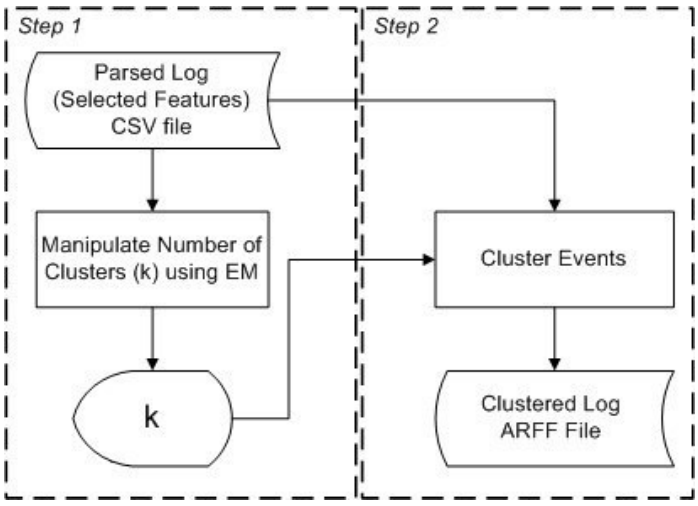


FIG. 5.4. *Two-Step Clustering Strategy*

Moreover, a distance function is needed to calculate the distance (i.e. similarity) between two instances and the most commonly used is the Euclidean distance where every attribute contributes evenly to the calculation of this value. The algorithm has the skill to treat the features in the events equally and segregates precisely in likely clusters. Additionally it has the capacity to handle larger datasets with a lesser processing time justifies the importance of using it in the second step of the clustering strategy.

Farthest First. Farthest first [46] is an alternative of K-Means that places every cluster center in turn at the point farthest from the existing cluster center and this point lies inside the data area. Since there is less reassignment, the process of clustering is faster. The logs (L_{ij}) captured from Apache server, Linux and Snort IDS were clustered according to the specification stated in Experiment 2. Therefore every log was clustered 12 times, and the clusters thus generated were maintained separately for further processing and evaluation. The clustered events were stored in its native format, i.e. Attribute Relation File Format (ARFF) for further processing and not the trained model.

5.1.2. Filtering Clustered Events. The filtering algorithm (Figure 5.5) was implemented using the script written in Perl which receives a clustered file in ARFF format as input and produces the filtered-in events in CSV format. The ARFF file was parsed to eliminate the header generated during clustering and the relevant

```
Algorithm Filtering_IL( )
Input: Clustered log (Eij')
Output: Filtered log (Eij*)
Step 1: Identify the number of events (Eij) and clusters(kij)
Step 2: Caculate the threshold (Et), where Et = Eij / kij
Step 3: Calculate the size of the clusters
        For each cluster Ekijn in Eij' do
           Calculate Cluster Size (kijn)
        End For
Step 4: Identify and extract clusters (Ekijn), whose cluster
        size (kijn) is less than the threshold.
        For each cluster Ekijn in Eij' do
          If  kijn  < Et then
             Eij* ← Ekijn  or  Ekijn ∈ Eij*
          End If
        End For
```

FIG. 5.5. *Filtering Strategy for Individual Logs*

features of every log together with the cluster number were extracted to a CSV file before proceeding with filtering. In some cases, the number of clusters generated was lesser than the number of clusters requested and therefore the parsed clusters were scanned to identify the number of clusters. As such, the threshold was calculated based on the identified number of clusters and the total events clustered. The volume of events and the identified number of clusters varies from log to log and therefore the threshold calculated also varies. The volume of events in every cluster was found and those clusters whose size was less than the calculated threshold (E_t) was filtered-in for further scrutiny as mentioned in the algorithm (Figure 5.5). The cluster an event belongs to was also included in the filtered log to verify the patterns of events in each cluster.

5.1.3. Aggregating Filtered-In Events. Filtered-in log was received as input and the redundant events were combined by checking every event to produce an aggregated log and the algorithm is illustrated in Figure 5.6. Even though the log was previously clustered and filtered, the order in which the log event appears in the original log was maintained. Therefore, every event in the log was compared with the event that is immediately following it. As an analogy, if the current event being scrutinized is equivalent to the previous event automatically the current event is dropped and the next event becomes the current event whereas if the current event is different from the previous event, then the current event is retained and it becomes the previous event.

5.1.4. Transferring Events to GFL. The process of extracting the events from various logs and transferring the events to GFL is automated using custom written script. The events in Apache server, Linux syslog and Snort IDS logs were transferred to GFL excluding Apache SSLError and Linux mail log, since many of the GFL features were not available in these logs; but they are maintained separately to be used at some stage in the analysis. As IPTables firewall log was also used only during analysis, these events were stored separately

```
Algorithm Aggregating ( )
Input:    Filtered log (Eij*)
Output :  Aggregated Log (Eij**)

// To check and extract unique events
For each event eijn in Eij* do
     If  eijn ≠ eij(n-1) then
          Eij** ← eijn or eijn ∈ Eij**
     End If
End For
```

FIG. 5.6. *Aggregating Strategy for Individual Logs*

as per the features specified in GF. The transfer of events from various log sources is depicted in Figure 5.7. In some logs, source IP address was not recorded as separate feature, but then as a part of another feature

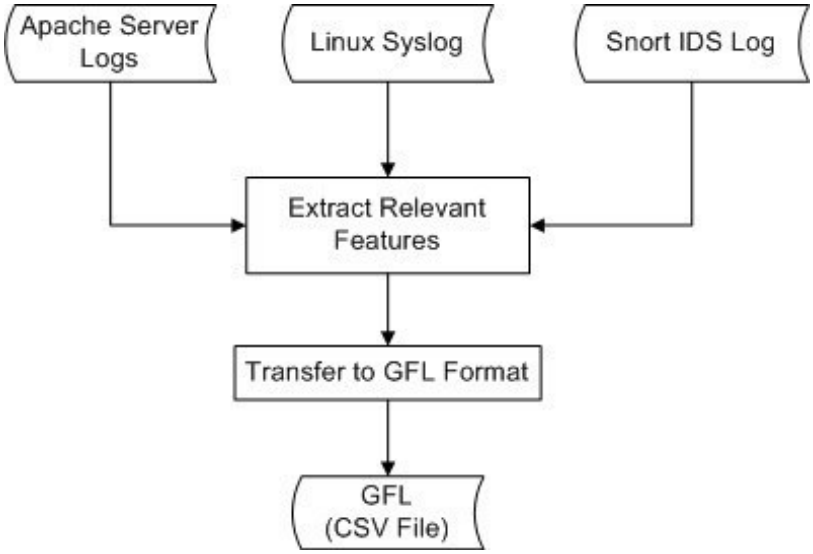


FIG. 5.7. *Transferring Events to GFL*

and therefore it was extracted and represented as a separate feature. Since we compare the performance of three algorithms with four settings, the GFL events were maintained separately according to the algorithm and parameter settings.

5.1.5. Clustering GFE. The GFL events which was maintained according to clustering algorithms and their respective settings were considered separately for this step. In the first step a given set of GFL events (GFE) is loaded into Weka to manipulate the ideal number of clusters (K_g) using EM. The default seed value 100 was used with the number of clusters (K) set to -1 to enable EM to predict the number of clusters. The clusters (K_g) thus predicted is used to cluster GFE. The respective GFL events were clustered using the same algorithm and parameter settings. The clustered events were stored in ARFF format for further processing.

5.1.6. Analyse Features to Identify Anomalous Events. The analysis process intends to discover the relationship between events pertaining to the various features represented by the events to detect anomalous events. We performed analysis using the features IP address (f_{sip}; f_{dip}), port number (f_{sp}; f_{dp}) and cluster number (f_k).

IP Address Analysis. The analysis procedure (Figure 5.8) receives clustered log events (GFE) as input which was scanned several times, before an event was decided as anomalous. During the analysis process three features were manipulated which are source IP address (f_{sip}), destination IP address (f_{dip}) and cluster number (f_k). Although checking the IP address of the events was focused, the cluster number of these events was also

```
Algorithm IPAddressAnalysis( )
Input:  An array GFE' consisting n clustered events
Output: An array IPAE consisting n anomalous events
Step 1: To identify unique IP Address in the GFE'
        For each event GFe_n in GFE' do
            If f_sip != 11.11.79.* then
               Add f_sip to iplist[j] if f_sip not available in iplist[j].
            Else if f_sip == 11.11.79.* then
               Add f_dip to iplist[j] if f_dip not available in iplist[j].
            End If
        End For
Step 2: To identify every unique IP Address in logs from multiple
         sources
        For each element in iplist[j] do
        For each event GFe_in in GFE' do
        //GFe_in refers to all events belonging to a particular source S_m
        If iplist[j] exists in (GFe_1n and GFe_2n) or (GFe_1n and GFe_2n) then
           ipselected[j] = f_sip of GFe_in
        End If
        End For
        End For
Step 3: To identify the cluster of the selected IP's
        For each event GFe_n in GFE' do
           If  ipselected[j] matches with  f_sip OR f_dip then
               Add f_k to clusters[j] if f_k not in clusters[j]
           End If
        End For
Step 4: To extract the events belonging to the clusters of the
           selected IP's
        For each event GFe_n in GFE' do
           If  clusters[j] ==  f_k of GFe_n then
               IPAE ← GFe_n or GFe_n ∈ IPAE
           End If
        End For
```

FIG. 5.8. *IP Address Analysis*

manipulated to get a big picture of the events related to intrusions. The first pass was to identify the unique IP address in log events, where f_{sip} was checked in case of inbound connections and f_{dip} for outbound connections. Those identified unique IP addresses existing in Apache and Linux logs were checked for its existence in Snort IDS log, and vice versa. Since there were no labels specified in GFL to identify the log which an event belongs to, the events were identified by the non-availability of feature values in that event, i.e., Apache and Linux log events do not have values for the features destination IP address (f_{dip}), source port (f_{sp}), destination port (f_{dp}) and protocol (f_{pr}).

The events containing IP addresses that were selected for its existence in multiple log sources, were checked to identify the cluster which they belong to. All the events belonging to the identified clusters were extracted and deemed as anomalous which serves as the result of the analysis. The results were evaluated based on the volume of anomalous events identified by the analysis and also the number of IP addresses covered by these events from where the intrusion have originated. To verify the validity of the events detected, it was compared with the results of SOTM#34 challenge, to substantiate the coverage of anomalies by UHAD.

Port Number Analysis (PA). Ports possess significant discriminative control in classifying certain types of traffic when used with other features [51] and most anomalous activities utilises the unused and unassigned ports. The same set of GFL events (GFE) used for IP address analysis were used for port analysis, too. Since

the analysis (Figure 5.9) was based on port numbers, those events without it will be automatically excluded from analysis. In the case of inbound connections the source port number of the events was checked against

```
Algorithm PortNumberAnalysis( )
Input:       An array GFE' consisting of n clustered events
       An array UAPN consisting of a set of Unassigned Port Numbers
Output: An array PAE consisting n anomalous events
For each event GFe_n in GFE' do
      For each port number P_j in UAPN do
            If f_sip != 11.11.79.* then
            If  f_sp == P_j  then
                 If  f_dp ==  20 OR 21 OR 22 OR 23 OR 25 OR 80 then
                     PAE ← GFe_n or GFe_n ∈ PAE
                 End If
            End If
            Elsif f_sip == 11.11.79.*
            If  f_dp == P_j  then
                PAE ← GFe_n or GFe_n ∈ PAE
            End If
            End If
       End For
End For
```

FIG. 5.9. *Port Number Analysis*

the listing of the dynamic and unassigned port numbers as per Internet Assigned Numbers Authority (IANA). In this manner, the destination ports of the matching events were checked for the availability of well-known port numbers, similar to 20, 21, 22, 23, 25 and 80 to generate anomalous event list, as intrusive events mostly succeed through these ports on the victim host. Then again for outbound connections, the destination port number of the events was checked against the listing of the dynamic and unassigned port numbers according to IANA. All the matching events were regarded as anomalous and was moved to a distinct CSV file.

5.1.7. Consolidate Anomalous Events. Three operations were performed, namely uniting, associating and concentrating to consolidate the anomalous events detected by the analysis methods, i.e. IPA and PA, and the process is illustrated in Figure 5.10. Initially all the anomalous events detected by IPA was copied to a CSV file referred as an intermediate anomalous events list (iAE) and every single event in the list is referred as iAe. Every anomalous event detected by PA was compared with every single anomalous event of IPA and non-matching anomalous events of PA will be appended to iAE. Unique IP addresses excluding internal IP addresses, i.e., 11.11.*.*, represented by the iAE events were identified and every such IP address was compared with IP address of IPTables firewall log events maintained in a separate log as per GF. All the IPTables firewall log events containing IP address matching with the identified IP addresses were appended to iAE. Now iAE contains the detected anomalous events (AE) captured from various sources that serves as the result of the framework. In order to concentrate on significant anomalous events, a threshold (A_t), i.e., 3, 6, 9, was used to consolidate the anomalous events based on the IP address. Since the unique IP addresses were already found before correlation, the occurrence of the iAE events containing these IP addresses were counted. Those IP addresses satisfying the threshold (A_t) were identified. The iAE events were checked for these IP addresses, and those events satisfies the threshold were extracted and maintained separately. This to verify that, whether applying the threshold at this point supported to focus on the most significant anomalous events or it reduced such events. All the new algorithms proposed to implement various components of the framework were written in Perl.

The limitation of the filtering component deployed in UHAD [10] is further improved with the refined filterer [52] by increasing the volume of retained abnormal events; hence, the other components of the framework [52] in are basically the same as UHAD [10]. The aim of the refined filterer is to retain all the abnormal events in the log for subsequent processing, irrespective of the existence of such events in larger number in the logs and the inaccuracies in clustering. The refined filterer receives a clustered log (E_{ij}) which is initially scanned to

```
Algorithm Consolidate_AnomalousEvents( )
Input: An array IPAE consisting of anomalous events of IP analysis
       An array PAE consisting of anomalous events of port analysis
       An array IPTGFE consisting of n events
Output: An array AE consisting n anomalous events
Step 1: Transfer all the IPAE  to iAE //iAE: Intermediate Anomalous
         Events
Step 2: Transfer events in PAE to iAE that is not available in IPAE
        For each PAe_y in PAE do
          For each event IPAe_x in IPAE do
           If PAe_y  != IPAe_x then
              iAE ←   PAe_y // iAE = Intermediate Anomalous Events
           End If
          End For
        End For
Step 3: To identify unique IP Address in iAE
         For each event iAe_x in iAE do
           If f_sip != 11.11.79.* then
              Add f_sip to iplist[j] if f_sip not available in iplist[j].
           Else if f_sip == 11.11.79.* then
              Add f_dip to iplist[j] if f_dip not available in iplist[j].
           End If
         End For
Step 4: To extract the events form IP Tables whose IP matches with the
        iplist[]
         For each event IPTGFe_n in IPTGFE do
          For each element in iplist[j] do
               If (f_sip == iplist[j] OR f_dip == iplist[j]) then
                  iAE ←   IPTGFe_n
              End If
          End For
         End For
Step 5: Count the appearance of each IP Address in the events iAE
        For each event iAe_x in iAE do
         For each element in iplist[j] do
            If iplist[j] == f_sip OR iplist[j] == f_dip then
             ipcount[j]++;
            End If
         End For
        End For
Step 6: Select the IP's that satisfies a specific threshold (A_t), i.e.,
        3, 6, 9
        For each element in iplist[j] do
         If ipcount[j] >= A_t then
             selectedip[k] = iplist[j];
         End If
        End For
Step 7: Extract the events of this selected IP's and deem as anomalous.
        For each event iAe_x in iAE do
          If selectedip[k] == f_sip OR selectedip[k] == f_dip then
                AE ←   iAe_x
          End If
        End For
```

FIG. 5.10. *Comparing Anomalous Events*

identify the volume of events (nE_{ij}) and the number of clusters (K_{ij}) to calculate the filtering threshold (E_t), i.e., $E_t = nE_{ij}/\ K_{ij}$.

The filtering threshold calculated as such is equivalent to the average cluster size. The calculated filtering threshold was used to identify the sparse and dense clusters in the clustered log, and the events of sparse ($E_{ij}{}^*$) and dense clusters ($E_{ij}^{\#}$) were identified, extracted and maintained separately. But then, there are chances that some of the abnormal events may have been mixed up with normal events in dense clusters due to the

inaccuracy in clustering or the similarities between the feature values of abnormal and normal events. Therefore, such abnormal events in dense clusters must be identified and included in filtered-in events for further processing.

In order to achieve that, subsequently, every event in the dense clusters was compared against every event in the sparse clusters, i.e., the respective features in these events were separately or individually compared. The cluster number in the log that was included during the process of clustering, which also appears in the filtered events is excluded from the counted features for a particular log. Those events in the dense clusters which match with any one of the events in sparse clusters (abnormal events) are considered abnormal, and the matching events ($E_{ij}^{\#}$) i.e., those events having a match of at least 50% of features in between the events, were added to the set of filtered-in events after all such events were identified. For instance, if there are six features in the log events compared, then at least three features of an event in the dense clusters must exactly match with any one of the events in the sparse clusters in order to be filtered-in for further processing. As there may be subtle difference in the patterns of abnormal events grouped in sparse and dense clusters, comparing both categories by features for exact matches is not reasonable and will not assist in retaining majority of the abnormal events. Therefore, 50% is chosen as the minimum matching between events in order to consider for further processing. This is based on the assumption that the sparse clusters contain only abnormal events and the misclassified abnormal events in dense clusters should be partially or completely matching with any of the events in sparse clusters. Therefore, the dense cluster events were checked against sparse cluster events, and the matching events of dense clusters were also filtered-in together with sparse clusters for further examination.

Moreover, the refined filterer strives to retain every abnormal event in the log irrespective of the inaccuracy in some of the clusters generated. The refined filterer is very much process intensive as every event in the dense clusters has to be compared with every event in sparse clusters and moreover every feature of the event is individually compared with another event. Additionally, the logs clustered with the reordered events were also filtered using the refined filterer to verify whether the filterer managed to retain the abnormal events as it does with the clustered events of original log. This is to evaluate the effect of reordering and the subsequent clustering on the refined filterer especially on the retention of abnormal events.

6. Results and Discussions. To assess the capability of the framework in discovering anomalies, the log events captured by multiple sources in a Honeynet system were used. A total of four subsets with varying duration were tested. First and foremost, the idea behind the development of this framework is to detect a wide range of anomalous events by analysing various logs without using any kind of knowledge on anomalies or the models of network traffic behaviour. All the components implemented in this framework work towards the aforesaid objective. Apart from this, every component has a specific objective towards the main objective, and hence the performance of every component was measured with different metrics as stated in Chapter 3. The performance of the framework as a whole in detecting various anomalous events were evaluated by comparing its results with the SOTM/#34 challenge results [12, 13, 14, 15] provided at Honeynet.org.

6.1. Clustered Events. This section describes the results of the two-step clustering strategy implemented in the framework. Firstly, the predicted clusters for various logs were discussed. Secondly, the accuracy achieved with various clustering algorithms with different parametric settings were compared. Additionally, the relation of clustering accuracy with the actual formation of clusters with respect to abnormal events were examined.

6.1.1. Predicted Clusters. The clusters for each log were predicted according to Experiment 1 specification. EM clustering was applied with the default values for the parameters (i.e. K = -1, seed = 100) on the selected features of the individual logs (E_{ij}) to estimate the best number of clusters (K_{ij}). For example, while predicting the number of clusters for Apache Access log, EM at the start selects eight clusters by cross validation, but concludes generating only five clusters (0, 1, 2, 6, 7) as shown in Figure 6.1.

Therefore, the best number of clusters selected for this log was five, and the similar approach was followed for all the logs in manipulating the ideal number of clusters. The cluster predicted by EM for various logs is provided in Table 6.1. This reveals the fact that the number of clusters predicted were not influenced by the number of features and events in a log, but by the patterns of the events, i.e, the more the number of distinct patterns, the higher the number of clusters predicted. Apache error log consisting of 433 events with five features resulted in seven clusters whereas Snort IDS log consisting of 4423 events with 11 features resulted in only three clusters. This also revealed the fact that there is no relation between the number of events and

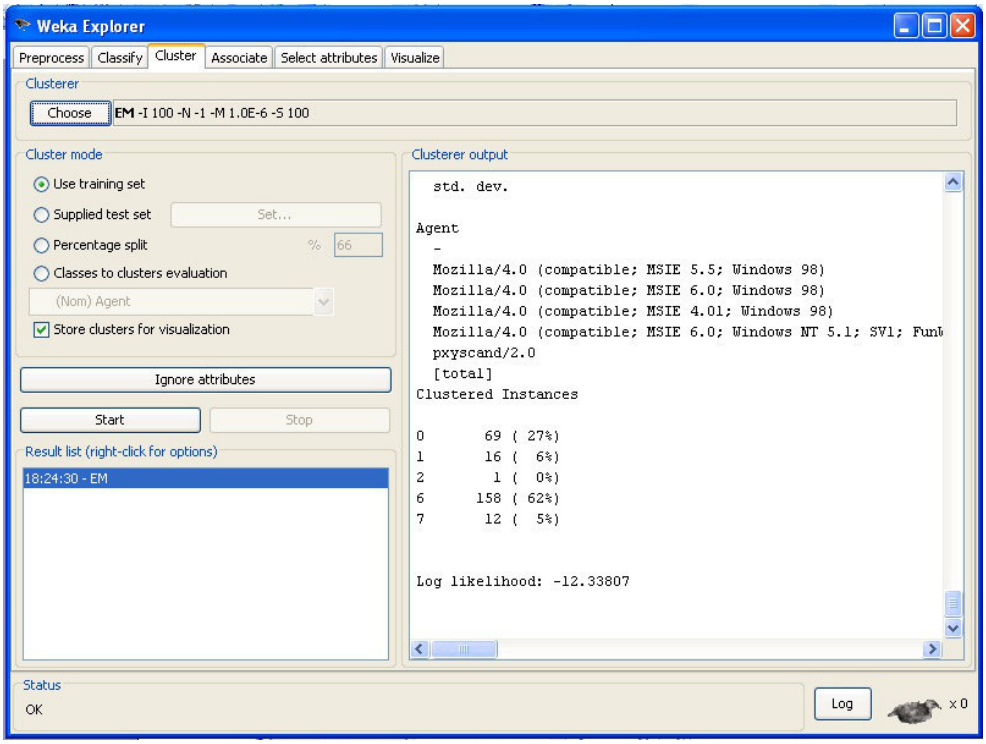


FIG. 6.1. *Manipulating Clusters using EM*

TABLE 6.1
Ideal Clusters by EM

Log Type	Suset-1	Subset-2	Subset-3	Subset-4
Access	5	5	8	6
Error	7	6	7	7
SSL-Error	2	4	4	3
Message	5	5	5	7
Mail	2	2	4	4
Security	4	4	4	6
Snort IDS	3	8	6	7

its features with the clusters predicted.

The time taken to predict the ideal number of clusters by EM for each log varied from seconds to hours depending on the volume of events and its features. The time taken to manipulate the events to predict clusters for Linux Message log with 112 events took five seconds whereas Snort IDS log with 17049 events took 1 hour 30 minutes and 12 seconds. Moreover, prediction time also depends on the existence of varying patterns in the log, eventually resulting in more number of clusters, e.g., Linux message log with 105 events took 10 seconds to predict seven clusters, whereas 112 events belonging to the same type of log took five seconds to predict five clusters. The manual effort and time spent in finding the ideal clusters for a particular log using our strategy is far lesser than that of the usual method, i.e., applying the clustering algorithm on a dataset several times with different clusters (K) and choosing the best clusters among them. Thus the clustering strategy implemented in our framework facilitates the process in finding the appropriate clusters for the logs with less time consumption and manual effort.

6.1.2. Clustering Accuracy and Cluster Analysis. Using the clusters (K_{ij}) manipulated from step 1 (Experiment 1), the logs were clustered according to the algorithms and settings stated in Experiment 2, and the clustered events (E_{ij}) were recorded in ARFF format. Every clustered log was maintained separately according to subsets, algorithms and settings. The accuracy of the clusters generated for various logs by the algorithms with different settings were evaluated. The maximum time taken to cluster 17049 events of Snort IDS with seven clusters, i.e., the log containing the highest number of events in subset-2, was 2 minutes and 5 seconds by EM whereas K-Means and FF took only 17 seconds and 2 seconds, respectively. The accuracy of K-Means, EM and FF in clustering various logs with different parameter settings was evaluated, i.e., comparing the accuracy of the default setting (setting-1) with other settings (settings 2, 3 and 4). This is to recognize the impact of seed and K on accuracy, when increased. In addition, the relation of accuracy and the parametric settings with the actual grouping of abnormal events in separate clusters were also examined. As we have tested four subsets containing various logs, the results are discussed subset by subset. Previously the results of subset-1 was presented in Hajamydeen et.al. [7], and therefore the results of the other three subsets, i.e., subset-2, subset-3, subset-4, were only discussed.

Subset-2. The accuracy achieved with various settings for this subset exhibits a similar pattern like subset-1 and is illustrated in Figure 6.2. Increasing the seed value alone (setting-2) improved the accuracy in all logs

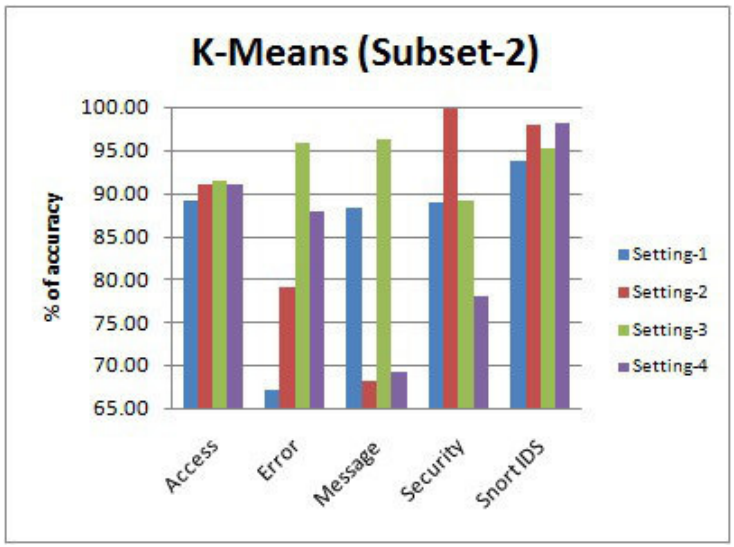


FIG. 6.2. *K-Means Clustering Accuracy (Subset-2)*

except the Linux Message log. This was because of the similarity between the feature values in this log. Doubling k with the default seed (setting-3) increased the accuracy with all the logs. But, increasing both seed and k together (setting-4) decreased the accuracy with all logs except Snort IDS logs, thus signifying the sensitive nature of K-Means to initialisation parameters. This also shows the need for customised seed value than using the default seed especially with those logs having events of bigger volume. The increase in accuracy with settings 2 and 4 for this log also shows the importance of a customised seed value especially with bigger sized logs. There were 14 successful intrusive events and 58 unsuccessful intrusive events in the Apache error log.

Analysing the clusters generated with various settings showed that setting-4 formed better clusters by grouping the intrusive events in separate clusters. There were 12 clusters generated from 0 to 11 consisting of events 120, 54, 75, 14, 1, 60, 67, 13, 1, 24, 33 and 142, respectively. All the 14 successful intrusive events were grouped in Cluster 3 and the 58 unsuccessful intrusive events with another nine abnormal events in Cluster 6. This was due to the impact of increased seed and k on clustering even though the accuracy with various settings did not reflect that. The accuracy of EM with subset-2 as illustrated in Figure 6.3 expressed a reverse accuracy pattern with respect to settings compared to subset-1. The accuracy of clustering was improved when the ideal clusters (k) were doubled (setting 3) with all the logs, and declined when the seed was increased with settings

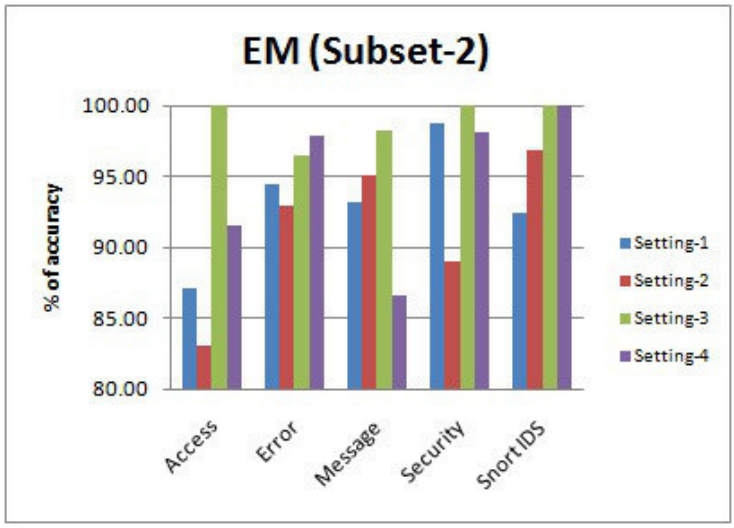


FIG. 6.3. *EM Clustering Accuracy (Subset-2)*

2 and 4 with most of the logs. This shows that the accuracy factor is not only influenced by k and seed, but also by the volume of events and their patterns. A better cluster formation was achieved by EM with setting-4 for the Apache error log. Out of the 12 clusters (107, 51,39, 54, 24, 0, 47, 60, 26, 54, 142 and 0) generated for this log, two of them were empty clusters, i.e., Clusters 5 and 11. The 58 unsuccessful intrusive events were grouped together in Cluster 7 with another two abnormal events. The 14 successful events were joined together with another 93 abnormal events. In terms of cluster formation, K-Means formed better clusters compared to EM, even though EM achieved the highest accuracy for this log.

The clustering accuracy of FF with subset-2 (Figure 6.4) expressed a similar pattern like subset-1 with settings 3 and 4, however, the pattern was opposite with setting-2 which resulted in a sharp decline in accuracy with Apache error, Linux message and Snort IDS log. Although FF is not very sensitive to seed values, the

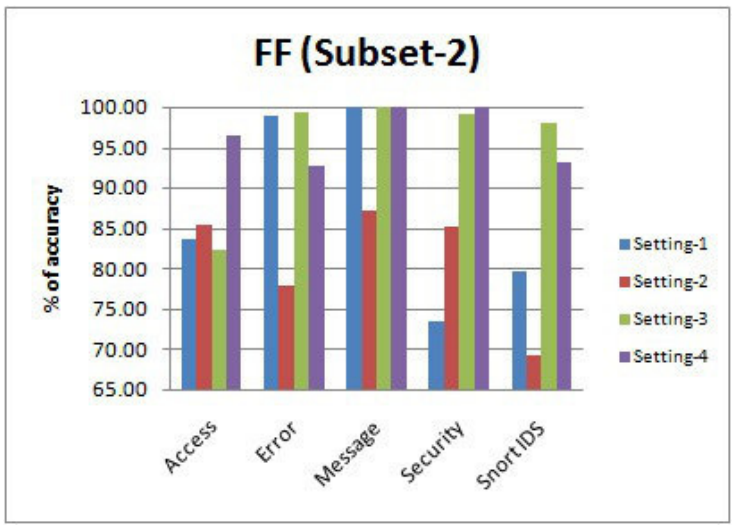


FIG. 6.4. *FF Clustering Accuracy (Subset-2)*

sharp increase and decrease in accuracy with settings 2 and 4 was due to the diverse event patterns in the logs. Moreover, the cluster formation for the logs was similar between settings 1 and 2 and also with settings 3 and 4.

The unsuccessful intrusive events of Apache error log were grouped together in Cluster 4 and successful ones in cluster 6 together with other abnormal events. Doubling k with settings 3 and 4 resulted in better formation of similar clusters. The clusters generated by FF was better than EM but inferior to K-Means. This subset contained more events in all the logs compared to subset-1 and KMeans formed better clusters for this subset. This shows the capacity of K-Means in handling voluminous log events.

Subset-3. The accuracy of K-Means with subset-3 is illustrated in Figure 6.5. A similar accuracy pattern like subsets 1 and 2 was achieved with settings 3 and 4 for this subset. But with setting-2, the increase in seed value decreased the accuracy. This shows the sensitive nature of K-Means to initial parameters. Although, the

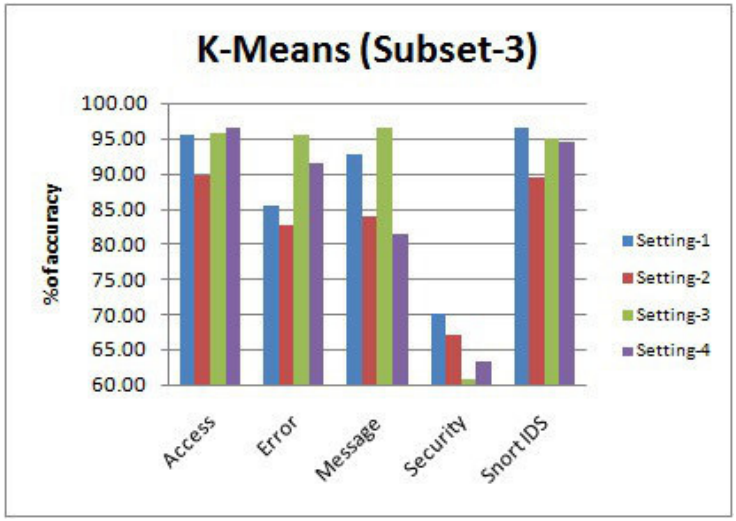


FIG. 6.5. *K-Means Clustering Accuracy (Subset-3)*

volume of events in Apache access and error log was high compared to subsets 1 and 2, K-Means achieved a better accuracy for this log than the other subsets. This exposes the capacity of K-Means in handling larger datasets. Linux security log consists of 234 events where 12 events are related to xinetd crash and the pattern of these events were very different from others.

Even though there were eight features recorded by this log, the majority of the events do not have values for all these features resulted in a lower accuracy, i.e., below 70%. This was also reflected appropriately in cluster formation. All these intrusive events were joined together in Cluster 0, i.e., the biggest cluster, for the settings 1, 2 and 4. For setting-3, these events was separated in two clusters, i.e, Cluster 0 and 4. All the 10 events in Cluster 4 was related to xinetd crash, and another two events were in Cluster 0, i.e, the biggest cluster. Although, the accuracy achieved for this setting is the lowest, the cluster formation was better. This also shows that cluster formation was not directly reflected in accuracy.

In the case of Snort IDS logs consisting of 5013 events, all the events pertaining to the intrusion were grouped in smaller clusters for all the settings. Especially, K-Means with setting-1 grouped majority of the intrusive events in the smallest cluster, i.e., Cluster 5, and in several smaller clusters for setting-3. This shows that the default seed performed better in cluster formation for this log and is also reflected in accuracy. The pattern of accuracy achieved by EM with subset-3 (Figure 6.6) is similar to subset-1 for settings 2 and 3 whereas it was similar to subset-2 for setting-4. The default setting (setting-1) achieved the highest accuracy for Apache access and Linux message log. The increase in seed and k (setting-4) improved the accuracy in Snort IDS log, exposing the contribution of seed and k together with bigger sized logs. In spite of the absence of values for the features in most of the events of Linux security log, EM formed better clusters compared to K-Means, showing its capacity in handling events of this nature.

All the 12 events related to xinetd crash were grouped together in a smaller cluster for settings 1, 2 and 3. For setting-4, 10 of these events were grouped in a separate cluster and another two events in a smaller cluster.

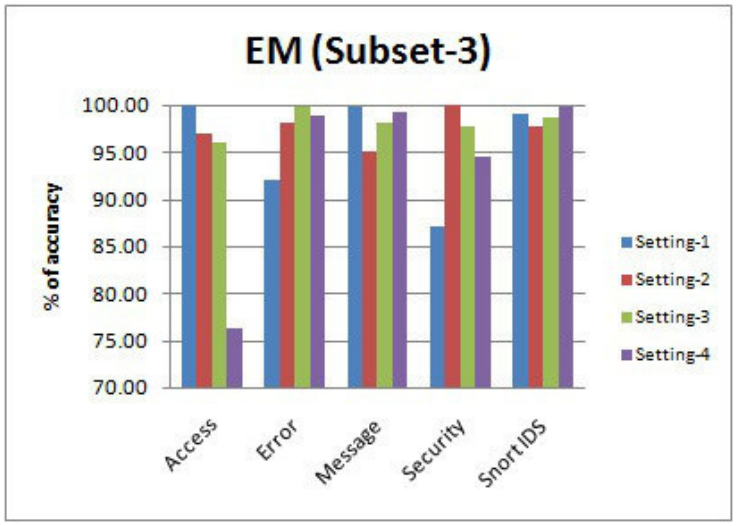


FIG. 6.6. *EM Clustering Accuracy (Subset-3)*

All the intrusive events captured by Snort IDS log were grouped in smaller cluster with setting-4. This shows the involvement of seed and K together in cluster formation. With subset-3, FF achieved accuracy (Figure 6.7) of identical pattern like subset-1. An average accuracy above 90% was achieved by FF for various settings with all the logs, except the Snort IDS log for the default setting. The accuracy achieved by FF with various settings were not appropriately reflected in cluster formation. Intrusive events related to xinetd crash in Linux security

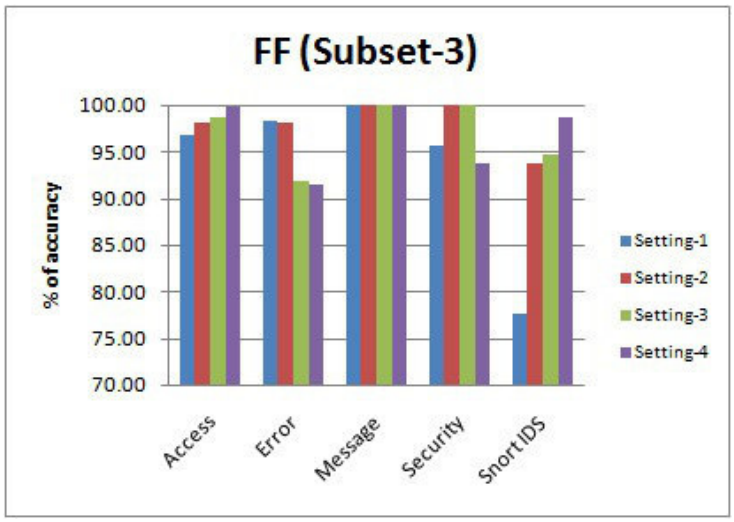


FIG. 6.7. *FF Clustering Accuracy (Subset-3)*

log that has a unique pattern than the other events were grouped in the biggest cluster for all the settings. Even in Snort IDS log, the events related to various intrusions were grouped in bigger clusters.

Subset-4. The accuracy achieved by K-Means for this subset plotted in Figure 6.8 exposed a similar pattern with subset-3. Even though doubling K (setting-3) increased accuracy, increasing the seed together with doubled K did not improve the accuracy. This is due to sensitive nature of K-Means to initialisation parameters. There were 434 events in Apache access log, out of which 78 events were the outcomes of unsuccessful intrusions originated from two IP addresses and the patterns of events from these IP addresses were different from each

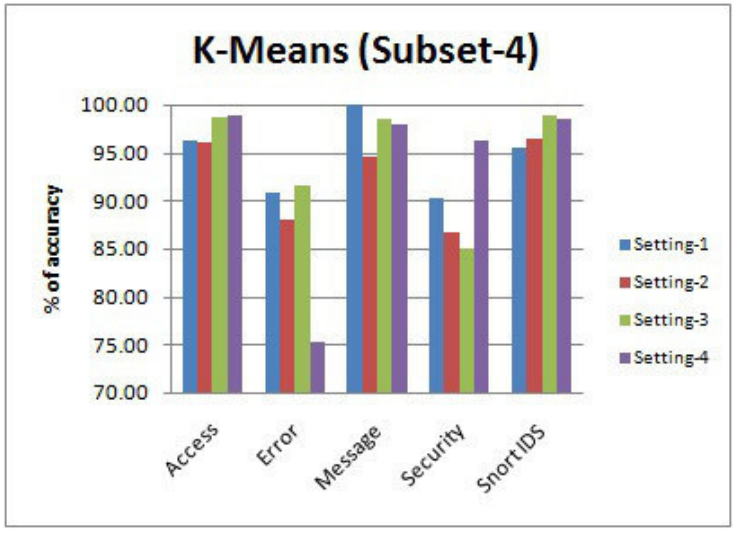


FIG. 6.8. *K-Means Clustering Accuracy (Subset-4)*

other. Settings 3 and 4 generated better clusters by separating intrusive events in two clusters according to IP address. One of these clusters contained only intrusive events and the other cluster contained intrusive events together with other events as well. In case of Snort IDS log, majority of the events related to intrusions were grouped in smaller clusters and a small number of such events gathered in big clusters together with other events for all the settings.

Better clusters were generated with settings 1 and 2 for this log, although settings 3 and 4 showed better accuracy. EM achieved the highest average accuracy of 96.89% for this subset, but the pattern of accuracy with various settings were very different from all the other three subsets and exactly opposite to subset-3. The accuracy achieved by EM for various logs is depicted in Figure 6.9. Better cluster were formed by EM with setting-4 for Apache access log, but the cluster formation was inferior to the ones generated by K-Means. In case of Snort IDS log, all the events related to intrusions were grouped in smaller clusters for the same setting. Like EM, FF also achieved the highest accuracy (Figure 6.10) with this subset, but then the pattern of accuracy

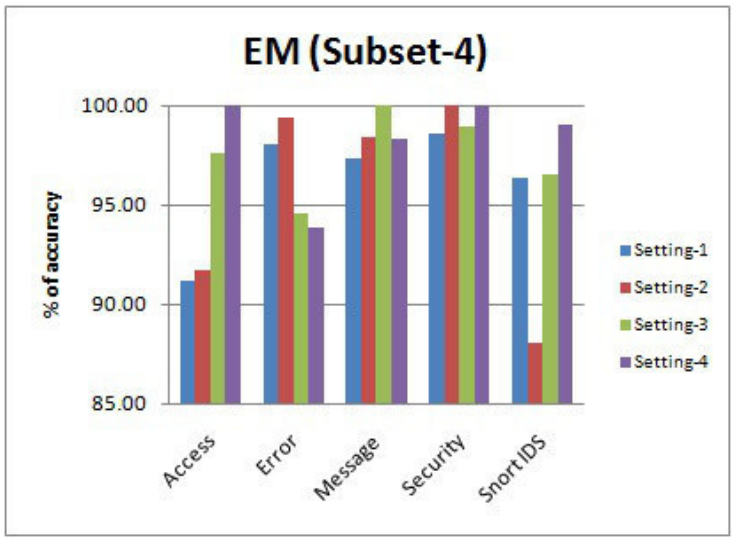


FIG. 6.9. *EM Clustering Accuracy (Subset-4)*

with respect to settings was similar to the achievement of K-Means with this subset. Analysing the clusters

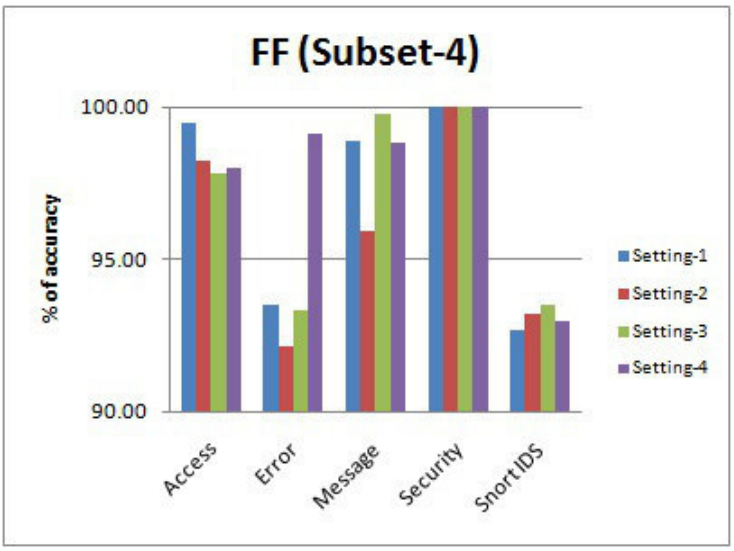


Fig. 6.10. *FF Clustering Accuracy (Subset-4)*

generated by FF with settings 3 and 4 for Apache access log revealed that all the unsuccessful intrusive events were precisely grouped in separate clusters and the cluster formation was better than K-Means. Like subset-3, all the intrusive events captured by Snort IDS log were grouped in bigger clusters for all the settings together with other events. The poor cluster generation of FF for this log with various settings were also reflected in the accuracy.

Analysing the clusters generated with the four log subsets tested provides the following conclusions:

1. Although the clustering strategy implemented achieved a better accuracy, none of the algorithm showed a consistent performance in terms of accuracy and cluster formation with all the subsets. Better clusters were formed by EM with subsets 1 and 4, and K-Means with subsets 2 and 3. This was due to the volume of events contained in these subsets for various logs.
2. Accuracy and cluster formation by a particular clustering algorithm was inconsistent with different logs in the same subset. This was due to capacity of the algorithm in handling a particular type of data as the number of features and the content of features varies with logs. Even in some logs the values for certain features were not available in most of the events.
3. Usage of settings 3 and 4 improved cluster formation with most of the logs. This was due to the increase in k which grouped the abnormal events precisely in separate clusters. EMs performance with this setting varied with subsets was due to its nature of forming clusters based on the available patterns in the log irrespective of the requested clusters. The cluster formation by FF with settings 3 and 4 were similar and this was due to the non-reactive nature of FF to seed values.
4. Examining the clusters generated shows that, the accuracy was not directly reflected in the formation of clusters. This is because calculation of accuracy is based on the context of the algorithms capacity in handling data, whereas the clusters produced were analysed based on the formation of abnormal events in separate clusters within the context of the research.
5. K-Means performed better even with the big sized logs, showing its capacity in handling larger datasets.
6. EM performed better with the logs where the values for certain features were not available in most of the events in the log.
7. FF generated better clusters with those logs where the events patterns were qualitatively different. This is due to its nature of calculating the cluster centroids in successive iterations which is opposite to that of K-Means.

Overall, the clustering strategy implemented in this phase achieved a better accuracy and cluster formation.

Due to the inconsistent performance of a particular algorithm and setting with various logs, deploying multiple algorithms and settings becomes necessary. This phase could be further extended by designing a recommender method to suggest the best clusters of various logs among those generated with various algorithms and settings, for further processing.

6.2. Filtered Events. In this phase, the clustered logs were filtered using a calculated threshold and the filtered-in events were retained for further process. The percentage of events reduced and the volume of abnormal events retained from the clustered logs with the three algorithms and its settings were evaluated (Experiment 3).

6.2.1. Event Reduction. Every clustered log was scanned to identify the number of events and clusters to calculate the threshold, and those clusters which satisfies the threshold were filtered-in.

Subset-2. The volume of events in various logs were relatively higher than that of subset-1 and especially the Snort IDS log contained 17049 events. Applying the threshold on various clustered logs for this subset reduced an average of 71% events and retained an average of 29% events for further examination and is provided in Table 6.2. Applying the threshold on Apache access log clustered by FF with setting-4 reduced 93.88% events whilst retaining 6.12% events, which was due to the better clusters formed by FF for this log.

TABLE 6.2
Percentage of Reduction (Subset-2)

Algorithms	Settings	Access	Error	Message	Security	Snort IDS
K-Means	Setting-1	75.51	57.28	68.87	86.09	72.79
	Setting-2	38.03	67.22	84.77	96.62	79.36
	Setting-3	77.74	85.60	67.55	79.70	74.86
	Setting-4	69.02	85.76	78.81	89.47	65.40
EM	Setting-1	57.88	60.60	64.24	81.95	55.90
	Setting-2	74.77	56.79	52.98	69.17	63.78
	Setting-3	79.78	61.42	52.98	68.42	67.44
	Setting-4	70.69	41.23	52.98	65.41	59.86
FF	Setting-1	76.25	76.32	68.87	82.71	56.57
	Setting-2	86.09	65.56	68.87	68.42	67.26
	Setting-3	85.16	77.98	76.82	92.48	82.62
	Setting-4	93.88	71.69	76.82	92.48	77.53

Applying the threshold on Apache error log clustered by K-Means with setting-4 reduced 85.76% events whilst retaining 14.24% events was due to the better cluster formation achieved by K-Means for this log. This also reveals the fact that better cluster generation will increase the volume of reduction which in turn reduces the processing overhead of the subsequent framework components.

Subset-3. The volume of events in this log subset was higher than subset-1, especially the events in Apache access and error log. Application of threshold on this log subset reduced an average of 73% events while retaining 27% events and the details of the event reduced with various logs is provided in Table 6.3. In some cases, due to the availability of more abnormal events in the log it was grouped in bigger clusters. Usage of the calculated threshold to filter the clustered events has reduced such abnormal events in bigger clusters making it unavailable for subsequent examination. For instance, FF produced better clusters for Apache access and error log, resulting in reducing more than 85% of the events from these logs. But, this reduction has removed most of the abnormal events in bigger clusters that did not satisfy the threshold.

Subset-4. The volume of events in this log subset was higher than subset-1, especially the events in Apache access and error log. Application of threshold on this log subset reduced an average of 74% events while retaining 26% events, and the details of the events reduced with various logs is provided in Table 6.4. Applying the threshold on Apache access log clustered by K-Means with setting-3 reduced 81.9% of events. But then all

TABLE 6.3
Percentage of Reduction (Subset-3)

Algorithms	Settings	Access	Error	Message	Security	Snort IDS
K-Means	Setting-1	72.27	75.67	74.76	81.62	65.53
	Setting-2	71.56	79.76	90.95	74.79	75.80
	Setting-3	68.67	80.00	87.14	76.92	51.53
	Setting-4	53.59	75.28	72.38	61.97	74.93
EM	Setting-1	55.70	67.40	45.71	57.69	78.48
	Setting-2	70.86	85.28	45.71	57.69	78.48
	Setting-3	68.83	78.11	60.48	57.69	62.24
	Setting-4	85.16	64.41	50.00	57.69	74.29
FF	Setting-1	89.30	89.84	60.48	84.62	75.82
	Setting-2	83.91	89.84	60.48	84.62	83.12
	Setting-3	88.91	93.46	64.29	75.21	87.65
	Setting-4	86.88	93.46	64.76	72.22	88.25

the abnormal events in this log were filtered-out by the threshold, due to the formation of such events in bigger clusters.

TABLE 6.4
Percentage of Reduction (Subset-4)

Algorithms	Settings	Access	Error	Message	Security	SnortIDS
K-Means	Setting-1	44.40	71.28	69.52	96.75	70.56
	Setting-2	74.57	89.67	68.57	96.75	50.33
	Setting-3	81.90	80.17	80.00	86.99	65.93
	Setting-4	76.72	80.58	76.19	91.87	65.26
EM	Setting-1	69.18	73.97	64.76	82.11	68.62
	Setting-2	69.18	64.26	71.43	91.06	54.44
	Setting-3	78.66	67.56	64.76	82.11	73.62
	Setting-4	61.64	54.55	74.29	82.11	71.09
FF	Setting-1	43.97	67.15	83.81	82.11	83.43
	Setting-2	54.09	62.81	91.43	82.11	80.24
	Setting-3	62.28	72.93	79.05	91.06	83.07
	Setting-4	72.41	73.55	79.05	91.06	83.47

6.2.2. Abnormal Event Retention. The abnormal events filtered-in by the threshold were analysed to verify whether all the anomalous events were retained. This is to ensure that, the application of threshold on the clustered events have not removed the significant events needed for further scrutiny.

Subset-2. There were two unsuccessful and one successful intrusive events in Apache access log and all these events were retained with clusters generated by all the algorithms and settings. But in the case of Apache error log, there were 14 successful intrusive events and 58 unsuccessful events. The 14 successful events were retained from K-Means clusters for setting 2 and 4, and from EM clusters with setting-2. The 58 unsuccessful intrusive events were retained from EM clusters for settings 1 and 4, and from FF clusters with settings 1 and 2. None of the algorithms retained both successful and unsuccessful events together for all the settings. This was due to the cluster formation with various settings and the placement of such events in bigger clusters which were filtered by the threshold.

Subset-3. There were 72 successful and 436 unsuccessful intrusive events in Apache access log. All the successful events were retained by the threshold from EM clusters. A maximum of 209 (47.94%) unsuccessful

events were retained from K-Means cluster with setting-4. There were 484 intrusive events recorded by Apache error log, and a maximum of 102 (21.07%) such events were retained by the threshold from EM clusters with settings 1, 3 and 4. This was due to the high volume of such events in this log, which eventually joined together in bigger clusters and therefore filtered-out by the threshold.

Subset-4. There were 78 unsuccessful intrusive events in Apache error log. All these events were retained by the threshold from K-Means clusters with setting-1, and from FF clusters with settings 1 and 2. But the volume of events retained gets declined for settings 3 and 4 with K-Means and FF-clusters. This was because, the calculated threshold value decreased due to the increase in the number of clusters generated for these settings. Therefore, those unsuccessful events in a slightly bigger clusters was filtered out. But then, with EM clusters, 51 events were retained for settings 1 and 2, and 61 events for settings 3 and 4. The increase in retention was due to the nature of EM, by generating clusters depending only on the available patterns irrespective of the requested clusters.

The following are the conclusions that were drawn, analysing the filtered logs:

- There was no uniform increase or decrease in reduction of events with respect to algorithms and its settings. This was due to the varying event patterns in different logs and the influence of initialisation parameters, i.e, K and seed, on cluster formation.
- Event reduction varies with the calculated threshold and cluster size of the log being filtered. Threshold varied with the number of clusters generated whereas cluster size was influenced by the cluster formation which changes with the algorithms and settings, thereby influencing the reduction of events.
- Precise cluster formation did not always retain the maximum number of intrusive events when the volume of such events in the log was high. This was due to formation of such events in bigger clusters, which was eventually filtered out by the threshold.
- Application of calculated threshold managed to retain most or all the intrusive events in various logs, but then in some cases it failed, when such events were high or wrongly placed in the bigger clusters.

6.3. Aggregated Events. The filtered-in events were aggregated to reduce the duplicates, i.e., an event that contains similar values for all the features like the previous event. No additional feature was introduced in aggregated log to specify the number of events that were combined together to represent an aggregated event. Subsequently, all the intrusive events filtered-in were unique, accumulation did not eliminate any of these events. Fewer number of filtered-in events were reduced in logs, but then further process was not affected by this reduction. Apache access and error log were trimmed to an average of 12.75%, whereas an average of 2% in Linux syslog. Nearly 75% of the filtered in events were reduced in SSL-Error log was because of analogous event patterns recorded in this log with the same timestamp. Aggregation reduced an average of 20% filtered-in events in Snort IDS log. A similar percentage of reduction was accomplished by accumulation with the other three subsets as well. All the events that were retained by the respective logs after it has been filtered and aggregated were abnormal of some kind.

6.4. Transferred Events. The events transferred to GFL from various logs were filtered and aggregated in the previous phases. The unavailability of a feature in a log was replaced with a hyphen when transferred to the GFL. No additional features were introduced to represent the log type it belongs to, as it may mislead the clustering method. Although all the logs with the specified features in GFL were used for investigation to detect intrusions, only certain logs were transported to GFL format to be used in the clustering phase. Apache and Linux events were captured directly by the victim system and every abnormal activity expressed by Apache server logs and Linux syslog should have generated events in Snort IDS log, as it has been designed to capture such actions. As we planned to group the events to evaluate the relationship between the abnormal events of Apache server logs and Linux syslog with the events of Snort IDS, these events were transferred to GFL. Since the events of Apache SSL-Error log and Linux mail log did not have IP address and port number, these events were not transferred. As stated in GF the respective features of the excluded logs were maintained separately to be used in the analysis phase. A custom written Perl script extracted the features from various logs as stated in GF and transferred it to GFL, which took less than 5 seconds for transferring 11000 events. Since we have tested three algorithms with four settings, the volume of events transferred from individual logs to GFL varies with the algorithms and their respective settings. Therefore, the respective GFL events were maintained

separately to be processed in subsequent phases and volume of GFL events for all the subsets is provided in Table 6.5. Every GFL was verified to ensure whether all the events in the aggregated logs were completely

TABLE 6.5
GFL Events

Algorithms	Settings	Subset-1	Subset-2	Subset-3	Subset-4
K-Means	Setting-1	2067	5001	2300	3258
	Setting-2	1145	3731	1773	4975
	Setting-3	1803	4566	2954	3583
	Setting-4	1966	5957	2041	3733
EM	Setting-1	1232	7808	2076	3375
	Setting-2	2664	6623	1738	4760
	Setting-3	1776	5917	2600	2881
	Setting-4	2635	7171	2021	3221
FF	Setting-1	2193	7515	1544	1971
	Setting-2	1319	5962	1222	2279
	Setting-3	946	3157	913	1969
	Setting-4	1545	3842	907	1873

transferred to GFL and the features extracted from various logs has been appropriately placed in the respective GFL features. Verification showed that the events from the respective logs were completely and appropriately placed in GFL without errors.

6.5. Clustered GFE. The GFL constitutes events from various logs that were mostly abnormal and these events were clustered to find the relationship between them using the same clustering strategy used before. Different volumes of GFL events which varies according to the algorithms and settings were clustered separately. The ideal number of clusters for GFL was manipulated using EM as stated in Experiment 4, and the resulting clusters (K) were used to group GFL as per Experiment 5 stated. As the GFL constitutes events from various logs which were clustered and filtered previously, accurate cluster formation and subsequent retention of abnormal events in those phases affected the clustering accuracy and formation in this phase.

Since the volume of events in GFL that was clustered varies with algorithms and settings, the accuracy achieved with various algorithms and settings could not be directly compared.

Subset-2. The accuracy of clustering by various algorithms and settings for subset-2 is illustrated in Figure 6.11.

The maximum accuracy 100% achieved by FF with setting-4 was due to the grouping of likely events precisely in separate clusters. Especially, the Snort IDS events with different patterns were grouped in separate clusters. K-Means with setting-1, generated clusters which contained events from various logs together. Like K-Means, FF and EM with setting-1 also generated clusters containing events from various logs together in the same cluster. This was due to the similarity between the event features contained in the GFL.

Subset-3. The accuracy of clustering by various algorithms and settings for subset-3 is illustrated in Figure 6.12. Like subset-2, the highest accuracy of 100% was achieved with FF due to the precise cluster formation with respect to the patterns. All the three algorithms generated clusters separating the events according to the patterns for this subset, i.e., Linux and Apache events joined together in the same clusters whereas Snort IDS events joined together in separate clusters. But the clusters generated by K-Means with setting-2 contained clusters with events from various logs.

Subset-4. The accuracy of clustering by various algorithms and settings for subset-4 is illustrated in Figure 6.13. Like subsets 2 and 3, the highest accuracy was achieved by FF due to the separation of events in various logs according to patterns. K-Means generated clusters that constitute events from various logs was due to its nature of giving equal importance to all the features in the log and also the feature similarity of the events in GFL.

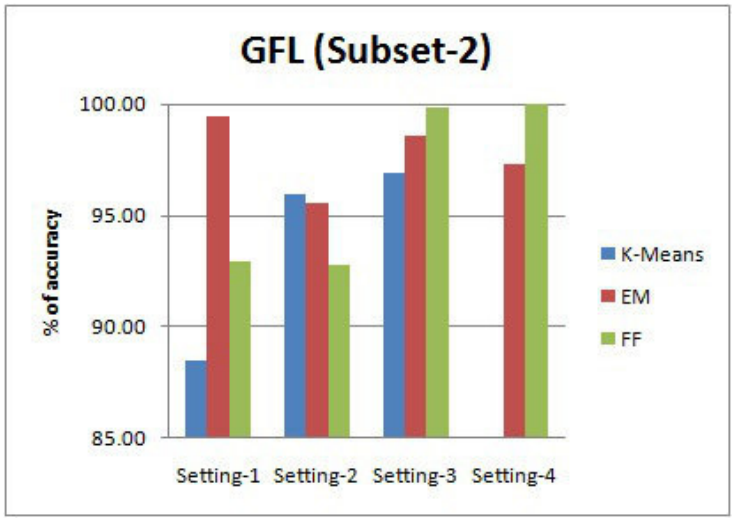


FIG. 6.11. *Clustering Accuracy with GFL (Subset-2)*

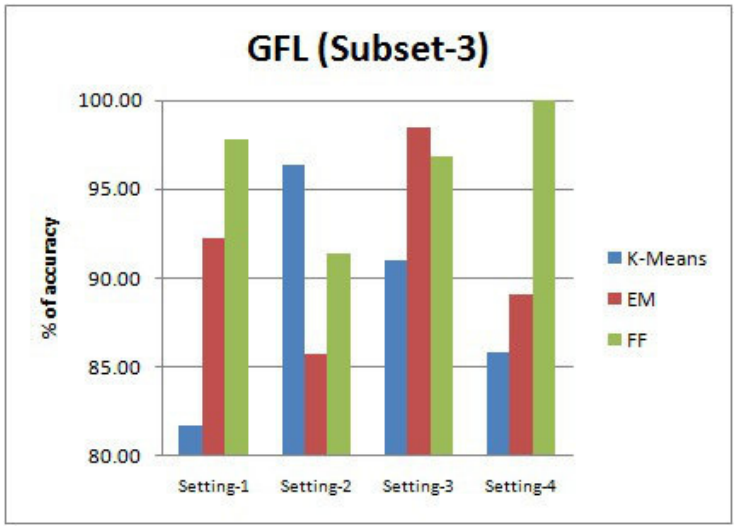


FIG. 6.12. *Clustering Accuracy with GFL (Subset-3)*

Analysing the clusters generated by various algorithms with varying volume of GFL events provides the following conclusions:

1. The variations in clustering accuracy and cluster formation with different algorithms and settings was not only due to the parameter initialisation for that setting, but also due to the variations in volume of events clustered with that algorithm and setting.

2. FF achieved the maximum accuracy with all the four subsets exposed its capacity in clustering heterogeneous log events of varying patterns, despite the absence of values for some of the features.

3. K-Means produced clusters that constitutes events from various logs together for most of the subsets. This was not only due to the similarity in the event features, but also shows its capacity in treating features in an event equally.

6.6. Anomaly Detection. In this phase, the clustered GFL events were examined to detect anomalies by finding the relationship between the features recorded by the events form various logs. The features source

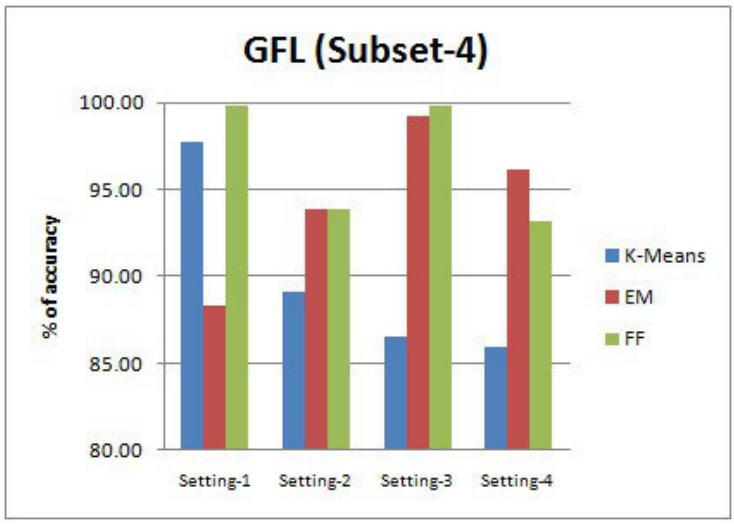


FIG. 6.13. *Clustering Accuracy with GFL (Subset-4)*

IP address (f_{sip}) and destination IP address (f_{dip}) together with cluster number (f_k) were examined during IP Address analysis and the features source port number (f_{sp}) and destination port number (f_{dp}) were examined during port number analysis. Previously the results of subset-1 for anomaly detection was presented in Hajamydeen et.al. [7], and therefore the results of the other three subsets, i.e., subset-2, subset-3, subset-4, were only discussed. Moreover, a portion of the results on the other three subsets, i.e., subset-2, subset-3, subset-4, were mentioned in Hajamydeen et.al. [52] for comparison purposes, but not detailed.

6.6.1. IP Address Analysis. Every clustered GFL with the respective algorithms and settings were analysed separately and the resulting anomalous events were evaluated as per Experiment 6. Most or all clustered events of GFL were basically anomalous and IP analysis managed to locate the anomalies by identifying the relation between events with respect to IP Address followed by cluster number. IP analysis indicated its capacity in recognising most of the anomalous events in the logs and the volume of anomalous events recognised was affected by the creation of clusters in individual logs and GFL.

Subset-2. The volume of events in this subset was larger compared to other subsets. The anomalous events detected by IP analysis using the clustered events generated by various algorithms with different settings are illustrated in Figure 6.14. The best performance for this subset was achieved with K-Means clusters by discovering most of the anomalous events through IP analysis. This was due to the capacity of K-Means in generating better clusters even with bigger datasets. There were 16 events related to successful intrusion and 59 events related to unsuccessful intrusion exploiting the AWStats vulnerability recorded in Apache access and error log which originated from two IP addresses. The events pertaining to this activity was not available in Snort IDS log, as it was missed by Snort. But IP analysis managed to detect most of these anomalous events. The volume of events detected by IP analysis from these IP addresses are illustrated in Figure 6.15. IP analysis detected all the unsuccessful intrusive events with EM clusters for settings 1 and 4, and with FF clusters for settings 1 and 2, but then, most of the successful intrusive events were not detected with these settings. All the successful events were detected with K-Means clusters for settings 2 and 4, and with EM clusters for setting-2. This was due to the unavailability of these events for analysis, since it was filtered-out in the previous phase. Including the IP addresses for which the results are plotted (Figure 6.15), there were seven IP addresses from where the anomalous activity have originated and there were 1252 events related to these IP addresses in various logs.

Over 900 anomalous events from six IP addresses were detected with the clusters generated by K-Means for settings 1 and 2. With K-Means clusters for settings 3 and 4, the events related to all the seven IP addresses were detected, but then the volume of such events were less compared to the other two settings. The volume

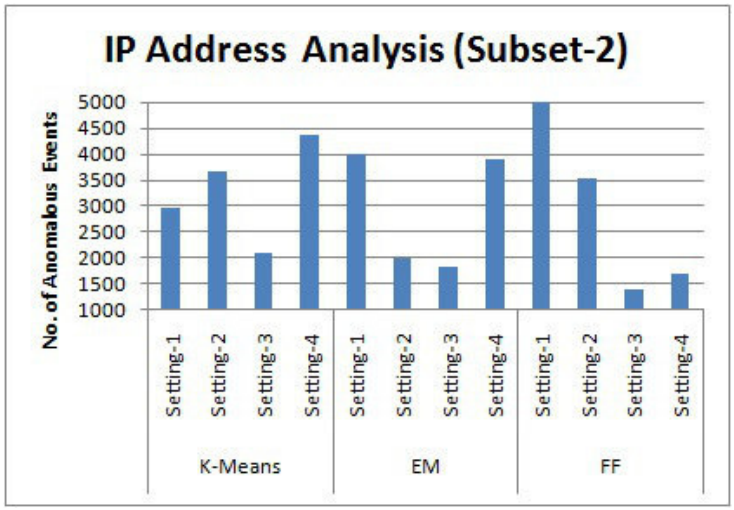


FIG. 6.14. *Anomaly Detection by IP Address Analysis (Subset-2)*

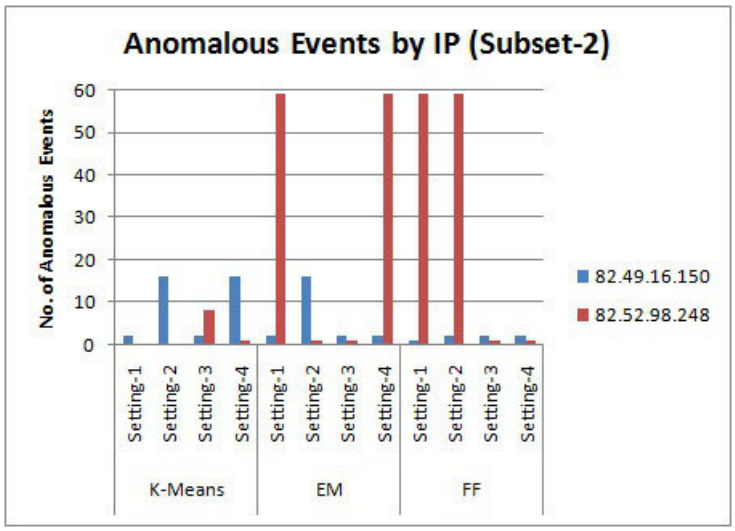


FIG. 6.15. *Anomalous Events by IP (Subset-2)*

of events or the number of IP addresses detected with the clusters of a particular setting was influenced by the cluster formation and the subsequent application filtering threshold based on cluster formation. Although IP analysis detected the most number of anomalous events with FF clusters for this subset, it failed to detect most of the significant anomalies pertaining to these seven IP addresses. This was because of the unavailability of these events due to cluster formation and subsequent application of filtering threshold in the previous phases.

Subset-3. Like subset-2, IP analysis detected majority of the anomalous events with K-Means clusters for this subset too. The anomalous events detected by IP analysis using the clustered events generated by various algorithms with different settings are illustrated in Figure 6.16. The anomalous events have originated from ten IP addresses and there were 1901 such events in this subset. A maximum of 423 events from nine IP addresses were detected by IP analysis with K-Means clusters. This was due to the volume of such events were large and most of these events were filtered out by the threshold, making it unavailable for analysis. A maximum of 126 events were detected with EM clusters and 71 events with FF clusters. This reveals the performance of the EM

and FF clustering in handling bigger size logs.

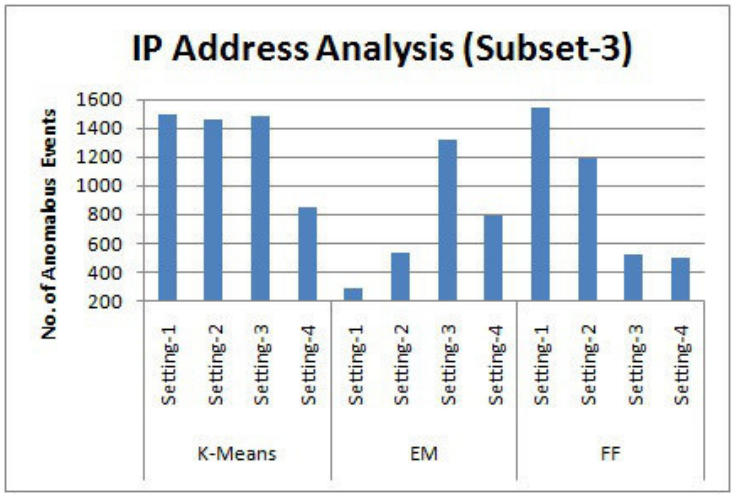


FIG. 6.16. *Anomaly Detection by IP Address Analysis (Subset-3)*

There were 30 events related to xinetd crash which originated from three IP addresses that were recorded by Linux security and Snort IDS log. A total of 20 such events were detected by IP analysis with K-Means and FF clusters as illustrated in Figure 6.17. The figure also shows that, the event from the IP address 195.22.66.28

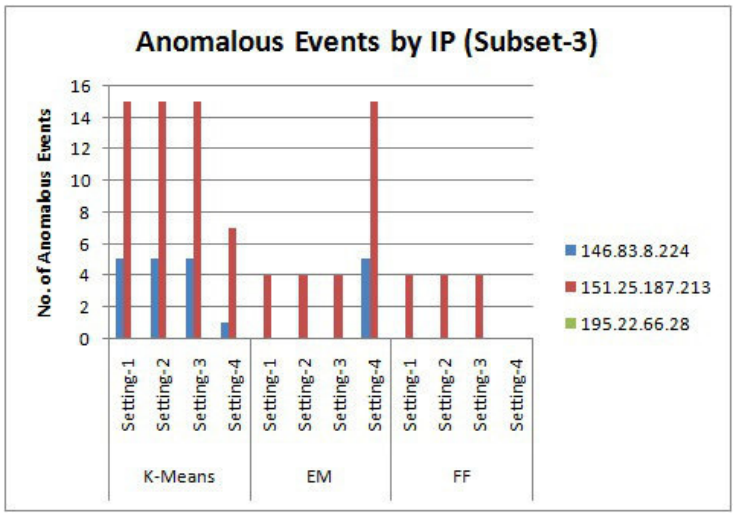


FIG. 6.17. *Anomalous Events by IP (Subset-3)*

was not detected with the clusters of any algorithms. There was only one event from the IP address 195.22.66.28 that was recorded by Linux security log and this event was filtered out for all the settings with K-Means and FF clusters, making it unavailable for analysis. Even though this event was retained after filtering with EM clusters, the failure of IP analysis to detect it was due to the cluster formation with GFL.

Subset-4. The anomalous events identified by IP analysis for various algorithms with different settings are illustrated in Figure 6.18. IP analysis detected most of the anomalous events with K-Means and EM clusters for this subset as illustrated in Figure 6.19. There were 144 events originated from three IP addresses and especially the events related to RPC attack was recorded only in Snort IDS log.

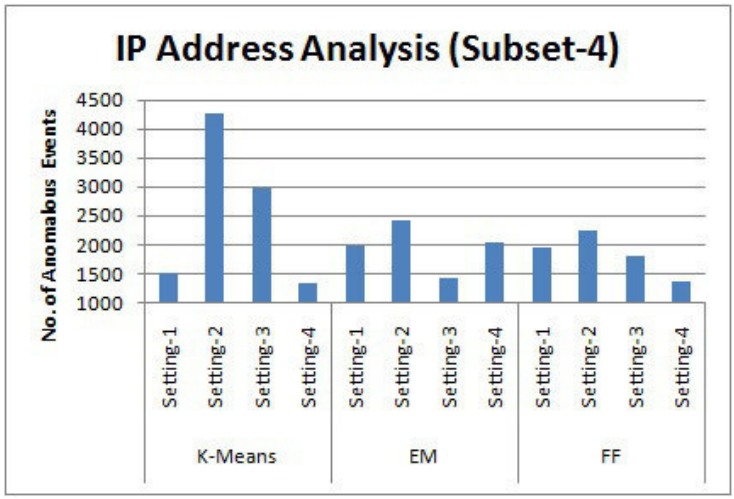


FIG. 6.18. *Anomaly Detection by IP Address Analysis (Subset-4)*

IP analysis detected majority of the anomalous events from the IP addresses 220.110.29.27 and 59.120.2.133 with the clusters generated by various algorithms and the respective settings. The RPC attack originated from 62.111.213.88 were not detected by IP analysis, which was due to the cluster formation with GFL with some of the settings.

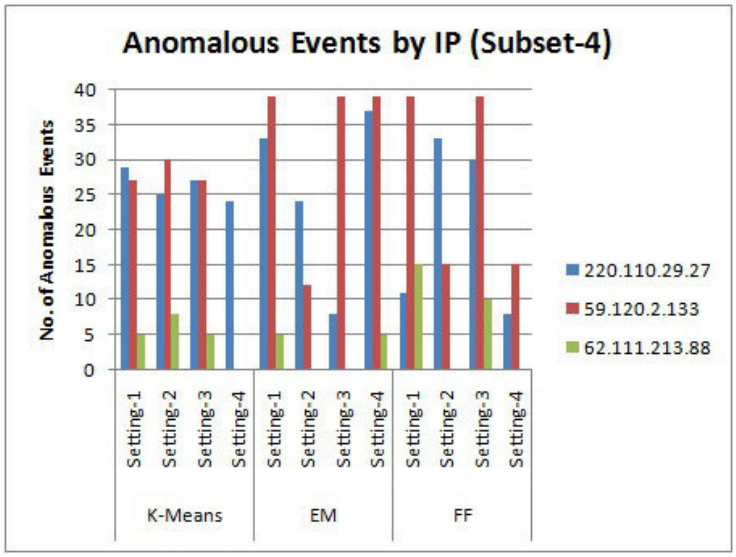


FIG. 6.19. *Anomalous Events by IP (Subset-4)*

The following are the conclusions drawn based on the detected anomalies by IP analysis:

- The detection performance of was dependent on the performance of the previous phases. As IP analysis bases its decisions from the log events received as input which was manipulated by the previous phases, the performance of these phases affected the detection process.
- The detection performance declined with those log subsets having a larger volume of anomalous events. Although, the clustering strategy implemented generated better clusters, the high volume of anomalous events in the logs were grouped in bigger clusters. The application of threshold filtered-out these anomalous events and hence not available for analysis.

- The maximum number of anomalous events from various IP addresses were detected with EM clusters for subsets 1 and 4, whereas with K-Means clusters for subsets 2 and 3. The volume of events in various logs with subsets 2 and 3 were high, and K-Means forming better clusters than EM especially with these subset shows its ability in handling larger datasets.

6.6.2. Port Number Analysis. Every clustered GFL with the respective algorithms and settings were analysed separately and the resulting anomalous events were evaluated as per Experiment 7. A simple port analysis was done by comparing the port number of the events with the IANA listing of unassigned and dynamic port numbers. Only the Snort IDS events in GFE were scrutinized, since it contains both source and destination ports.

Subset-2. The volume of anomalous events detected by port analysis with various algorithms and its settings are illustrated in Figure 6.20. A maximum of 595 events were detected by port analysis for this subset

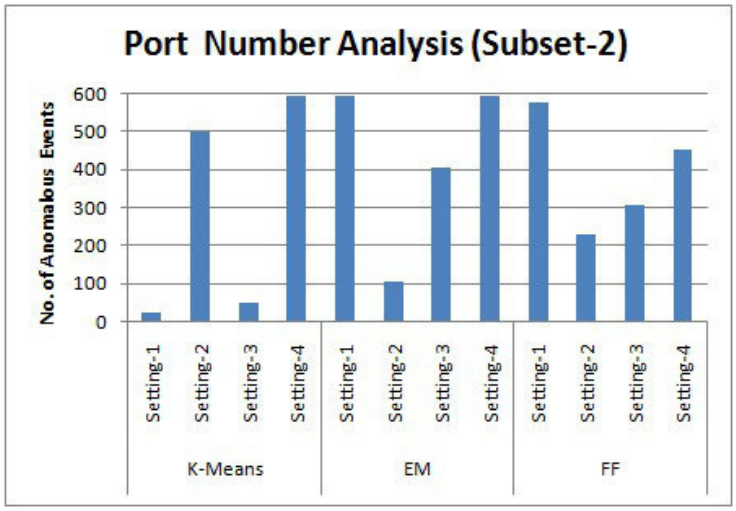


FIG. 6.20. *Anomaly Detection by Port Number Analysis (Subset-2)*

with EM clusters and the highest number 105 of anomalous events detected with setting-4 shows the impact of K on cluster formation. The anomalous events pertaining to AWStats vulnerability were not detected by port analysis which was available in this subset. The events related to this activity were recorded by Apache access and error log and was missed by Snort IDS. Since port analysis considers only those events with port numbers, these events were not detected. There were five IP addresses recorded by Snort IDS log from where the intrusions originated, out of which the events related to one IP address was detected with most of the clusters generated with various algorithms. This was due to cluster formation and the subsequent retention of events by the threshold for this log.

Subset-3. The anomalous events detected by port analysis for this subset with various algorithms and its settings are illustrated in Figure 6.21. A maximum of 111 events were detected with FF clusters (setting-2), which includes seven anomalous events from two IP addresses as stated in SOTM#34 analysis results. Similarly, a maximum of 78 events were detected with EM clusters (setting-4) which comprises 10 anomalous events from two IP addresses. This was due to cluster formation and the subsequent retention of events by the threshold for this log.

Subset-4. The anomalous events detected by port analysis for subset-4 with various algorithms and its settings are illustrated in Figure 6.22. There were 90 anomalous events originated from three IP address that were recorded by Snort IDS log. A maximum of 20 anomalous events from two IP addresses were detected by port analysis with K-Means clusters (setting-2). Although the events from these two IP addresses were detected with the clusters generated with EM and FF with certain settings, the volume of events detected were

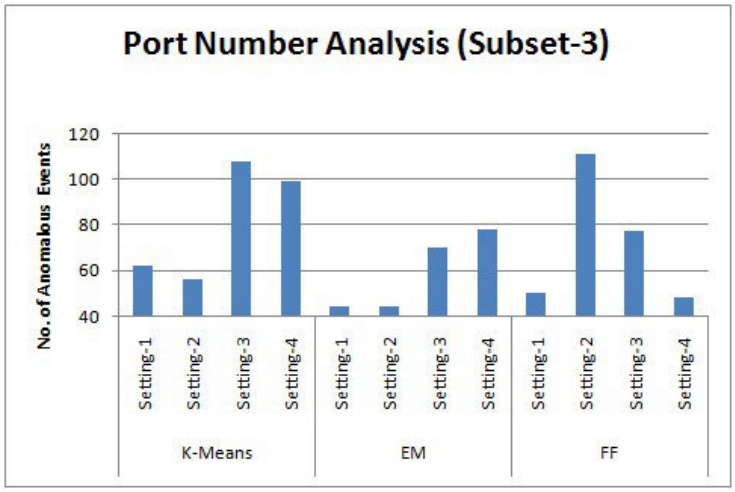


FIG. 6.21. *Anomaly Detection by Port Number Analysis (Subset-3)*

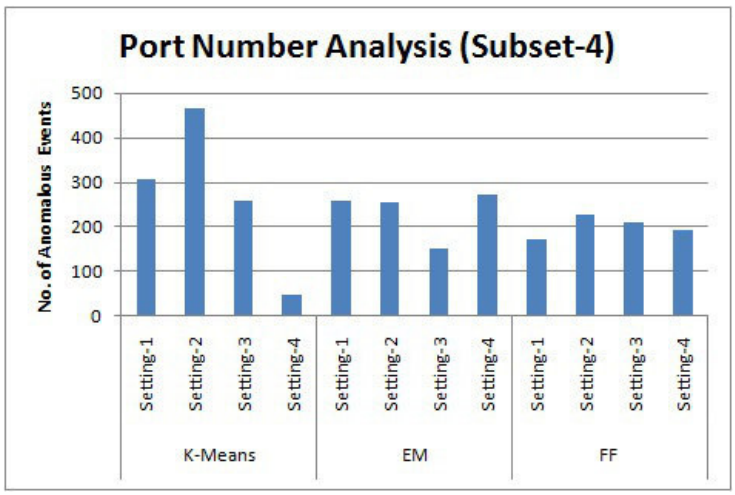


FIG. 6.22. *Anomaly Detection by Port Number Analysis (Subset-4)*

less compared to K-Means. This was due to the cluster formation of these algorithms with Snort IDS log. Moreover, the volume of events in Snort IDS log for this subset was large. The better performance of K-Means with this subset shows its ability in clustering voluminous log events, thereby supporting the detection process.

In conclusion,

- Approximately 10% to 15% of the anomalous events in the logs were detected by port analysis was due to the unavailability of port numbers in most of the log events.
- A range of 1 to 187 anomalous events that were missed during IP analysis were discovered by port analysis with various subsets. Most of the events identified by port analysis were already identified during IP analysis reduces the need for port analysis.
- The detection of anomalous events by port analysis was influenced by the cluster formation in Snort IDS log and the succeeding events retained by the filtering threshold.

6.7. Consolidated Anomalous Events. The anomalous events detected by IPA and PA with the respective algorithms and settings were consolidated separately and the resulting anomalous events were evaluated as per Experiment 8. In the process of consolidation, the distinct anomalous events identified by both analysis

methods were initially combined together. To get a better picture of the successful and unsuccessful intrusion, those IPTables firewall log events whose IP address matches the IP address of the combined anomalous events were extracted and merged with the combined analysis results. The volume of events that gets added during correlation depends on the available IP addresses of the events detected during analysis and the corresponding match with IPTables firewall log. The anomalous events were concentrated using a threshold on the occurrence of events pertaining to an IP address after correlating with IPTables firewall log. This was performed to trim down the insignificant events that can be disregarded. Since varying volume of events were detected with the clusters generated with various algorithms and settings in the previous phase, combining and correlating these events also yielded varying volume of events. Concentrating these events with a specific threshold reduced a small number of events according to the occurrence of events related to an IP address. Although concentration trimmed down most of the insignificant events, there are chances of significant anomalous events being reduced, as the threshold is based on IP address. Therefore, after concentrating the events with several thresholds, the consolidated events with various threshold was verified to ensure whether the application of threshold removed or reduced the significant anomalous events, i.e., as stated in SOTM#34 analysis results, from various IP addresses. Therefore the significant anomalous events detected by UHAD which was also mentioned in SOTM/#34 analysis results were discussed in this section. But then, UHAD has detected more number of anomalous events than those mentioned in SOTM#34 analysis results.

Subset-2. Even though the volume of events in this subset was high, concentrating the events with the threshold, i.e., A_t= 9, reduced an average of 3% significant anomalous events. The details of the significant anomalous events retained from the clusters generated with various settings and algorithms is provided in Figure 6.23. A total of 1339 significant anomalous events which was launched from several IP addresses were retained

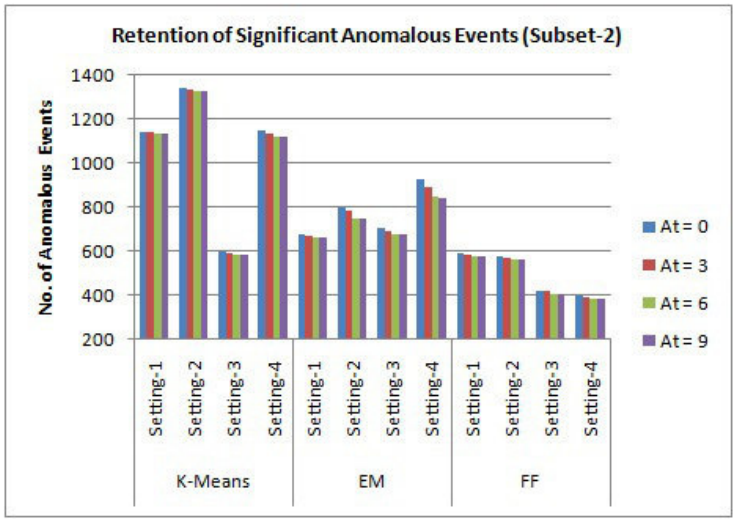


Fig. 6.23. *Retention of Significant Anomalous Events (Subset-2)*

after combining and correlating the events detected with K-Means clusters. Applying the threshold, i.e., A_t= 9, removed 31 of these events from three IP addresses.

Subset-3. The effect of threshold in retaining significant anomalous events is presented in Figure 6.24. A minimum average of 0.8% significant anomalous events were reduced with K-Means clusters and a maximum average of 4.3% events were removed with EM clusters during concentration with the maximum threshold, i.e., A_t= 9. This was due to the poor cluster formation by EM for this subset with most settings. A maximum of 2205 events were yielded after combining and correlating the detected events with K-Means clusters for setting-3. Applying the threshold, i.e., A_t= 9, removed 16 anomalous events from two IP addresses.

Subset-4. The details of the significant anomalous events retained from the clusters generated with various settings and algorithms is provided in Figure 6.25. A minimum average of 0.6% significant anomalous events

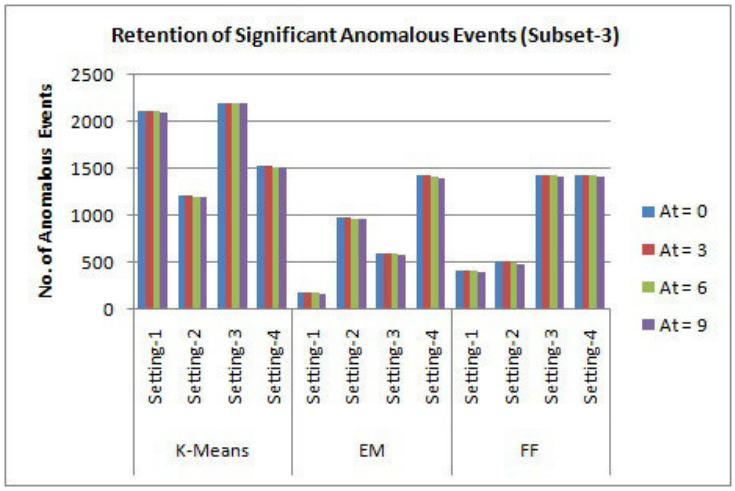


FIG. 6.24. *Retention of Significant Anomalous Events (Subset-3)*

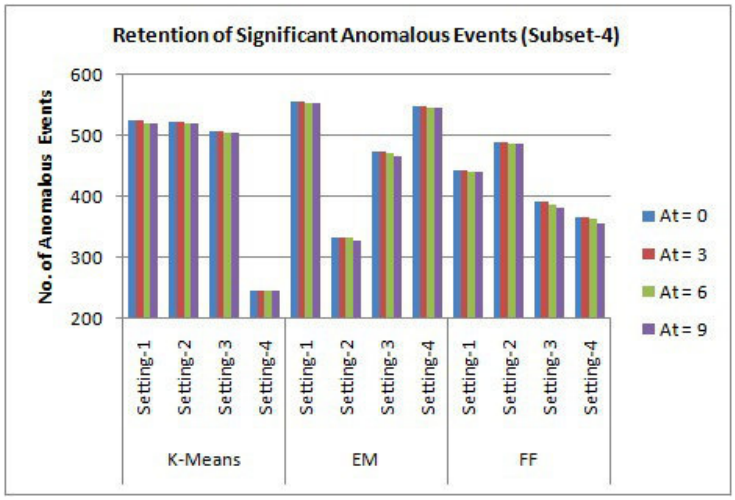


FIG. 6.25. *Retention of Significant Anomalous Events (Subset-4)*

were reduced with K-Means clusters and a maximum average of 1.8% events were removed with FF clusters during concentration with the maximum threshold, i.e., $A_t = 9$. A maximum of 555 significant anomalous events were captured after combining and correlating the detected events with EM clusters for setting-1. Application of threshold, $A_t = 9$ reduced three events from an IP address.

The following conclusions are drawn from consolidating the detected events:

- As the reduction was based on the IP addresses of those events that were detected and correlated previously, the output of this phase highly depends on the log events received from the previous phase. Therefore, the volume of events yielded after consolidation varies with the detected events of various algorithms and settings.
- Concentrating the detected events with the threshold not only reduced the insignificant events but also reduced a minimum number of significant anomalous events. Though the usage of threshold is not viable at this end, it will assist in focusing the most critical events that need to be immediately addressed.

UHAD detected a wide range of anomalies which includes Nimda scans, CodeRed worms, SSH Brute Force

scans, CONNECT scans, IRC Traces, RPC attack, Xinetd crash and the AWStats vulnerability exploit. The low percentage of anomalies detected with subset-3 is due to the high volume of events and anomalies in this subset.

7. Conclusion and Future Works. The majority of the intrusion detection mechanisms available were knowledge dependent which makes use of the characteristics of anomalies or the model of traffic behaviour to detect anomalies which restricts the mechanisms to detect only known anomalies. Moreover, the existing detection methods considers a single type of log for analysis, which confines the method to detect anomalies presented only in those logs and the anomalies in the other logs were left behind. To overcome these limitations, this thesis has presented a new framework UHAD to detect a variety of anomalies by scrutinizing logs from heterogeneous sources, without using the characteristics of anomalies that hold the specification of the actions to match with events or the usual method of training and testing commonly used in anomaly detectors. Although, the three clustering algorithms tested in the framework took less time to predict and generate clusters, the accuracy of the clusters generated by an algorithm were not consistent across different logs and subsets. This was due to the capacity of the clustering algorithms in handling the event patterns and features in a particular log. Additionally, the clustering parameters used to group the events also influenced the accuracy. Subsequently, applying the filtering threshold which was automatically calculated based on these generated clusters, managed to retain majority of the abnormal events and removed the normal or insignificant events with most of the logs and subsets. With some of the logs, the filtering threshold failed to retain the abnormal events was due to the existence of more number of such abnormal events of similar patterns that were grouped in larger clusters. The usage of GFL induced the process of classifying heterogeneous log events in a single structure and enabled faster analysis by the detection algorithms. Introducing more features in GFL, which did not existed in many of the logs, affected accurate cluster formation thereby reducing the volume of the events detected during analysis. This shows the criticality in selecting the appropriate features for GFL based on the logs considered. Further analysis of the GFL based on the IP addresses and port numbers detected the events related to a wide range of anomalies and most importantly, more anomalies were detected during IP address analysis than port number analysis. Manipulating IP address (source IP and destination IP) together with the cluster number supported the analysis process in detecting majority of the anomalous events. Moreover, almost all the anomalies detected by port analysis were also detected by IP address analysis thereby reducing the need for port analysis. Consolidating the detected anomalies assisted in focusing the most significant anomalous events, but then failed to retain a few of the anomalous events, when the occurrence of such events did not satisfy the threshold applied.

The proficiency and precision of UHAD in recognizing intrusions were scrutinized using three clustering algorithms with four parametric settings and the results achieved were compared with the ground truth available at Honeynet.org for this dataset, i.e. SOTM#34. The detected anomalies were analogous with the output of other methods, therefore demonstrating the accuracy of UHAD in detecting anomalies. Among the three algorithms used by the framework, EM and K-Means generated better clusters supporting the investigation process to identify the majority of the anomalies in all the four subsets. The coverage of anomalies with FF clusters were marginally lower than EM and K-Means, but is also appropriate for this framework due to its faster clustering even with larger datasets.

All the parameters used in UHAD were manipulated based on the tested dataset only. Therefore, the accuracy of clusters and subsequent retention of anomalies by the threshold were influenced by these parameters. None of the algorithm showed a consistent accuracy with a specific parameter setting; and also the performance of a particular algorithm was not steady with different subsets. Because of this, some of the anomalous events were filtered-out by the threshold, making it unavailable for further investigation. Since this is the first step towards building an unsupervised anomaly detector using heterogeneous logs that calculates all the required parameters based on tested data itself; the mechanism of manipulating the clustering parameters and filtering threshold could be refined to improve the precision of anomaly detection. A recommender method could be designed to select the most accurate clusters for a particular log, among those produced by various algorithms with different parametric settings. This method need to evaluate various output parameters based on the clusters generated which requires criteria and threshold to be framed in order to evaluate such parameters to choose the accurate clusters. Subsequently, these selected clusters can be used for further investigation to detect almost all the anomalies. UHAD can also be tested with various datasets collected from heterogeneous sources

to substantiate its capacity in discovering the anomalous events pertaining to a wide range of intrusions.

REFERENCES

[1] S. More, M. Matthews, A. Joshi and T. Finin, *A Knowledge-Based Approach To Intrusion Detection Modeling*. In 2012 IEEE Symposium on Security and Privacy Workshops (SPW),2012, pp. 75–81. IEEE.

[2] C. Abad,J. Taylor, C. Sengul, W. Yurcik, Y. Zhou and K. Rowe, *Log correlation for intrusion detection: A proof of concept*, In Proceedings of 19th Annual Computer Security Applications Conference., 2003, pp. 255–264.

[3] Z. Li, J. Taylor, E. Partridge,Y. Zhou, W. Yurcik, C. Abad, J. Barlow and J. Rosendale, *UCLog: A unified, correlated logging architecture for intrusion detection*, In the 12th International Conference on Telecommunication Systems-Modeling and Analysis (ICTSM)., 2004.

[4] W. Yurcik, C. Abad, R. Hasan, M. Saleem and S. Sridharan, *UCLog+: A Security Data Management System for Correlating Alerts, Incidents, and Raw Data From Remote Logs*, Arxiv preprint cs/0607111., 2006.

[5] R. Kumari and K. Sharma, *Cross-Layer Based Intrusion Detection and Prevention for Network*, In Handbook of Research on Network Forensics and Analysis Techniques, 2018, pp. 38-56. IGI Global.

[6] D. Denning, *An intrusion-detection model*, IEEE Transactions on software engineering., 2(1987), pp. 222-232.

[7] A. Hajamydeen, N. Udzir, R. Mahmod and A. Ghani, *An unsupervised heterogeneous log-based framework for anomaly detection*, Turkish Journal of Electrical Engineering and Computer Sciences., 24(2016), pp. 1117-1134.

[8] S. Peisert and M. Bishop, *How to design computer security experiments*, In Fifth World Conference on Information Security Education., 2007 , pp. 141–148. Springer.

[9] E. Barse and E. Jonsson, *Extracting attack manifestations to determine log data requirements for intrusion detection*, In 20th Annual Computer Security Applications Conference., 2004, pp. 158–167. IEEE.

[10] X. Wang, A. Abraham and K. Smith, *Intelligent web traffic mining and analysis*, Journal of Network and Computer Applications., 28.2(2005), pp. 147–165.

[11] A. Chuvakin, *Scan of the Month 34*, http://www.honeynet.org/scans/scan34/, 2005.

[12] Andrew, *Scan of the month 34-Solution*, http://www.honeynet.org/scans/scan34/sols/3/sotm/, 2005.

[13] M. Richard and M. Ligh, *Project Honeynet Scan of the Month 34*, http://project.honeynet.org/scans/scan34/sols/1/index.html, 2005.

[14] C. Kronberg, *Analysis of the log files given in SOTM34*, http://project.honeynet.org/scans/scan34/sols/2/proc.pdf, 2005.

[15] A. Chuvakin, *Scan of the Month Challenge 34- Official Solution*, http://project.honeynet.org/scans/scan34/sols/sotm34-anton.html, 2005.

[16] S. Panichprecha, *Abstracting and Correlating Heterogeneous Events to Detect Complex Scenarios*, PhD thesis, Queensland University of Technology, Brisbane, Australia., 2009.

[17] J. Herrerias and R. Gomez, *Log analysis towards an automated forensic diagnosis system*, In International Conference on Availability, Reliability, and Security, ARES'10., 2010, pp. 659–664.

[18] J. Herrerias and R. Gomez, *A log correlation model to support the evidence search process in a forensic investigation*, In Second International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE., 2007, pp. 31–42.

[19] K. Kent and M. Souppaya, *Guide to computer security log management*, NIST Special Publication, 800–92.,2006.

[20] G. Ferrar, *Sawmill (Version 8.1.10)*, http://www.sawmill.met, 2011.

[21] F. Amiri, M. Yousefi, C. Lucas, A. Shakery and N. Yazdani, *Mutual information-based feature selection for intrusion detection systems*, Journal of Network and Computer Applications., 34.4(2011), pp. 1184–1199.

[22] T. Chou, K. Yen and J. Luo, *Network intrusion detection design using feature selection of soft computing paradigms*, International Journal of Computational Intelligence., 4.3(2008), pp. 196–208.

[23] C. Sinclair, L. Pierce and S. Matzner, *An application of machine learning to network intrusion detection*, In Proceedings. 15th Annual Computer Security Applications Conference., 1999, pp. 371–377. IEEE.

[24] D. Barbara, N. Wu and S. Jajodia, *Detecting novel network intrusions using bayes estimators*, In Proceedings of the First SIAM International Conference on Data Mining., 2001.

[25] Y. Li, N. Wu, X. Wang and S. Jajodia, *Enhancing profiles for anomaly detection using time granularities*, Journal of Computer Security., 10.1-2(2002). pp.137– 157

[26] S. Staniford, J. Hoagland and J. McAlerney, *Practical automated detection of stealthy portscans*, Journal of Computer Security., 10.1-2(2002), pp.105–136 .

[27] H. Tribak, I. Rojas and O. Valenzuela, *Comparison of Soft-Computing Techniques for classification of Intrusion-Detection*, In Proceedings of the 2010 International Conference on Mathematical Models for Engineering Science., 2010, pp.284–288. World Scientific and Engineering Academy and Society (WSEAS).

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, *The Weka data mining software: An update*, ACM SIGKDD Explorations Newsletter., 2009, pp.10–18.

[29] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, *Anomaly-based network intrusion detection: Techniques, systems and challenges*, Computers & Security., 28.1-2(2009), pp. 18–28.

[30] J. Erman, M. Arlitt and A. Mahanti, *Traffic classification using clustering algorithms*, In Proceedings of the 2006 SIGCOMM workshop on Mining network data., 2006, pp.281–286.ACM.

[31] M. Aneja, T. Bhatia, G. Sharma and G. Shrivastava, *Artificial Intelligence Based Intrusion Detection System to Detect Flooding Attack in VANETs*, In Handbook of Research on Network Forensics and Analysis Techniques., 2018, pp. 87-100. IGI Global.

[32] I. Syarif, A. Prugel-Bennett and G. Wills, *Unsupervised clustering approach for network anomaly detection*, Networked

Digital Technologies, 2012, pp. 135–145.
[33] J. Song, H. Takakura, Y. Okabe and K. Nakao, *Toward a more practical unsupervised anomaly detection system*, Information Sciences., 231(2011), pp. 4-14.
[34] G. Wang, J. Hao, J. Ma and L. Huang, *A new approach to intrusion detection using artificial neural networks and fuzzy clustering*, Expert Systems with Applications., 37.9(2010), pp. 6225–6232.
[35] G. Munz, S. Li and G. Carle, *Traffic anomaly detection using K-Means clustering*, GI/ITG Workshop MMBnet., 2007.
[36] Y. Liu, W. Li and Y. Li, *Network traffic classification using kmeans clustering*, In Second International Multi-Symposiums on Computer and Computational Sciences., 2007, pp.360–365. IEEE.
[37] R. Smith, N. Japkowicz, M. Dondo and P. Mason, *Using unsupervised learning for network alert correlation*, Advances in Artifcial Intelligence., 2008, pp.308–319.
[38] U. Zurutuza, R. Uribeetxeberria, E. Azketa, G. Gil, J. Lizarraga and M. Fernndez, *Combined data mining approach for intrusion detection*, In International Conference on Security and Criptography., 2008.
[39] M. Panda and M. Patra, *A novel classification via clustering method for anomaly based network intrusion detection system*, International Journal of Recent Trends in Engineering., 2(2009), pp. 1–6.
[40] M. Siraj, M. Maarof and S. Hashim, *Intelligent alert clustering model for network intrusion analysis*, Int. J. Advance. Soft Comput. Appl., 1(2009), pp. 33–48.
[41] U. Zurutuza, R. Basagoiti and A. Aztiria, *Behavior analysis of domain servers through windows security event log mining*, J. Inform. Assurance Security., 5.4(2010), pp.418–425.
[42] G. Tjhai, S. Furnell, M. Papadaki and N. Clarke, *A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm*, Computers & Security., 29.6(2010), pp.712–723.
[43] O. Siriporn and S. Benjawan, *Anomaly detection and characterization to classify traffic anomalies case study: TOT public company limited network*, In World Academy of Science, Engineering and Technology., 2008, pp.407–415.
[44] M. Panda and M. Patra, *A hybrid clustering approach for network intrusion detection using cobweb and FFT*, Journal of Intelligent Systems., 18.3(2009), pp.229–246.
[45] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability., 1967.
[46] S. Dasgupta, *Performance guarantees for hierarchical clustering*, In Computational Learning Theory., 2002, pp.351–363.
[47] X. Zheng, Z. Cai and Q. Li, *An experimental comparison of three kinds of clustering algorithms*, In Proceedings of International Conference on Neural Networks and Brain., 2005, pp. 767–771. IEEE.
[48] A. Dempster, N. Laird and D. Rubin, *Maximum likelihood from incoming data via the EM algorithm*, J. Royal Stat. Soc., 1977, pp.1–38.
[49] M. Tavallaee, N. Stakhanova and A. Ghorbani, *Toward credible evaluation of anomaly-based intrusion-detection methods*, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews., 40.5(2010), pp. 516–524.
[50] X. Yu, L. Tang and J. Han, *Filtering and refinement: A two-stage approach for efficient and effective anomaly detection*, In 2009 Ninth IEEE International Conference on Data Mining., 2009, pp. 617–626.
[51] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos and K. Lee, *Internet traffic classification demystified: Myths, caveats, and the best practices*, In Proceedings of the 2008 ACM CoNEXT Conference., 2008.
[52] A. Hajamydeen and N. Udzir, *A refined filter for UHAD to improve anomaly detection*, Security and Communication Networks., 9(2016), pp. 2434-2447.

Edited by: Khaleel Ahmad
Received: Nov 19, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1488
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 161–180. http://www.scpe.org © 2019 SCPE

A SECURE STRUCTURE FOR HIDING INFORMATION IN A CRYPTOSYSTEM BASED ON MACHINE-LEARNING TECHNIQUES AND CONTENT-BASED OPTIMIZATION USING PORTFOLIO SELECTION DATA

CHANCHAL KUMAR* AND MOHAMMAD NAJMUD DOJA †

Abstract. Many systems including networks environment have higher complexity and ubiquitous connections than a normal system, hence design of a security system for hiding pertinent data a challenging task. This paper presents a secure structure that extends a protocol which is available for fast Diffe-Hellman protocol using Kummer surface. We show the extended version of the scheme by inclusion of an additional point, such that a more secure system can be constructed. The scheme has been build by employing machine-learning technique to select an appropriate class from multiple set of surfaces. A brief discussion on inclusion of multiple surfaces and making a selection of a specific surface using NSGA II algorithm is also provided. In this paper, we provide a brief overview of AES-128 (AES also known as Rijndael). In the starting, a short overview of the AES is given. This paper also has a description for altering the key generation module in AES based upon a newly designed content-based matrix which is built from portfolio selection data. The matrix is constructed using some predefined factors modifies the existing index which is computed based upon the context of the message. An optimization algorithm is employed for selecting specified entries from content matrix. These selected entries are used for altering the key generation algorithm in AES. The modified output obtained after altering the key generation scheme is provided in the paper. Lastly, a brief overview of LIM index, which is used as an index in cryptanalysis, is given. This paper has a description of the scheme to construct a more secure system that is capable of hiding the information with above-mentioned techniques.

Key words: Deffie-Hellman key exchange, Machine Learning techniques, NSGA II (Non-dominated Sorting Genetic Algorithm II), Key generation (AES) content based matrix, Optimization, LIM Index

AMS subject classifications. 68M12

1. Introduction.

The main objective of a cryptographic system may be stated as providing a capability for secure information exchange over a network. The plain text is the body of the message. Encryption is achieved by coding the input message using an algorithm along with a secret key. The coded message is called cipher-text. On the receiving side, the message is decoded using the secret key and reversing the steps of encryption algorithm. Thus, the plain text is obtained using the same secret key and decryption algorithm. The information which is encoded yields different sets of cipher. The study of information about structure of the cipher could be very beneficial for analysis. Cryptology deals with performing cryptanalysis in order to challenge the security of the encryption algorithm. Diffie-Hellman protocol is a commonly used protocol for secretly exchanging key [1]. In this protocol, the sender and receiver take a decision to select a finite set of points (Group) A, and choose a generator (v) from a subgroup. Using the private key (x_1) of the sender, a secret number $n_1 = (\alpha^{x_1} \bmod p)$ is generated. Here, p is a prime number and α is a primitive-root of p such that $\alpha < p$. The receiver generates a secret number $n_2 = (\alpha^{x_2} \bmod p)$ where x_2 is the private key of the receiver.The secret key exchange can be achieved by

$$k_1 = n_2^{x_1} \bmod p$$
$$k_2 = n_1^{x_2} \bmod p$$

Recently, a fast version of this protocol is provided that makes use of Kummer Surface [1, 23, 24]. In this paper, a new scheme is proposed for enhancing the security of the protocol by adding another point (u_5) on the surface. A brief overview of a scheme that can be incorporated for selecting one of the multiple surfaces using NSGA II is given in the paper.

1.1. A Brief Overview of AES.

During 1990's, it became clear that DES (Data Encryption Standard) will not be able to meet the required security standard [2]. In the year 1997, AES was adopted by NIST. AES was selected among other available ciphers. AES was chosen as a replacement for DES. AES now has a bigger key size and it is capable of handling the type of attacks meant for DES. In 2001, AES Rijndael was chosen

*Department of Computer Engineering, Jamia Millia Islamia, India (kumarchanchal943@gmail.com)
†Department of Computer Engineering, Jamia Millia Islamia, India (ndoja@yahoo.com)

among several other variants of ciphers. Previously DES was attacked using linear and differential cryptanalysis, these attacks were not successful in case of AES. AES uses block structure that means plain text is processed as blocks and the encryption is achieved on a block structure. AES composed of algebraic blocks which are relatively simple.

1.2. Different stages used in AES. AES is based upon block structure. The input plain text is divided in to various blocks of fixed size. The encryption takes place with each block separately. The size of the block is 128 bits. The allowed key size could be 128, 192 or 256 bits.

AES works with Galois Field (GF)(2^8), a polynomial $\left(b_7x^7 + b_6x^6 + ...b_0\right)$

is utilized for representing 8 bits (byte). Hexadecimal notation is used to represent bytes.

State: We are considering key size of 128 bits. The results are represented as state which is 128 bits in length and different operations are performed on the state. A matrix consisting of 4 rows and 4 columns is used as given below:

$$\begin{bmatrix} x(0,0) & x(0,1) & x(0,2) & x(0,3) \\ x(1,0) & x(1,1) & x(1,2) & x(1,3) \\ x(2,0) & x(2,1) & x(2,2) & x(2,3) \\ x(3,0) & x(3,1) & x(3,2) & x(3,3) \end{bmatrix}$$

The state is opterated with multiple rounds. Each round contains substitution box, Diffusion operation (Linear) and xoring with round key generated for each round.

The Diffusion operation consist of performing the shift rows operation then performing the transformation which is centred around mixing the different columns. For generating the round key for each round, the original key is used to generate a sub key for each round. Lastly, the algorithm used in Rijndael may described as below:

Firstly, the input message is placed in a state of 4×4 matrix. This state is xored with round key of 0^{th} round. Next, ten rounds are applied where the last round skips the mix column operation. The final state is the desired ciphertext.

This paper has description of a scheme that alters the key generation algorithm using a content-based matrix. The output of different keys for each round is given. The alteration in key generation is intended to enhance the hiding capability of AES and making it secure against different kinds of attacks.

1.3. Related Work. The use of theories of fuzzy extractor and secure sketch in the framework of key generation from biometrics is proposed in [4]. A description of lossy trapdoor function with their applications is given in [5, 6]. A significant use of pseudo-random number generators is presented in [7, 8]. A public key cryptosystem is more adaptive and secure against adaptive chosen ciphertext attack [9]. LIM (Lorenz Information Measure) based Cryptanalysis of AES-128 and AES-256 Block is presented in [11]. Some mathematical attacks like differential and linear cryptanalysis decrease the key search space but they not able to break the AES [12]. A description of some versions of Even-Mansour cipher scheme is given in [13]. Randomization based AES is better than first order differential electromagnetic and power analyses in term of low execution cost [14]. A combined approach of image processing and laser fault injections is given in [15] for security characterization of a hardware AES. A complete solution based on key updating framework to secure the execution of any kind of AES operation is proposed in [16]. A systematic toolbox is proposed in [17] for white-box implementation. A general study of the relationship between cryptography and machine learning is given in [18]. The primal machine algorithms are well performed in order to construct the encrypted models [19, 20, 21]. To perform successful key recovery deep learning techniques are appropriate [22]. A significant use of Kummer Surfaces is provided in [23, 24]. Symmetric and asymmetric cryptographic techniques based a new security protocol is proposed in [27] for online transaction.

2. Proposed System. One of the vastly used algorithm for secure key exchange is Diffie-Hellman protocol. Kummer surface is recently proposed protocol for a fast-version of Diffie-Hellman protocol. This section provides description of the new protocol build using an additional point (u_5) in the extended surface. The details of the protocol are given below:

Algorithm 1 Modified Algorithm for Multiplication by N on the extended surface)

Input: U = $(U_1, U_2, U_3, U_4, U_5) \in$ extended surface and N
Output: $N \times U$ (multiplication on the extended surface)
Function 1: calculate C_{jj} // (Output parameter C_{jj})
Input: integers i,j;//(These are index parameters) float U_v, U_w // (U_v and U_w are points on extended Kummer surface)
Output: value of C_{jj};
Details: The output is computed using the following equation: $C_{jj}(U_v, U_w) = (U_j(V+W) * U_j(V-W) + (U_j(V-W) * U_j(V+W))$, where $U_i(V)$ denotes the ith component of $U_v = U(V)$
Function 2: calculate C_{ij}
Input: integers i,j; float U_v, U_w
Output:value of C_{ij};
Details: The output is computed using the following equation: $(U_i(V+W)U_j(V-W) + U_i(V-W)U_j(V+W))$, for $1 \leq i \leq 4, 1 \leq j \leq 4$
Function 3: `Add_on_extended_surface`
Input: m = $(m_1, m_2, m_3, m_4$//(m is a point on extended Kummer surface))
integers i, j, float U_v, U_w
Output: Addition on extended surface
Details:The following steps are executed:

1. $temp_1$ = `calculate_`$C_{ij}(i, j, *U_v, *U_w)$ `where` $temp_1$ `and` $temp_2$ `are variables`
2. $temp_1$ = `calculate_`$C_{jj}(j, j, *U_v, *U_w)$
3. Output = $(2 * m[j] * temp_1$ - $m[i] * temp_2$;)
4. Parameters q_p is initialized in the beginning of main routine and it is updated here. Parameter t_9 is a teporary variable which is initialized to 0, in the beginning of main routine.
   $x[i] = Output1, 1 \leq i \leq 4$
   if $(x[i] \neq 0))$ then calculate q_p
   $q_p = q_p - ((\frac{1}{x[i]}) + (\frac{1}{x[i]*x[i]})^2)$;
   $t_9 = t_9 + x[i] * q_p, 1 \leq i \leq 4$
   $t_9 = abs(t_9)$
   Parameter ep_1 is intialized in the beginning of main routine.
   Now, compute value of $x[5]$ using the following equation:
   $x[5] = t_9 + ep_1$;
   Parameters x,y and z are used in the main routine
   Here, x,y, and z = $(u_1, u_2, u_3, u_4, u_5)$

2.1. New Scheme adopted for inclusion of point (u_5) in the extended surface. The description of new method based on inclusion of point (u_5) in extended surface is presented here:

Need: A variant of Diffie-Hellman protocol based on Kummer surface is described in [5]. To enhance the security aspects of the structure, a new point (u_5) is being added. A sample run with of $(u_1, u_2, u_3, u_4, u_5)$ is provided in the paper.

Significance: The additional point (u_5) could prove a quite useful index for decision-making. The computed values of $(u_1, u_2, u_3, u_4, u_5)$ are used in a machine-learning algorithm to select a specific surface, in case, the design considers multiple surfaces and then chooses one surface for multiplication.

Impact: The selected surface based on computed value of (u_5) will be utilized to perform a multiplication. The output obtained with added point $(u_1, u_2, u_3, u_4, u_5)$ are given below. The details of main routine along with various functions used are given below:

2.1.1. Algorithm for Machine-learning for a sample data set of three extended surfaces. A sample technique for classification based on Machine-learning is described below given in algorithm 3.

Algorithm 2 Main routine

Input: N,x = (0,0,0,1,u_5)
y = $(u_1, u_2, u_3, u_4, u_5)$
z = $(u_1, u_2, u_3, u_4, u_5)$
Output: N*Point(x)

1. Parameter: $m_1, i, j, ep_1, ep_2, q_p, t_9, t_{11}, t$;
2. Initialize:
   $t_9 = 0; t_{11} = 0.0$; // (Temporary variables)
   $ep_1 = 0.00567; ep_2 = 0.00598$; $q_p = 0.06908$;//(Coefficients)
3. Read the value of N if(N¡0) Then N = -N
   Initialize x($u_1 = 0, u_2 = 0, u_3 = 0, u_4 = 1, u_5 = 0.08$)
   Read the values of z $(u_1, u_2, u_3, u_4, u_5)$ and y $(u_1, u_2, u_3, u_4, u_5)$
4. Initialize $t_9 = \sum_{i=1}^{4} \frac{x[i]}{4}; t_{11} = \sum_{i=1}^{4} \frac{y[i]}{4}$;
5. Let h = H
6. **while** $h \neq 0$ **do**
7. **if** $((h\%2)) \neq 0$ //If h is odd **then**
   $m_i = y[i]$; // $1 \leq i \leq 5$
8. **for** $\big(i = 1; i \leq 4; i++ \big)$ **do**
   // Select value of j such that m[j] $\neq 0$
   Calculate x[i] = `add_on_extended_surface`(m,i,j,x,z)
9. **end for**N = N - 1
10. **else**
11. $m_i = x[i]$; //$i \leq 1 \leq 5$
12. **for** $\big(i = 1; i \leq 4; i++\big)$ **do**
    // Select value of j such that m[j] $\neq 0$
    Calculate x[i] = `add_on_extended_surface`(m,i,j,x,z)
13. **end for**
14. **end if**
15. $z[i] = 2 * z[i]$; //$1 \leq i \leq 4$
16. $h = \frac{h}{2}$
17. **end while**
18. 7. Display the values of $x(u_1, u_2, u_3, u_4, u_5)$

2.2. Using NSGA II for selecting a particular surface. The Kummer surface is described by the equation [1]:

$$y = (f_6 \times x^6) + (f_5 \times x^5) + (f_4 \times x^4) + (f_3 \times x^3) + (f_2 \times x^2) + (f_1 \times x^1) + f_0 \tag{2.1}$$

The next section describes different sample surfaces. An application of NSGA II for formulating multi-objective functions using different combinations of these surfaces is given next. NSGA II is an example of evolutionary algorithm designed for multi-objective optimization [28]. NSGA II could also be employed to make the decision about a particular surface.The different multi-objective equations with corresponding outputs are described below.

2.2.1. Multi-Objective Program 1. The multi-objective program 1 contains objectives Z_1 and Z_2 for two selected surfaces, which uses a parameter ep_1. The output obtained by running NSGA II for these objectives is shown in Fig. 2.1.

The initial values of parameters and equations used for objectives are given below:

- Coefficients (f_i) used in surface 1 are given below:

$$f_0 = 100, f_1 = 5, f_2 = 0.8, f_3 = -1.87063, f_4 = 0.06, f_5 = 0.0, f_6 = 0.0, \text{ and } ep_1 = 6.089$$

Algorithm 3 Algorithm for Machine-learning for a sample data set of three extended surfaces

Step 1: Key = Data - $(u_1, u_2, u_3, u_4, u_5)$
Data contains 42 entries - surface 1, surface 2 and surface 3 contains 14 points
Key = Target (surface 1 , surface 2 , surface 3)
// The algorithm given here has been tested for a sample data points of three extended surfaces.
Step 2: The in-built function `train_test_split()` , which is available in Python `sklearn.model_selection` is used for training based on this data set.
The splitting is achived by the function, is given below:
`X_train shape` = 31, `Y_train shape` = 31,`X_test shape` = 11, `Y_test shape` = 11
The parameters `X_train`, `Y_train`, `X_test` and `Y_test` are used for splitting training and testing data points.
Step 3: KneighborsClassifier() function is used for classification
This function is available in sklearn.neighbors (Python)
Knn = Call KNeighborsClassifier()
Call Knn.fit() with `X_train` and `Y_train` parameters
State 4: Take a sample point `x_new` using numpy - arrays
A sample point → `x_new`(numpy - array) = [13.716864, 864.196899, 192.78521, 11. 769211]
This sample point has values of $(u_1, u_2, u_3, u_4, u_5)$
Step 5: Call predict() function with `x_new` point as input parameter
The output obtained is index=1, which is, surface number = 2.
Here, Index 0: Surface 1, Index 1: Surface 2, Index 2: Surface 3
This algorithm may be utilized for selecting a specific surface from a set of multiple surfaces.

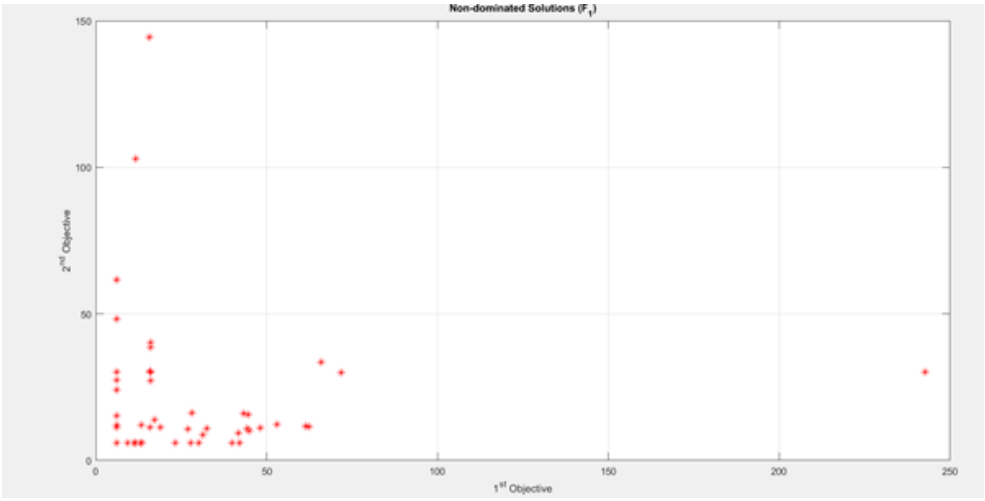


FIG. 2.1. *Output of Multi-Objective Program 1*

- Coefficients (f_i) used in surface 2 are given below:

$$f_{02} = 120, f_{12} = 3, f_{22} = 0.65, f_{32} = -1.87063, f_{42} = 0.08, f_{52} = 0.09, f_{62} = 0.00085$$

- Objective 1:

$$z_1 = ep_1 + \sqrt{((f_6 \times x^6) + (f_5 \times x^5) + (f_4 \times x^4) + (f_3 \times x^3) + (f_2 \times x^2) + (f_1 \times x^1) + f_0)} \quad (2.2)$$

- Objective 2:

$$z_2 = \sqrt{((f_{62} \times x^6) + (f_{52} \times x^5) + (f_{42} \times x^4) + (f_{32} \times x^3) + (f_{22} \times x^2) + (f_{12} \times x^1) + f_{02})} \quad (2.3)$$

2.2.2. Multi-Objective Program 2. The multi-objective program 2 contains objectives Z_1 and Z_2 for two selected surfaces, which uses a parameter ep_1.The parameter of f_1 of surface 1 has been changed here, as compared to multi-objective program 1. The output obtained by running NSGA II for these objectives is shown in Fig.2.2.

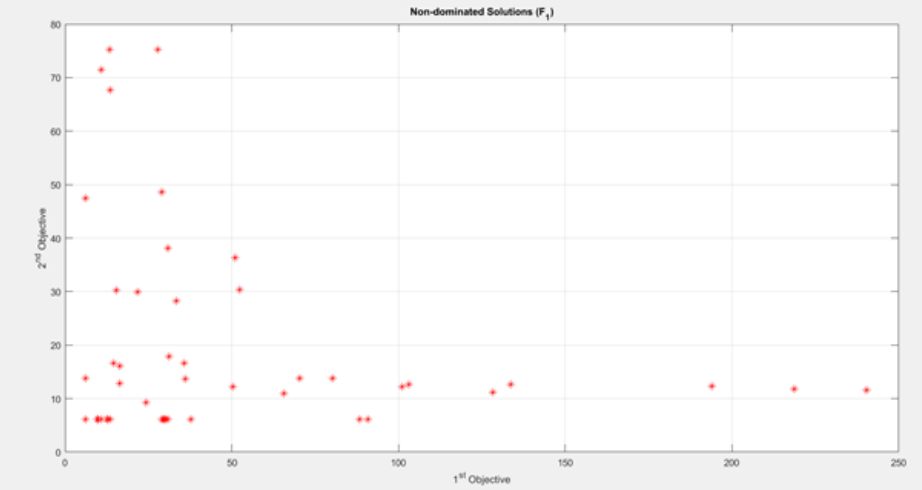


FIG. 2.2. *Output of Multi-Objective Program 2*

The initial values of parameters and equations used for objectives are given below:

- Coefficients (f_i) used in surface 1 are given below:

$$f_0 = 100, f_1 = 0.0, f_2 = 0.8, f_3 = -1.87063, f_4 = 0.06, f_5 = 0.0, f_6 = 0.0, \text{ and } ep_1 = 6.089$$

- Coefficients (f_i) used in surface 2 are given below:

$$f_{02} = 120, f_{12} = 3, f_{22} = 0.65, f_{32} = -1.87063, f_{42} = 0.08, f_{52} = 0.09, f_{62} = 0.00085$$

- Objective 1:

$$z_1 = ep_1 + \sqrt{((f_6 \times x^6) + (f_5 \times x^5) + (f_4 \times x^4) + (f_3 \times x^3) + (f_2 \times x^2) + (f_1 \times x^1) + f_0)} \tag{2.4}$$

- Objective 2:

$$z_2 = \sqrt{((f_{62} \times x^6) + (f_{52} \times x^5) + (f_{42} \times x^4) + (f_{32} \times x^3) + (f_{22} \times x^2) + (f_{12} \times x^1) + f_{02})} \tag{2.5}$$

2.2.3. Multi-Objective Program 3. The multi-objective program 3 contains objectives Z_1 and Z_2 for selected surface, which uses a parameter ep_1 . Here, objective 1 uses two different parameters q_1 and M_i , it is described by equation (6). The output obtained by running NSGA II for these objectives is shown in Fig. 2.3.

The initial values of parameters and equations used for objectives are given below:

- $ep_1 = 6.089, M_i = 120, q_1 = 7.41$
- Coefficients (f_i) used in surface 2 are given below:

$$f_{02} = 120, f_{12} = 3, f_{22} = 0.65, f_{32} = -1.87063, f_{42} = 0.08, f_{52} = 0.09, f_{62} = 0.00085$$

- Objective 1:

$$z_1 = abs(ep_1 + q_1 \times M_i^2 + q_1 \times M_i \times x^2) \tag{2.6}$$

- Objective 2:

$$z_2 = \sqrt{((f_{62} \times x^6) + (f_{52} \times x^5) + (f_{42} \times x^4) + (f_{32} \times x^3) + (f_{22} \times x^2) + (f_{12} \times x^1) + f_{02})} \tag{2.7}$$

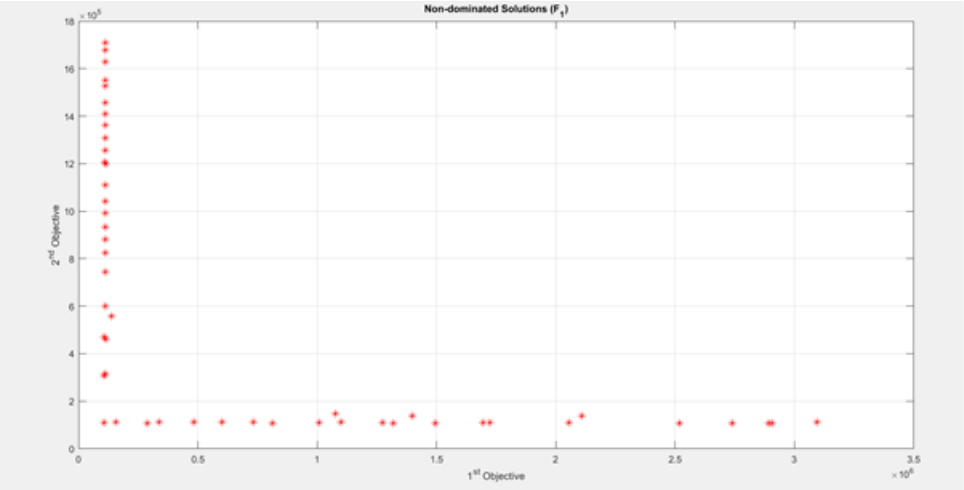


FIG. 2.3. *Output of Multi-Objective Program 3*

2.2.4. Multi-Objective Program 4. The multi-objective program 4 contains objectives Z_1 and Z_2 for selected surface, which uses a parameter ep_1. Here, objective 1 uses two different parameters q_1 and M_i, it is described by equation (8) and objective 2 has different parameters, as compared to multi-objective program 3. The output obtained by running NSGA II for these objectives is shown in Fig. 2.4.

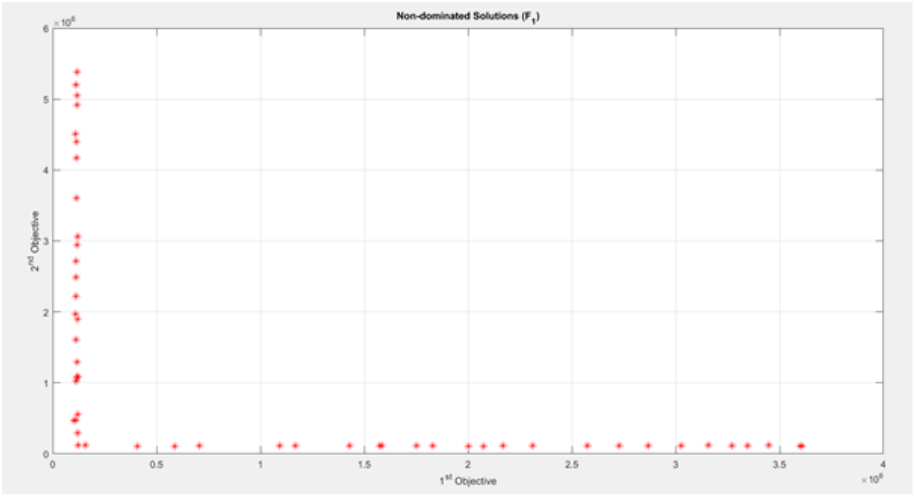


FIG. 2.4. *Output of Multi-Objective Program 4*

The initial values of parameters and equations used for objectives are given below:

- Coefficients (f_i) used in surface 1 are $f_0 = 100, f_1 = 0.0, f_2 = 0.8, f_3 = -1.87063, f_4 = 0.06, f_5 = 0.0, f_6 = 0.0$ and $ep_1 = 6.089, M_i = 120, q_1 = 7.41$
- Objective 1:

$$z_1 = abs(ep_1 + q_1 \times M_i^2 + q_1 \times M_i \times x^2) \tag{2.8}$$

- Objective 2:

$$z_2 = \sqrt{((f_{62} \times x^6) + (f_{52} \times x^5) + (f_{42} \times x^4) + (f_{32} \times x^3) + (f_{22} \times x^2) + (f_{12} \times x^1) + f_{02})} \tag{2.9}$$

TABLE 2.1
Sample input string composed of data streams of portfolio selection data that is used in SHA1 algorithm

S.No.	Input String used in SHA1 algorithm				
	Expected Return	Risk	Parameter T_a	Parameter B_5	Parameter B_{10}
1.	"0.2560	0.1622	0.023	0.5628	0.4372" +
	"0.2786	0.1641	0.023	0.5002	0.4998" +
	"0.3012	0.1698	0.023	0.4377	0.5623" +
	"0.3238	0.1788	0.023	0.3752	0.6248" +
	"0.3464	0.1907	0.023	0.3126	0.6874" +
	"0.3690	0.2050	0.023	0.2501	0.7499" +
	"0.3916	0.2050	0.023	0.1876	0.8124" +
	"0.4142	0.2390	0.023	0.1251	0.8749" +
	"0.4368	0.2579	0.023	0.0438	0.9259" +
	"0.4594	0.2779	0.023	0.0	1.0"

2.3. Using SHA-1 for authentication . For authentication SHA-1 algorithm could be utilized and a sample run with portfolio data is given in Table 2.1. Where, parameter T_a and parameters B_5, B_{10} are different values which are used in portfolio selection problem.
The output obtained after giving the input string, which is described in Table 2.1, to SHA1 algorithm is "DCF5C49BB3160A70788A72A06318FB2BE69BB233".

2.4. Cryptanalysis of the proposed protocol. It is proved that discrete logarithm solution for Kummer surface is equivalent to discrete logarithm solution for Jacobian [1]. With the additional point u_5 the security of the proposed protocol has not been lost moreover an additional direction for ensuring security has been included. The intension is to make the proposed protocol more secure with the help of new point u_5. The encryption scheme based on proposed protocol may utilize machine-learning methods for selecting a specified surface in the case of multiple surfaces. The machine learning algorithm has been tested on a sample data of three extended surfaces. As suggested in [1], ELGamal scheme for encryption may be designed that is based on calculation on the Kummer surface. The addition function provided here could be used for multiplication on the extended surface. Thus, the use of extended surface does not account for any security losses.

3. Generating keys of each round of AES using content-based matrix. This section provides details about the new scheme that is being used for key generation (AES). An overview of the structure of content-based matrix using sample data of portfolio selection is also given in Fig 3.1.

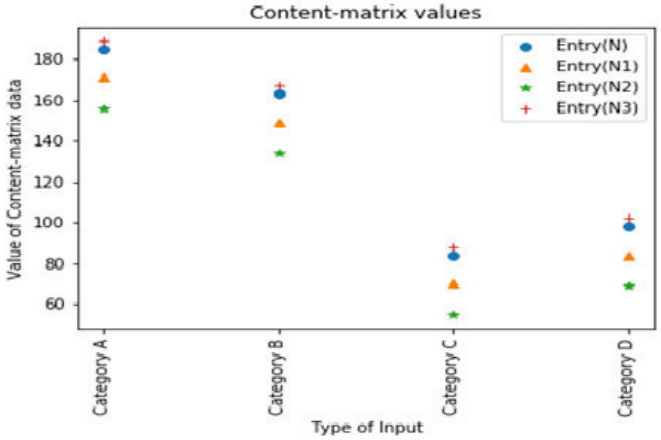


FIG. 3.1. *Content-matrix values*

The formulation of the content-matrix is given next. Here, the input is divided into four categories- category A for characters a-z / A-Z, category B for numbers 0-9, category C for special characters and lastly, the category D for images / videos etc.

Here, a sample frequency distribution of different classes in a sample input is given in Table 3.1. It contains assumed values of weight (x) and the mapping output (y) is found using the following equation:

$$y = x^3 + x^2 + w$$

$$N_o = y \bmod 255$$

The following constants are used here: $epk_1 = 13.89$, $epk_2 = 29.17$ and $epk_3 = -4.089$. The values of N_1, N_2 and N_3 are computed as described below:

$$N_1 = N - epk_1$$

$$N_2 = N - epk_2$$

$$N_3 = N - epk_3$$

TABLE 3.1
Output of content-matrix

S.No.	Type of input	Frequency(x)	Weight(w)	Mapping output(y)	Entry (N_0)	Entry(N_1)	Entry(N_2)	Entry(N_3)
1	Category A	34	15	40475	185	171.11	155.825	189.089
2	Category B	50	8	6252508	163	149.11	133.825	167.089
3	Category C	60	9	216069	84	70.11	54.825	88.089
4	Category D	12	11	1883	98	84.11	68.825	102.089

The reduced content-matrix values are based upon only three categories: category-$alpha_1$ =category 1, category-$alpha_2$ = category 2 and category-$alpha_3$ = category 3. The output obtained are given in Table 3.2 and Fig. 3.2.

TABLE 3.2
Output of content-matrix

S.No.	Type of input	Entry(N)	Entry(N_1)	Entry(N_2)	Entry(N_3)
1	Category-α_1	185	171.11	155.825	189.089
2	Category-α_2	163	149.11	133.825	167.089
3	Category-α_3	91	77.11	61.825	95.089

The following equations are used to compute optimal values :

$$cost_1 = a_1\alpha_1^2 + b_1\alpha_1 + c_1 \tag{3.1}$$

$$cost_2 = a_2\alpha_2^2 + b_2\alpha_2 + c_2 \tag{3.2}$$

$$cost_3 = a_3\alpha_3^2 + b_3\alpha_3 + c_3 \tag{3.3}$$

$$\texttt{Total\_cost} = cost_1 + cost_2 + cost_3 \tag{3.4}$$

$$189.089 \leq a_1 \leq 155.825$$

$$167.089 \leq a_2 \leq 133.825$$

$$95.089 \leq a_3 \leq 61.825$$

$$461.0 \leq \texttt{Total\_cost} \leq 361.0$$

The optimized values obtained after running classical Lagrangian multiplier method with sample input-values of a_i, b_i and c_i and using the inequality constraints values of a_1, a_2 and a_3 described above, are presented in Table 3.3 and Fig. 3.3.

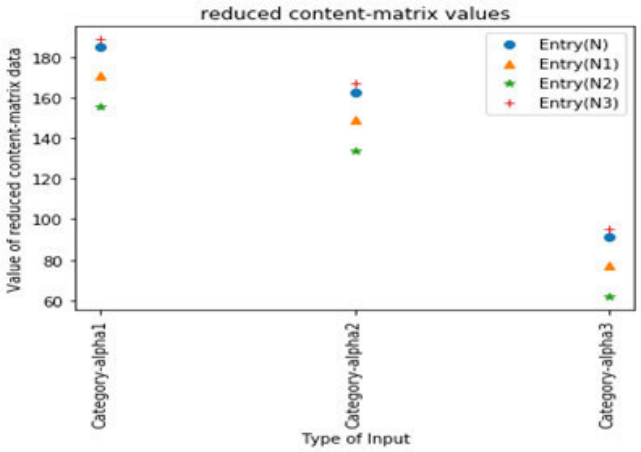


FIG. 3.2. *Reduced content-matrix values*

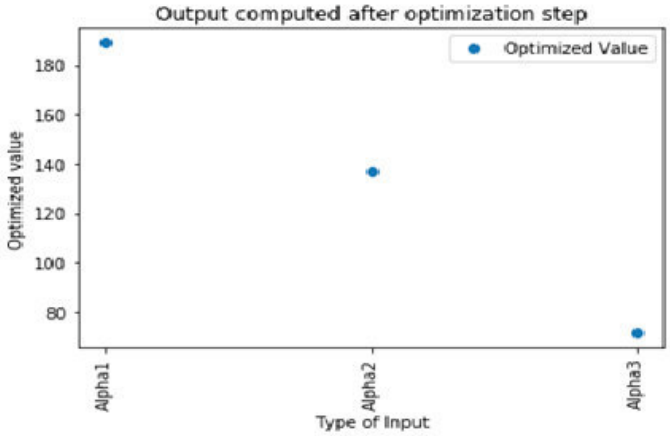


FIG. 3.3. *Output computed after optimization step*

TABLE 3.3
Optimized values of different categories

S.No.	Category	Optimized value
1	Category-α_1	189.0809
2	Category-α_2	137.2910
3	Category-α_3	71.6765

3.1. Cryptanalysis of the proposed scheme. Different kinds of cryptanalysis for AES (Rijndael) is provided in [28]. Using another technique based on calculating partial sum, the complexity of AES can be decreased [28]. The attacks on 7 and 8 rounds can be achieved by incorporating the information about extra texts. The next area to be considered is Key generation. Various undesired behaviour for key generation is discussed in [28].

This paper has made presentation about a novel scheme that may be adopted to conceal the information which could prove fruitful in increasing complexity of Key generation. Specifically, the structure of content-based matrix and selecting optimized values play a vital role in hiding the information. Since Key generation has a small number of non-linear components, the schemes presented here has given new dimension in this direction.

Algorithm 4 New Key Expansion Algorithm

The new modified key expansion algorithm is given below:
Modified-Key-Expansion(byte x[4*y], word out[4 * 11], y)
start
word t_{11} // (temporary variable)
j= 0
While (j ¡ y)
out [i] = word(x[4 * j], key[4 * j+1], x[4 * j+2], x[4 * j+3]
j = j+1
end of while
j = y
While (j ¡ (4 * 11))
t_{11} = out [j-1] // stage 1
if (j mod y == 0) **then**
    t_{11} = Rotate_Word(temp) // stage 2
    t_{11} = Substitute_Word(temp) // stage 3
    t_{11} = (t_{11} xor content matrix_data_entry) // stage 4
    t_{11} = t_{11} xor R_constant[j/4] // stage 5
elseif (y ¿ 6 and j mod y = 4)
    t_{11} = Substitute_Word(temp) // stage 6
end if
out[j] = out[j-y] xor t_{11} // stage 7 and stage 8
j = j + 1
end of while
end

4. Result Analysis and Discussion. The following section describes the various outputs obtained after executing extended Kummer surface algorithm explained in section 2.1. An overview of the results obtained for existing Kummer surface is also given. The next section has a description of data obtained after executing new key expansion (AES) algorithm. Lastly, an overview of the results obtained with LIM algorithm are provided.

4.1. Existing Kummer Surface Results. This section briefly presents results which are obtained for existing and extended Kummer surface. The results are given in Table 4.1 and Fig 4.1 for existing Kummer surface- surface 1.

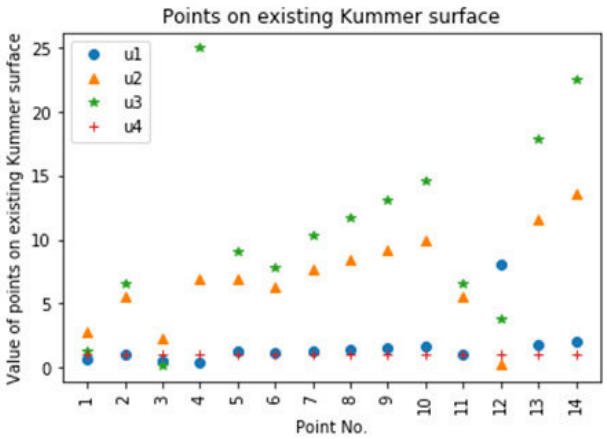


FIG. 4.1. *Points on existing Kummer surface (Surface 1)*

TABLE 4.1
Sample points on Existing Kummer Surface(u_1, u_2, u_3, u_4, u_5)(Surface 1)

S.No.	u_1	u_2	u_3	u_4
1	0.5915	2.7999	1.2000	1.0000
2	1.0000	5.5856	6.5497	1.0000
3	0.5229	2.2824	0.1000	1.0000
4	0.4217	6.8683	25.0000	1.0000
5	1.2000	6.9648	9.0642	1.0000
6	1.1000	6.2700	7.7984	1.0000
7	1.3000	7.6731	10.3607	1.0000
8	1.4000	8.3982	11.7018	1.0000
9	1.5000	9.1445	13.1036	1.0000
10	1.6000	9.9176	14.5872	1.0000
11	1.0000	5.5856	6.5460	1.0000
12	8.0000	0.2864	3.7636	1.0000
13	1.8000	11.5840	17.9428	1.0000
14	2.0000	13.6073	22.5198	1.0000

Similarly, sample data points are computed for surface 2 and surface 3.

4.1.1. Extended Kummer Surface Results. This section presents the results which are obtained after executing extended Kummer surface algorithm. This algorithm contains a newly added point (u_5). The input points z and y on the extended Kummer surface (surface 1) are given in Table 4.2 and are drawn in Fig. 4.2 and Fig. 4.3.

TABLE 4.2
Value of input point (z) and input point (y) on extended Kummer surface (Surface 1)

S.No.	z(u_1)	z(u_2)	z(u_3)	z(u_4)	z(u_5)	y(u_1)	y(u_2)	y(u_3)	y(u_4)	y(u_5)
1	1.0000	5.5856	6.5460	1.0000	0.8000	8.0000	0.2864	3.7636	1.0000	0.9800
2	1.1000	6.2700	7.7984	1.0000	0.8000	1.3000	7.6731	10.3607	1.0000	0.9800
3	1.2000	6.9648	9.0642	1.0000	0.8000	1.1000	6.2700	7.7984	1.0000	0.9800
4	0.5915	2.7999	1.2000	1.0000	0.8000	1.0000	5.5856	6.5497	1.0000	0.9800
5	8.0000	0.2864	3.7636	1.0000	0.8000	2.5033	13.0410	5.0000	1.0000	0.9800
6	1.5000	9.1445	13.1036	1.0000	0.8000	1.6000	9.9176	14.5872	1.0000	0.9800
7	0.4217	6.8683	25.0000	1.0000	0.8000	1.2000	6.9648	9.0642	1.0000	0.9800
8	1.4000	8.3982	11.7018	1.0000	0.8000	1.5000	9.1445	13.1036	1.0000	0.9800
9	0.5229	2.2824	0.1000	1.0000	0.8000	0.4217	6.8683	25.0000	1.0000	0.9800
10	1.6000	9.9176	14.5872	1.0000	0.8000	1.0000	5.5856	6.5497	1.0000	0.9800
11	1.0000	5.5856	6.5497	1.0000	0.8000	0.5229	2.2824	0.1000	1.0000	0.9800
12	1.8000	11.5840	17.9428	1.0000	0.8000	2.0000	13.6073	22.5198	1.0000	0.9800
13	1.3000	7.6731	10.3607	1.0000	0.8000	1.4000	8.3982	11.7018	1.0000	0.9800
14	2.0000	13.6073	22.5198	1.0000	0.8000	0.5915	2.7999	1.2000	1.0000	0.9800

Output points obtained are given in Table 4.3 and drawn in Fig.4.4. The data points obtained are sorted output according to point (u_5). Similar outputs obtained for surface 2 and surface 3. Moreover, the data points are sorted according to point (u_5). An effort is made to present output points pictorially in ascending order of point (u_5) such that the variation of point (u_5) can be studied with values of other points (u_1, u_2, u_3, u_4). A meaningful inference about the variation of point (u_5) can be achieved with large set of input points.

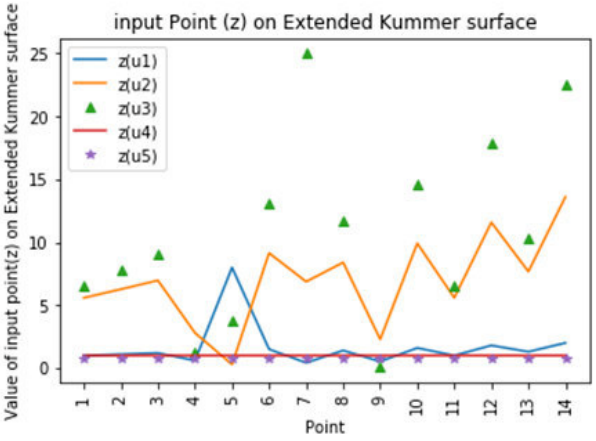


FIG. 4.2. *Input point (z) on Extended Kummer surface (surface 1)*

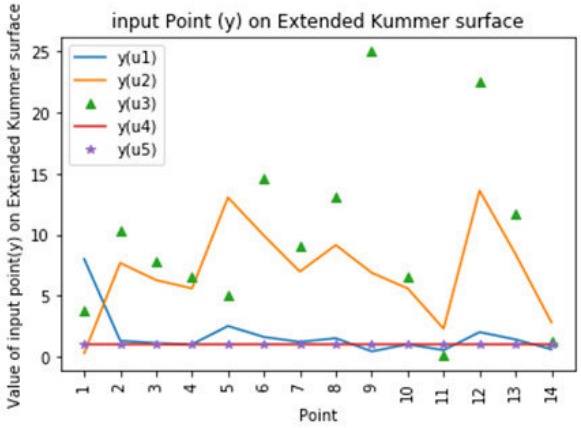


FIG. 4.3. *Input point (y) on Extended Kummer surface (surface 1)*

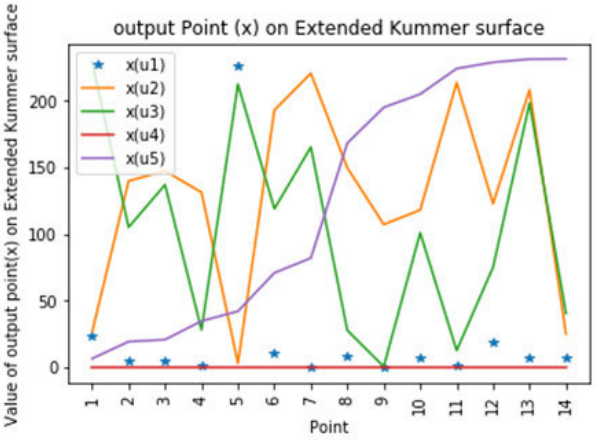


FIG. 4.4. *Output point (x) Extended Kummer surface (surface 1)*

TABLE 4.3
Output values of extended Kummer Surface($x(u_1), x(u_2), x(u_3), x(u_4), u_5$)(Surface 1)

S.No.	$x(u_1)$	$x(u_2)$	$x(u_3)$	$x(u_4)$	$x(u_5)$
1	24.0000	26.8055	229.3580	0.0000	6.7970
2	4.7190	139.9495	105.2500	0.0000	19.4924
3	4.7520	147.4385	137.1388	0.0000	20.9641
4	1.0496	131.3605	28.2947	0.0000	34.8777
5	225.6336	3.2089	212.4693	0.0000	42.2683
6	10.8000	192.9697	119.0908	0.0000	71.0500
7	0.6403	220.6695	165.3555	0.0000	82.1640
8	8.8200	149.8854	27.9238	0.0000	168.2817
9	0.3459	107.3347	0.7500	0.0000	195.0867
10	7.6800	118.1619	101.0610	0.0000	205.0549
11	1.5687	213.6205	12.8696	0.0000	224.2009
12	19.4400	122.8515	75.2773	0.0000	228.8854
13	7.0979	208.3531	198.3577	0.0000	231.2859
14	7.0981	25.2546	40.7057	0.0000	231.4460

4.2. Results obtained with new Key expansion algorithm in different stages. This section provides an overview of the outputs obtained after different stages in new Key expansion algorithm. For simplicity of the presented output, the decimal values on a scale of (value/10^9) is chosen. Table 4.4 has output of stage 4 and its (new Key expansion) diagram along with two trend lines - trade line 1(power trend line with forecast of 2.0 periods, equation used: y = $34092 \times x^{00743}$, $R^2 = 0.1335$, here y is output, x is input parameter for trend line and R^2 is R-squared value used for trend line 1) and trend line 2 (polynomial trend with forecast of 2.0 periods) is given in Fig.4.5.

TABLE 4.4
Output data obtained from new Key expansion algorithm: Stage 4 and Stage 6

S. No.	O(hex):Stage 4	O(dec):Stage 4	O(hex):Stgae 6	O(dec):Stage 6
1	bf85ef55	3.2132	be85ef55	3.1964
2	bdc5d7d4	3.1839	bfc5d7d4	3.2174
3	fdd76f5f	4.2588	f9d76f5f	4.1916
4	ff97ef7d	4.2881	f797ef7d	4.1539
5	ffe5c7ff	4.2932	efe5c7ff	4.0248
6	fdd5e6dc	4.2587	ddd5e6dc	3.7218
7	bf954f5e	3.2142	ff954f5e	4.2880
8	ffdfd75d	4.2929	7fdfd75d	2.1454
9	bff5ee5c	3.2206	a4f5ee5c	2.7676
10	ffbdcf74	4.2906	c9bdcf74	3.3847

Table 4.4 has also output of stage 6 and its (new Key expansion) diagram along with two trend lines - trade line 1(power trend line with forecast of 2.0 periods, equation used: y = $3.7022 \times x^{0.049}$, $R^2 = 0.0266$) and trend line 2 (polynomial trend with forecast of 2.0 periods) is given in Fig.4.6.

In Table 4.4, O(hex):Stage 4 denotes Output of new Key expansion algorithm: stage 4 (Hex value), O(dec):Stage 4 denotes Output of new Key expansion algorithm : stage 4 (Decimal value / 10^9), O(hex):Stage 6 denotes Output of new Key expansion algorithm : stage 6 (Hex value), and O(dec):Stage 6 denotes Output of new Key expansion algorithm : stage 6 (Decimal value / 10^9).

In Table 4.5 ,O(hex):Stage 7 denotes Output of new Key expansion algorithm : stage 7 (Hex value), O(dec):Stage 7 denotes Output of new Key expansion algorithm : stage 7 (Decimal value / 10^9), O(hex):stage

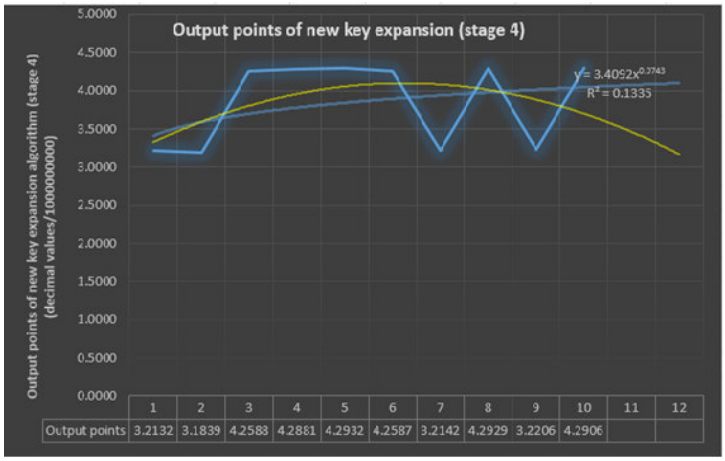


FIG. 4.5. *Output points of new key expansion (stage 4)*

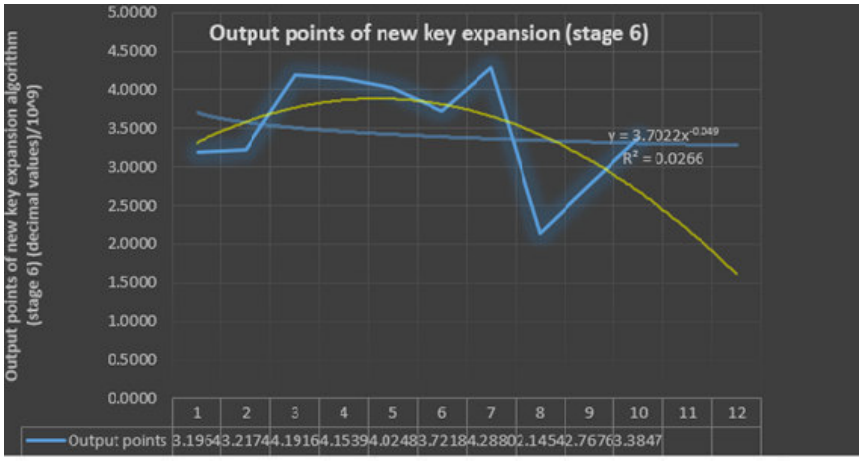


FIG. 4.6. *Output points of new key expansion (stage 6)*

8 denotes Output of new Key expansion algorithm : stage 8 (Hex value), and O(dec):stage 8 denotes Output of new Key expansion algorithm : stage 8 (Decimal value / 10^9).

Table 4.5 also has output of stage 7 and its (new Key expansion) diagram along with two trend lines trade line 1(power trend line with forecast of 2.0 periods, equation used: $y = 0.9656 \times x^{0.0177}$, $R^2 = 0.0002$) and trend line 2 (polynomial trend with forecast of 2.0 periods) is given in Fig.4.7.

Table 4.5 has output of stage 8 and its (new Key expansion) diagram along with two trend lines-trade line 1(power trend line with forecast of 2.0 periods, equation used: $y = 0.9656 \times x^{0.0177}$, $R^2 = 0.0002$) and trend line 2 (polynomial trend with forecast of 2.0 periods) is given in Fig.4.8.

As evident from the outputs presented in Table 4.4 and Table 4.5, the trade lines are having different patterns in the existing and new Key expansion algorithm. The output of stage 1, 2, 3 and 5 have same outputs in both existing and new Key expansion algorithm, and are given in Table 4.6. The stage 4 is available only in the new Key expansion. The outputs in the stage 6 are altogether different in new algorithm. The variation of points is different in new stage 7, whereas the trend lines are having similar patterns in this stage. Lastly, the stage 8 has different variation of points in new Key expansion, moreover trade lines depicts different behaviour. Thus, the proposed scheme adopted can be utilized by a designer in order to provide better security by having newer variation patterns, which are different from existing AES Key expansion algorithm.

TABLE 4.5
Output data obtained from new Key expansion algorithm: Stage 7 and Stage 8

S.No.	O(hex):Stage 7	O(dec):Stage 7	O(hex):Stage 8	O(dec):Stage 8
1	2b7e1576	0.7297	2b7e1576	0.7297
2	28aed2a6	0.6825	28aed2a6	0.6825
3	abf71588	2.8851	abf71588	2.8851
4	09cf4f3c	0.1646	09cf4f3c	0.1646
5	95fbfa43	2.5163	95fbfa43	2.5163
6	bd5528e5	3.1765	bd5528e5	3.1765
7	16a23d6d	0.3797	16a23d6d	0.3797
8	1f6d7251	0.5273	1f6d7251	0.5273
9	2a3e2d97	0.7087	2a3e2d97	0.7087
10	976b572	0.1588	976b572	0.1588
11	81c9381f	2.1774	81c9381f	2.1774
12	9ea44a4e	2.6616	9ea44a4e	2.6616
13	d3e942c8	3.5553	d3e942c8	3.5553
14	448247ba	1.1494	448247ba	1.1494
15	c54b7fa5	3.3101	c54b7fa5	3.3101
16	5bef35eb	1.5424	5bef35eb	1.5424
17	247eadb5	0.6123	247eadb5	0.6123
18	60fceaf	0.1017	60fceaf	0.1017
19	a5b795aa	2.7803	a5b795aa	2.7803
20	fe58a041	4.2672	fe58a041	4.2672
21	cb9b6a4a	3.4160	cb9b6a4a	3.4160
22	ab678045	2.8757	ab678045	2.8757
23	ed015ef	0.2485	ed015ef	0.2485
24	f088b5ae	4.0355	f088b5ae	4.0355
25	164e8c96	0.3742	164e8c96	0.3742
26	bd29cd3	0.1984	bd29cd3	0.1984
27	b3f9193c	3.0194	b3f9193c	3.0194
28	4371ac92	1.1315	4371ac92	1.1315
29	e9dbc3c8	3.9235	e9dbc3c8	3.9235
30	54f2cf1b	1.4252	54f2cf1b	1.4252
31	e7bd627	0.2430	e7bd627	0.2430
32	a47a7ab5	2.7595	a47a7ab5	2.7595
33	9641495	0.1576	9641495	0.1576
34	c2f6db8e	3.2710	c2f6db8e	3.2710
35	25fdda9	0.0398	25fdda9	0.0398
36	8187771c	2.1731	8187771c	2.1731
37	32f1fac9	0.8547	32f1fac9	0.8547
38	f072147	0.2521	f072147	0.2521
39	d5fa2cee	3.5899	d5fa2cee	3.5899
40	547d5bf2	1.4175	547d5bf2	1.4175

4.3. Output of LIM. This section presents computed output values of LIM obtained from histogram data taken from different extended Kummer surface. The values of LIM for data used in extended Kummer surface 1 is given in Table 4.7, the diagrams showing histograms for first five values (set 1) and last five values (set 2) of Table 4.7 are given in Fig.4.9 and Fig.4.10. The computed values of LIM for surface 1 are depicted in Fig. 4.11. The algorithm of LIM is presented in [10].

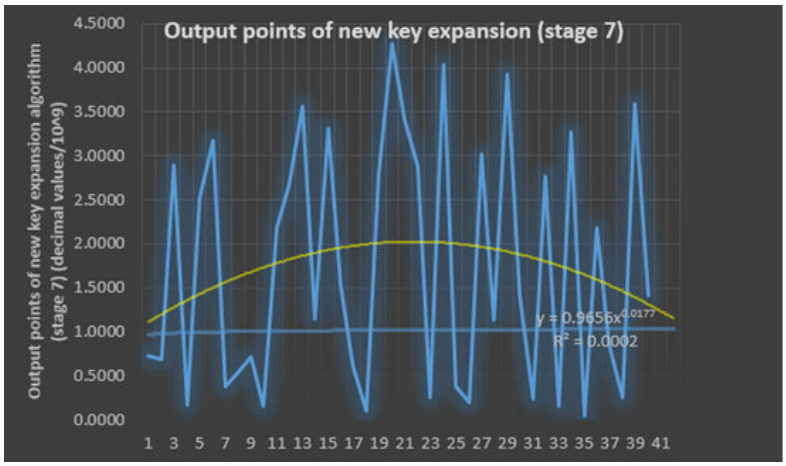


FIG. 4.7. *Output points of new key expansion (stage 7)*

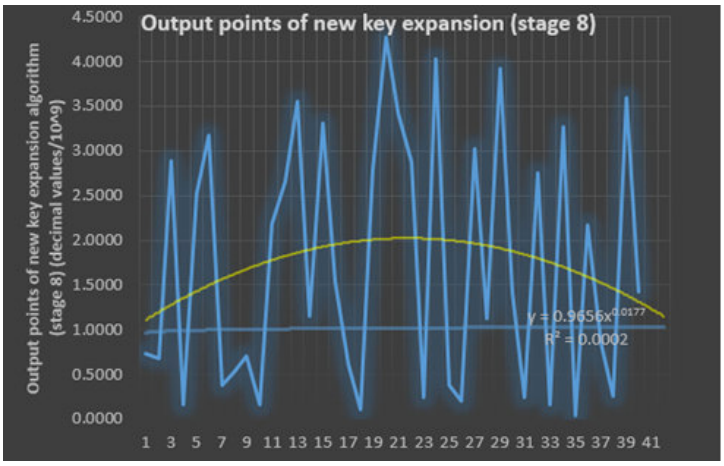


FIG. 4.8. *Output points of new key expansion (stage 8)*

TABLE 4.6
Output of stage 1, stage 2 and stage 3 (existing and new Key expansion)

S.No.	Variable t11 (stage 1)	`Output after Rotate_word ( ) (stage 2)`	`Output after Subtitute_word ( ) (stage 3)`
1	09cf4f3c	cf4f3c09	8a84eb01
2	1f6d7251	6d72511f	3c40d1c0
3	9ea44a4e	a44a4e9e	49d62fb
4	5bef35eb	ef35eb5b	df96e939
5	fe58a041	58a041fe	6ae083bb
6	f088b5ae	88b5aef0	c4d5e48c
7	4371ac92	71ac9243	a3914f1a
8	a47a7ab5	7a7ab5a4	dadad549
9	8187771c81	87771c81	17f5acc
10	547d5bf2	7d5bf254	ff398920

As given in Table 4.7, Step 1 denotes Histogram Data (step 1), Step 3 denotes Sorted Data computed in the Algorithm (step 3), and Cval(LIM) denotes Computed value of LIM. The values of LIM for data used in extended Kummer surface 2 are 0.3451010 and 0.477174 for two different sample data points. Same values of LIM is obtained for surface 3.

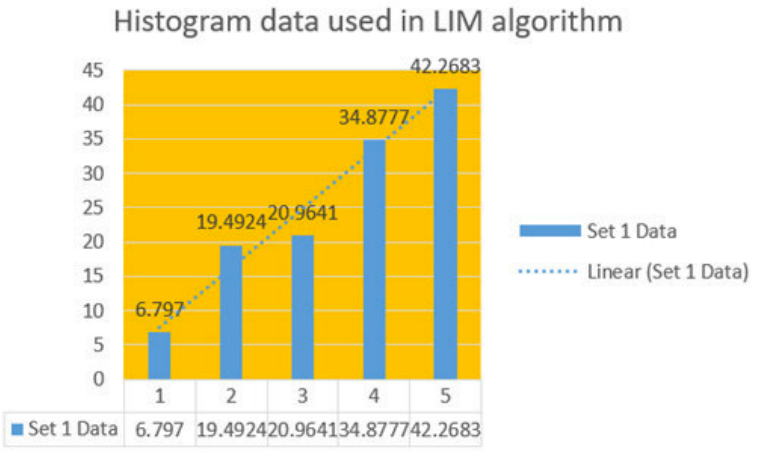


FIG. 4.9. *Histogram data used in LIM algorithm (surface 1) (set 1 data)*

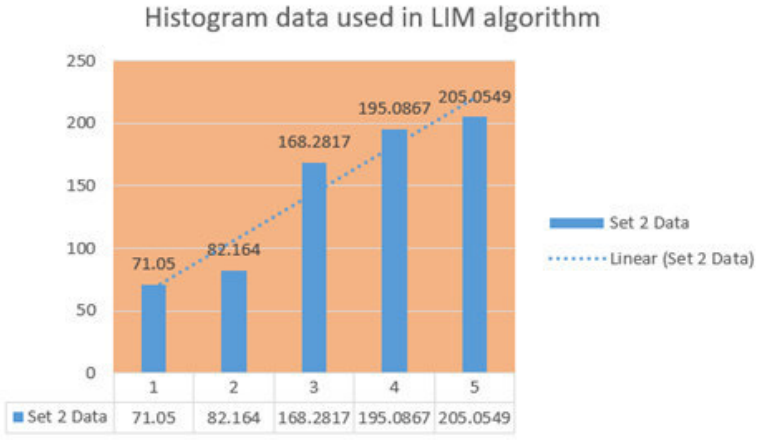


FIG. 4.10. *Histogram data used in LIM algorithm (surface 1) (set 2 data)*

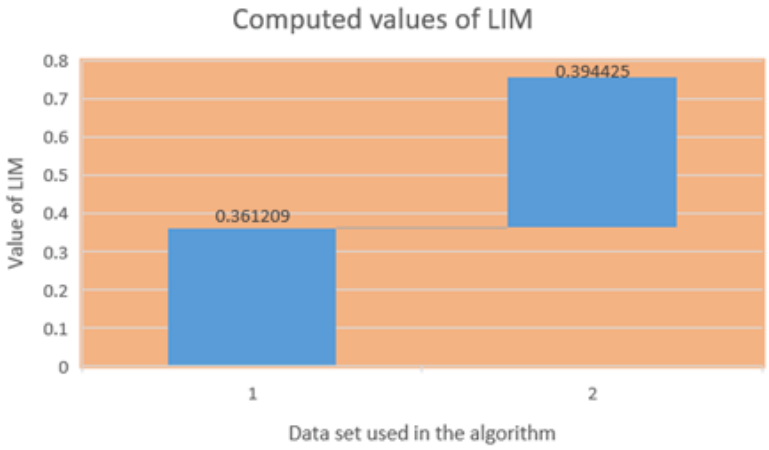


FIG. 4.11. *Computed values of LIM (surface 1)*

TABLE 4.7
Output of LIM (surface 1)

S. No.	Step 1	Step 3	Cval(LIM)
1	6.797	16.080608	0.361209
	19.4924	46.115883	
	20.9641	49.597691	
	34.8777	82.51503	
	42.2683	100	
2	71.05	34.649258	0.394425
	82.164	40.069271	
	168.2817	82.06665	
	195.0867	95.138763	
	205.0549	100	

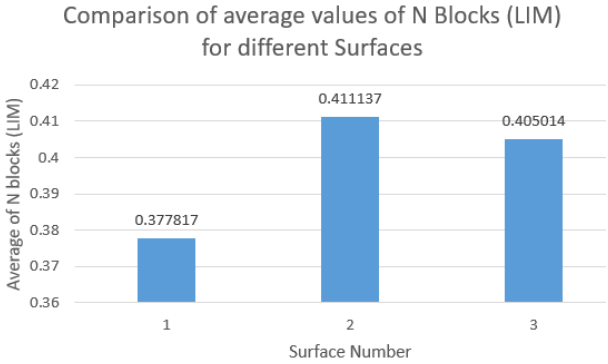


FIG. 4.12. *Comparison of average values of N Blocks (LIM) for different surfaces)*

4.3.1. Importance of computed values of LIM. An encrypting method, which has randomly distributed cipher texts inside the group of text used while encrypting, would be classified as a potential nice method encrypting. A computed value of LIM, that is, close to a value of 0.5 is an indication of randomly distributed [10]. Moreover, a drift from this value, significance occurrence of a kind of pattern in the cipher text [10]. The average values shown in Fig. 4.12 signifies that all the three surfaces are different, and there is a possibility of occurrence of a pattern, because these values drift from the value of 0.5.

5. Conclusion. This paper has a secure structure which contains mainly two aspects of a cryptographic system. Firstly, a scheme comprising of newly added point (u_5) in the kummer surface and then how this information inclusion could be utilized for a machine-learning algorithm in selecting an appropriate surface. Here, the selection of a particular surface based on a machine learning technique is provided. The work based on kummer surfaces and building a Fast Diffie-Hellman protocol is previously given in earlier papers. The focus of the proposed scheme is to enhance the security of the above-mentioned work by selecting an additional point (u_5) on the surface. Later, selection of multiple surfaces using NSGA II algorithm is given. Secondly, an encryption based on AES is taken. A new scheme for generating modified key-expansion in AES algorithm is described. The scheme builds a content-matrix using frequencies of different categories in the input message. Next, optimized values of the entries are chosen by running classical Lagrangian multiplier method. The selected entries are utilized in the modified key-expansion algorithm. Lastly, a brief overview of LIM index is given.

Acknowledgement: This publication is an outcome of the R & D work under taken project under the Visvesarvaraya PhD Scheme of Ministry of Electronics and IT, Government of India, being implemented by the Digital India Corporation.

REFERENCES

[1] N. P. Smart, and S. Siksek, *A Fast Diffie Hellman Protocol in Genus 2*, in Journal of cryptology, vol.12(1), 1999, pp. 67-73.
[2] H. Nover, *Algebraic cryptanalysis of AES: an overview*, in University of Wisconsin, USA, (2005).
[3] L. Ning, L.Kanfeng, L. Wenliang, and D. Zhongliang, *A joint encryption and error correction method used in satellite communications*, in China communications, vol. 11(3), (2014),pp. 70-79.
[4] Y. Dodis, R. Leonid, and S. Adam, *Fuzzy extractors: How to generate strong keys from biometrics and other noisy data*, in International conference on the theory and applications of cryptographic techniques, (2004), pp. 523-540.
[5] C. Peikert, and W. Brent, *Lossy trapdoor functions and their applications*, in SIAM Journal on Computing, vol.40(6), (2011), pp.1803-1844.
[6] A. C. Yao, *Theory and application of trapdoor functions*, in 23rd Annual Symposium on Foundations of Computer Science SFCS'08, (1982), pp.80-91.
[7] L. Blum, B. Manuel, and S. Mike, *A simple unpredictable pseudo-random number generator*, in SIAM Journal on computing, vol. 15(2), (1986), pp.364-383.
[8] P. S. Mehra, M. N. Doja, and B. Alam, *Codeword Authenticated Key Exchange (CAKE) light weight secure routing protocol for WSN*, International Journal of Communication Systems, 32 (2019).
[9] R. Cramer, and S. Victor, *A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack*, in Annual International Cryptology Conference, (1982), pp. 13-25.
[10] V. Karuvandan, C. Senthamarai and P. Shantharajah, in *Cryptanalysis of AES-128 and AES-256 block ciphers using lorenz information measure*, in Int. Arab J. Inf. Technology, vol. 13(6B), (2016), pp. 1054-1060.
[11] N. Ferguson, K. John, L. Stefan, S. Bruce, S. Mike, W.David, and W.Doug, *Improved cryptanalysis of Rijndael*, in International Workshop on Fast Software Encryption, 2000, pp. 213-230.
[12] A. Kaminsky, K. Michael, and R. Stanisaw, *An overview of cryptanalysis research for the advanced encryption standard*, in MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM, (2010), pp.1310-1316.
[13] S. Chen, L. Rodolphe, L. Jooyoung, S. Yannick, and S. John, *Minimizing the two-round EvenMansour cipher*, in Journal of Cryptology, vol. 31(4), (2018), pp. 1064-1119.
[14] M. Masoumi, and M. Hadi Rezayati, *Novel approach to protect advanced encryption standard algorithm implementation against differential electromagnetic and power analysis*, in IEEE Transactions on Information Forensics and Security, Vol. 10(2), 2015, pp.256-265.
[15] F. Courbon, JJ. A. Fournier, P. L. Moundi, and A. Tria, *Combining image processing and laser fault injections for characterizing a hardware AES*, in IEEE transactions on computer-aided design of integrated circuits and systems, vol. 34(6), (2015), pp.928-936.
[16] M. Taha, and P. Schaumont, *Key updating for leakage resiliency with application to AES modes of operation*, in IEEE transactions on information forensics and security, vol. 10(3), (2015), pp.519-528.
[17] C. H. Baek, J.H.Cheon, and H.Hong, *White-box AES implementation revisited*, in Journal of Communications and Networks, vol. 18(3), (2016), pp.273-287.
[18] R. L. Rivest, *Cryptography and machine learning*, in International Conference on the Theory and Application of Cryptology, (1991), pp.427-439.
[19] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, *Machine learning classification over encrypted data*, in NDSS, (2015).
[20] T. Graepel, L. Kristin, and M. Naehrig, *ML confidential: Machine learning on encrypted data*, in International Conference on Information Security and Cryptology,(2012), pp. 1-21.
[21] F. M. Barbosa, A.R.S.F. Vidal, H.L.S. Almeida, and F.L. de Mello, *Machine Learning Applied to the Recognition of Cryptographic Algorithms Used for Multimedia Encryption*, in IEEE Latin America Transactions, vol. 15(7),(2017), pp. 1301-1305.
[22] H. Maghrebi, T.Portigliatti, and E. Prouff, *Breaking cryptographic implementations using deep learning techniques*, in International Conference on Security, Privacy, and Applied Cryptography Engineering, (2016), pp. 3-26.
[23] J.S. Muller, *Explicit Kummer surface formulas for arbitrary characteristic*, in LMS Journal of Computation and Mathematics, vol. 13, (2010), pp. 47-64.
[24] A. Garbagnati, and A. Sarti, *Kummer surfaces and K3 surfaces with* $(\frac{Z}{2Z})^4$ *symplectic action*, in Rocky Mountain Journal of Mathematics, vol. 46(4),(2016), pp. 1141-1205.
[25] Standard, Advance Encryption, *Federal information processing standards publication 197*, in FIPS PUB, (2001), pp. 46-53.
[26] C. Kumar, and M.N.Doja, *A Novel Framework for Portfolio Selection Model Using Modified ANFIS and Fuzzy Sets*, in Computers, vol. 7(4), (2018), pp. 57-64.
[27] R. K. Chahar, G. Datta, and N. Rajpal, *Design of a new Security Protocol*, in International Conference on Computational Intelligence and Multimedia Applications, vol. 4, (2007), pp. 132-136.
[28] N. Ferguson, K. John, L. Stefan, S. Bruce, S. Mike, W. David, and W. Doug, *Improved cryptanalysis of Rijndael*, in International Workshop on Fast Software Encryption, (2000), pp. 213-230.
[29] K. Deb, A.Pratap, S. Agarwal, and T. A. M. T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, in IEEE transactions on evolutionary computation, vol. 6(2), (2002), pp. 182-197.

Edited by: Khaleel Ahmad
Received: Nov 25, 2018
Accepted: Feb 11, 2019

DOI 10.12694/scpe.v20i1.1473
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 181–189. http://www.scpe.org © 2019 SCPE

AN EFFICIENT ZERO-KNOWLEDGE PROOF BASED IDENTIFICATION SCHEME FOR SECURING SOFTWARE DEFINED NETWORK

HAMZA MUTAHER* AND PRADEEP KUMAR †

Abstract. Researchers and enterprise networks are extensively adopting Software Defined Networking (SDN) due to its feature of decoupling data and control planes from network devices which enable them to implement new networking ideas to solve networking issues like the lack of security. Communication between data and control planes in SDN faces various security issues where many users in data plane approach controller device in control plane to gain networking policies. In this paper, we proposed an efficient Zero-knowledge proof based identification scheme for securing the SDN controller during data and control plane communication. This scheme ensures that only users who prove their knowledge about secrecy without revealing the actual secret or any other information about it can communicate with the controller. The computation cost, communication cost and storage overhead analysis are discussed along with the security analysis to validate the efficiency of the proposed work.

Key words: SDN, Controller, Identification, Zero Knowledge Proof, data plane, control plane.

AMS subject classifications. 68M12

1. Introduction. Software Defined Networking (SDN) is one of the most important networking topics discussed in recent years [1]. SDN is defined in the article of Open Network Foundation (ONF) [2] as: In the SDN architecture, the control and data planes are decoupled, Network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications. The Idea of SDN is to separate the control plane from the data plane in network devices. Control plane manages the network policies and has a wide view of the whole network, where the data plane follows the rules and policies sat up by control plane such as packet forwarding policies [3]. The architecture of SDN contents of 3 layers, Application layer, Control layer and the Infrastructure layer. The application layer contains the network applications like network management applications and business applications, and it communicates with the control layer through northbound APIs. Control layer contains the brain of the network (the controller) which has the complete ability of network management and runs the Network Operating System (NOS). Infrastructure layer contains the switching devices. Switching devices are responsible to forward the network packets from Source to destination as per controller decisions and communicate with control layer through southbound APIs. The architecture of SDN illustrated in Figure 1.1. SDN came in to picture to overcome the complexity of the traditional network and to add some new features which facilitate network innovations. These features are explained in Table 1.1 and as follows:

i. Centralization: It is one of the SDN features that ease the deployment of new policies and applications among network devices. It also mitigates the complexity of network management which is not there in a traditional network, i.e. No central point of management. In SDN, the controller is the networking device runs NOS. NOS is responsible for managing and controlling the centralized devices like controller. The controller is responsible for managing the SDN network and has the full knowledge about every device in the network as it is a centralized management point [4].
ii. Scalability: It is an SDN feature that allows the network to have a massive number of devices and enable the network to add a new number of devices into the network as per network needs. Distributed controllers are used to scale up the network and make the network highly available [5].
iii. Heterogeneity: SDN network allows a different type of networks to communicate with each other as per controller decisions. The controller is responsible for this communication as it can manage heterogeneous networks working with different routing protocols. Applications and devices from different platforms also can communicate and exchange services among each other in an SDN network [6].
iv. Programmability: SDN simplifies the network management and innovation by the feature of programmability. Programmability allows the controller to program new rules and deploy them into data plane devices in order of network stability. The third parties can simply program and execute their business applications

*Department of Computer Science and IT, Maulana Azad National Urdu University, India (hamzamutaher@gmail.com)
†Department of Computer Science and IT, Maulana Azad National Urdu University, India (drpkumar1402@gmail.com)

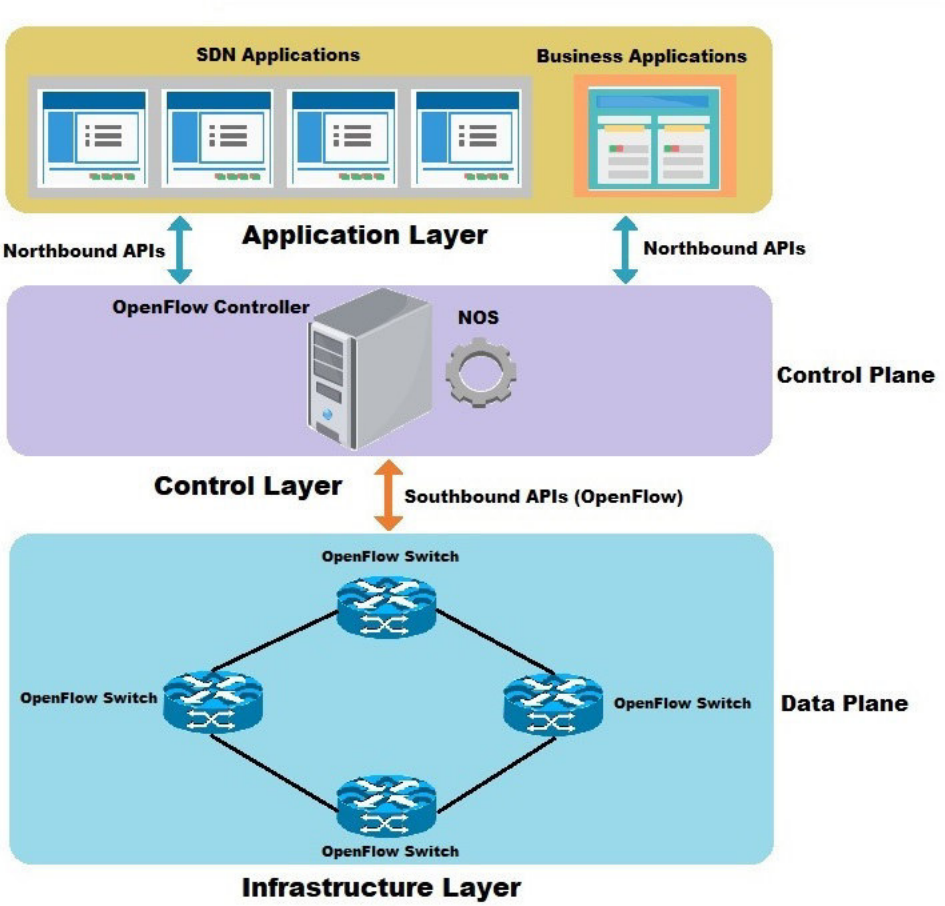


FIG. 1.1. *SDN Architecture*

TABLE 1.1
Difference between SDN and traditional network

Feature	SDN Network	Traditional Network
Centralization	Yes	No
Scalability	Yes	No
Heterogeneity	Yes	No
Programmability	Yes	No

in SDN environment. Researchers and network developers use the programmability feature to implement their new ideas and innovations [7, 8].

OpenFlow is a standard networking protocol used to communicate between control and data planes through southbound APIs in SDN. OpenFlow standard has various versions like 1.0.0 [9], 1.1.0 [10], 1.2.0 [11], 1.3.0 [12], 1.4.0 [13] and 1.5.0 [14]. OpenFlow-Configuration (OF-CONFIG) protocol is proposed by ONF [15] as a configuration protocol. The controller device which works with the OpenFlow protocol called OpenFlow controller. OpenFlow controller can add or remove entries in the flow table of the OpenFlow switch. The switch which works with OpenFlow protocol is called OpenFlow switch. OpenFlow switch contains two parts described in [16] as follows:

i. A flow table: Every OpenFlow switch has a flow table. Each entry of flow table is associated with an action decided by the controller.

ii. A secure channel: It is a communication channel that connects between OpenFlow switch and controller. It allows commands and packets pass through it using OpenFlow protocol.

The communication between control plane device (controller) and data plane devices (switches) must be secured as many commands and packets are sent between them. The importance of security in this case is to ensure the integrity of these commends and packets from being modified by third party. These commands and packets contain sensitive rules and policies of the network generated by the controller to be placed in switch's flow table entries. If these commands and packets manipulated, many changes may occur to the network and may the controller will not have the full control on switches. Hackers may also change these commands and packets to gain access to the controller. In this case, the hackers can control the whole network and disable some services. In the rest of this paper, we provide the related work in Section 2. The background and motivation are explained in Section 3. Section 4 talks about the proposed scheme along with its phases. We discussed the result of the proposed scheme in Section 5. Then we conclude this paper in the last section.

2. Related Work. Security between data and control planes in SDN is one of the major aspects that researchers and network administrators give much effort to ensure its efficiency. This effort aims to make robust authentication between network users in data plane and controller device in the control plane in order to manage the performance of the network and allow different applications take place in SDN environment securely. Many of researches have been conducted regarding this issue and they are as follow:

Osamah et al. [17] investigated the security threats occur on the SDN control channel by the Denial of service attack (DoS), then they evaluated the impact DoS attack by simulating a DoS attack against the controller device. They used the controller benchmarking (cbench) [18] as a tool to calculate and evaluate the performance of the SDN controller during normal operation (NO). After investigation and evaluation, they proposed a lightweight information hiding mechanism to avoid threats on SDN control channel. This authentication mechanism obscures the sensitive information from being revealed using information hiding algorithm, unlike the cryptography mechanisms which mingle the sensitive data. Finally, they suggested various recommendation to enhance the security of the proposed mechanism.

Mengmeng et al.[19] aimed to develop a security solution technique to guarantee the integrity of SDN flow rules. The authors analyzed the previous techniques which protect and control the SDN flow rules and then designed a PERM-GUARD. PERM-GUARD is a scheme for managing permissions and authenticating the flow rules in the SDN network. This scheme adds some components and functions to SDN architecture as follows:

i. Identity-based signature module
ii. Security center
iii. Application zone
iv. Manage the generation process of flow rules permissions for the valid applications.
v. Verify the flow rules validity.
vi. Authenticate the identities of the applications which generate flow rules insertion requests.
vii. Monitor the anomalous behaviors in SDN network conducted by malicious applications.

Then authors analyze, simulate and evaluate the proposed scheme by extending the Floodlight controller source code to support the functionality of PERM-GUARD.

Yustus et al. [20] proposed Northbound API security scheme to ensure the secure implementation of the REST NBI in SD controller, the scheme is designed to support the authentication and authorization and to capable to respond two questions:

i. Who are you?
ii. Do you have permission to access this network?

This scheme is using a token-based authentication and authorization process. The scheme was designed based on the OAuth 2.0 protocol [21].

Diogo et al. [22] proposed AuthFlow mechanism to guarantee the security between the hosts and servers in the SDN environment. This mechanism uses layer 2 protocols for authentication procedure. AuthFlow uses RADIUS authentication server which verifies the authenticity between the hosts and servers. The messages which are being sent between the RADIUS server and hosts are encapsulated by Extensible Authentication Protocol (EAP). To translate these messages from IEEE 802.1X to RADIUS packets is the responsibility of the authenticator server.

Hamza et al. [23] demonstrated the SDN OpenFlow controller security issues. They associated every security issue with existing countermeasures along with the description of these countermeasures. This demonstration concentrated to cover the security threats which may make the network service unavailable like DoS attack and some other attacks like (host hijacking attack, tampering attack, spoofing attack and man-in-the-middle attack). This research helps for further research to enhance the security of SDN OpenFlow controller.

Hong et al. [24] proposed host location hijacking attack and link fabrication attack which are attacking the OpenFlow network topology. The Floodlight controller is used to validate these two attacks. Mininet 2.0 simulation [25] is also used to demonstrate the topology of SDN network. These two attacks successfully poisoned the SDN topology. Therefore, the authors proposed TopoGuard security extension for SDN network topology using Floodlight controller. This extension protects the network from network poisoning attack.

3. Background and Motivation. To solve the issues of controller security, we used Zero- knowledge proof (ZKP). It is also known as Zero-Knowledge Protocol. It is a method occurs between two parties A and B where A is (the prover) tries to proof to B (the verifier) that he knows the secret X without exposing any other information about the secret or the secret itself. The proof procedure in ZKP is just knowledge about Knowledge, i.e. A will convince B that he knows the secret, B will be entirely confident that secret is true but will not learn anything as a result about this procedure, in this case, B gets zero-knowledge. Goldwasser, Micali and Rackoff introduced ZKP in [26]. To state that, the proving procedure is a ZKP, it must satisfy three properties:

i. Completeness: If the secret is true, the verifier will be convinced by the prover. Both prover and verifier must be honest and follow the protocol.
ii. Soundness: If the secret is false, the fake rover must not convince the honest verifier that, the secret is right.
iii. Zero-knowledge: If the secret is true, the verifier will learn nothing other than that, the secret is true.

The steps of ZKP occurs as follows: We assume that, there is A prover and B verifier, A wants to prove to B that he knows the secret$x$where $y = g^x$ without informing B what exact x is.

i. A generates a random number v and calculates $k = g^v$ and sends k to B.
ii. B generates a random number c as a challenge and sends it to A.
iii. A calculates $r = v - cx$ and sends it to B.
iv. B verifies $t = g^r * y^c$ if they are equal, then A is verified.

ZKP is the proof which revealed no information apart of the validity of the secret which the verifier demands the prover to proof. ZKP guarantees a high-level security protocol regardless of what verifier does, he will not learn any other information about the secret of the prover.

4. Proposed Scheme. The proposed scheme main concern is to secure the controller device in the SDN network from being accessed by malicious users. These malicious users try to control the whole SDN network, disable some services or shut the controller down by sending malicious requests to the controller. To protect the controller from this kind of malicious users, we preferred to use an identification scheme, so the user has to identify himself to the controller but not as a traditional way by sending his password in every attempt of login. The way we prefer is to let the user convince the controller that he knows the password without revealing the password itself or any partial information about it and prove that he is the real user. We utilized the ZKP identification scheme proposed by Feige, Fait and Shamir in [27] to improve the security of SDN controller against fake users. In this scheme, the user does not need to send his password in every login attempt where he has to prove that he knows the secret which controller has without revealing the actual secret or any information related to it. Three significant elements used in this scheme:

i. The controller: the verifier.
ii. The user: the prover.
iii. The Authentication Server: the trusted center which generates modulus n as a secret and has all user 's passwords and n numbers.

The user and controller have to agree on modulus n which is a product of two prime numbers, but no one will know the factorization of it. The module n will be assigned to both the user and the controller by the trusted center (the authentication server). After assigning the module $n$to both parties, the user has to prove to the controller that he knows the password without exposing the password itself. Thus both parties have

to follow some steps to satisfy the ZKP. We divided these steps into four phases, User key generation phase, Registration, Login and Authentication phases.

4.1. User key-generation phase. In this phase, the user has to generate public and private keys as follows:

i. The user generates S random numbers S_1,.....,S_k in Z_n.
ii. The user generates I_j randomly and independently as $1/S_j^2$ (mod n).
iii. The user publishes $I = I_1$,.....,I_k as public keys and keeps $S = S_1$,.....,$S_k$as private keys.

4.2. Registration Phase. In this phase, the user has to create his credentials and register himself in SDN network to connect to the controller and get the network policies and rules from it, see Figure 4.1. Therefore, the user has to perform the following steps:

i. The user creates a user *ID (IDu)*, user password *(PW)* and public key I send them along with user IP address *(IPu)* to the controller as a registration request,$ID_u//$h(PW)$//IP_u//I_j$where h is a hash function to encrypt the password.
ii. The controller will forward the registration request to the authentication server to record user credential in the database.
iii. Authentication server stores the user credentials in its database and generates the modulus n (a product of two prime numbers) ($n = p * q$) where n is considered as a secret and assign it to both user and controller. Now, the user is registered in SDN network and has to login into it.

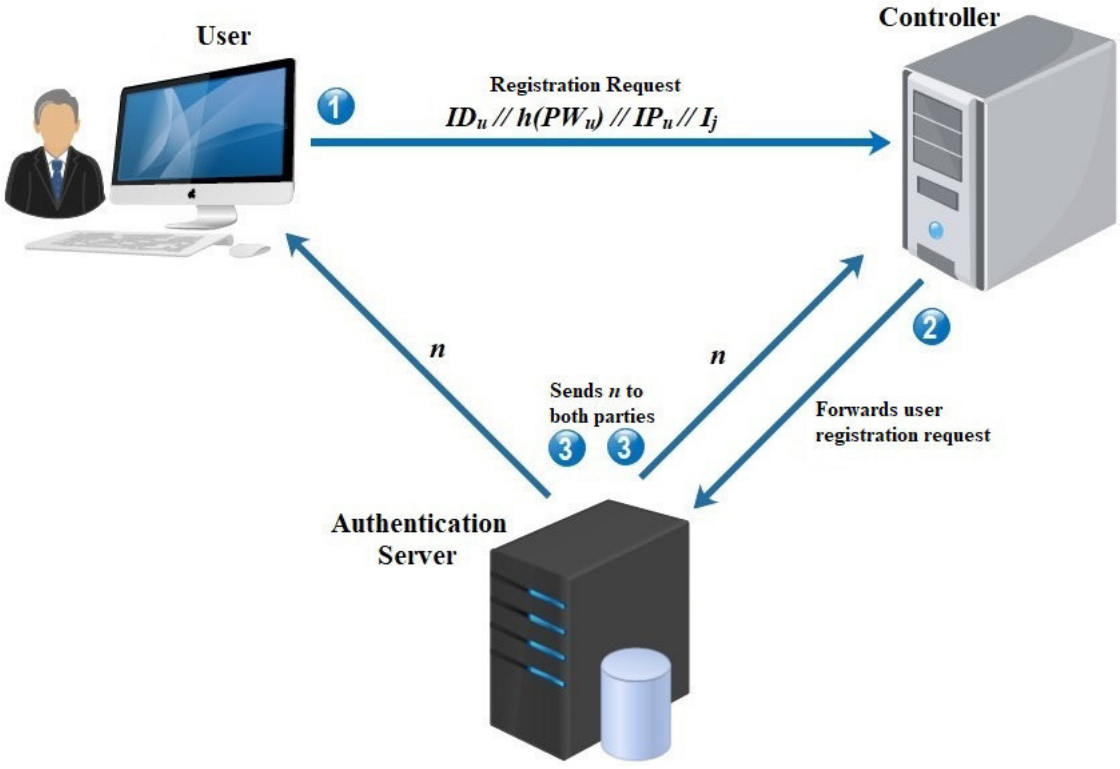


FIG. 4.1. *Registration Phase*

4.3. Login phase. In this phase, the user will try to login into SDN network by sending the controller a login request, see Figure 4.2. The user will not send the actual password instead he will use ZKP to prove his identity, and this occurs as the following steps:

1. The user chooses a random number$R$ and calculates $X = \pm R^2$ *(mod n)* then sends X to the controller.

2. The controller selects random numbers as Boolean vector $A_j = (A_1,.....,A_k)$ and sends them to the user as challenges.
3. The user calculates the value of $Y = R\prod A_{j=1}S_j \, (mod\ n)$ and sends Y to the controller.

Step 3 should be repeated in t iterations depends on the number of challenges in boolean vector $(A_1,.....,A_k)$. Suppose the controller sent 3 challenges in the boolean vector as $(A_1, A_2, A_3$ then the number of iteration in step 3 will be t=3. So the user should calculate this step 3 times as:

i. $Y = R\prod A_{1=1}S_1 \ (mod\ n)$
ii. $Y = R\prod A_{2=1}S_2 \ (mod\ n)$
iii. $Y = R\prod A_{3=1}S_3 \ (mod\ n)$

where S_1, S_2, S_3 will be 3 private keys of the user.

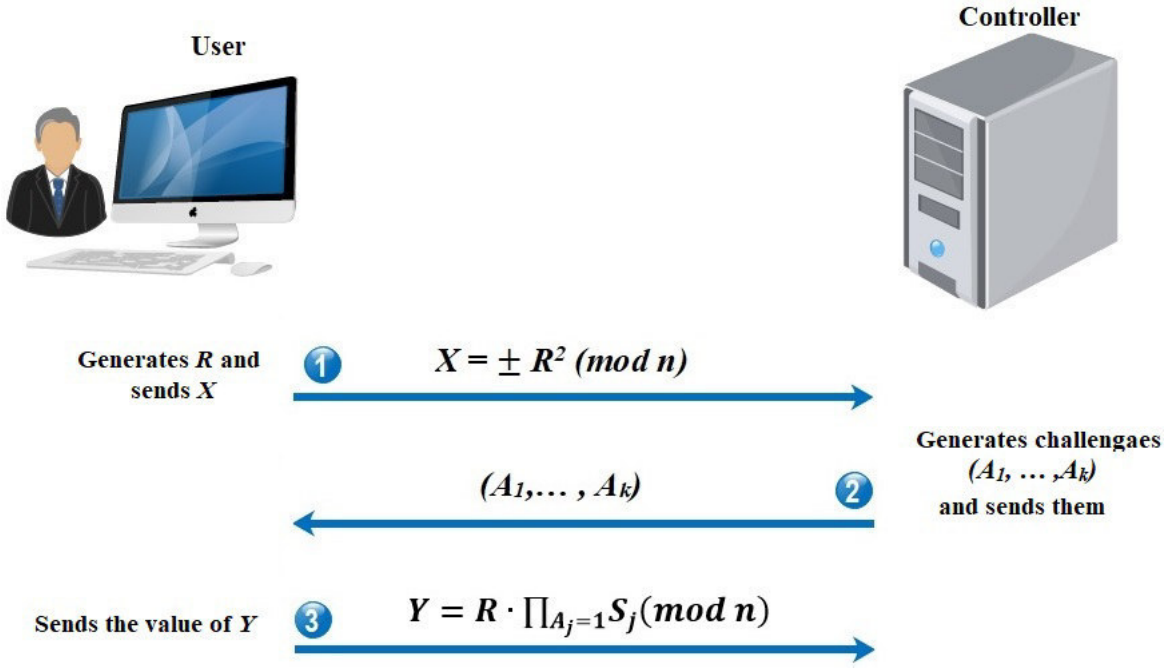


FIG. 4.2. *Login Phase*

Now, the user is waiting to the controller to accept his proof of identity to start the communication and exchange packets between them.

4.4. Identification phase. In this phase, the controller identifies the user proof of identity to allow him to send and receive the packets, see Figure 4.3. The identification step occurs as follow:

i. $X = \pm Y^2 \prod A_{j=1}I_j \ (mod\ n)$ The identification step must be repeated in t iterations to accept the user proof of identity. The identification iterations are decided by the number of challenges in boolean vector $(A_1,.....,A_k)$ sent by the controller as we assumed that in login phase, the numbers of challenges are 3, then the number of the verification iterations t will be $t = 3$ and the calculation will be as follow:
ii. $X = \pm Y^2 \prod A_{1=1}I_1 \ (mod\ n)$
iii. $X = \pm Y^2 \prod A_{2=1}I_2 \ (mod\ n)$
iv. $X = \pm Y^2 \prod A_{3=1}I_3 \ (mod\ n)$

where $(I_1, I_2, I_3$) will be 3 public keys of the user.

The controller accepts the user proof of identity if and only if the identification step is performed successfully in all t iterations otherwise the user proof of identity will be rejected. If the controller accepts the user's request, then the policies will be installed as entries into the flow table of the switch. Therefore the user will be able to communicate with network resources and other devices in SDN network. If the controller rejects the user's request, the user has to repeat the operation of login phase again with correct evidence.

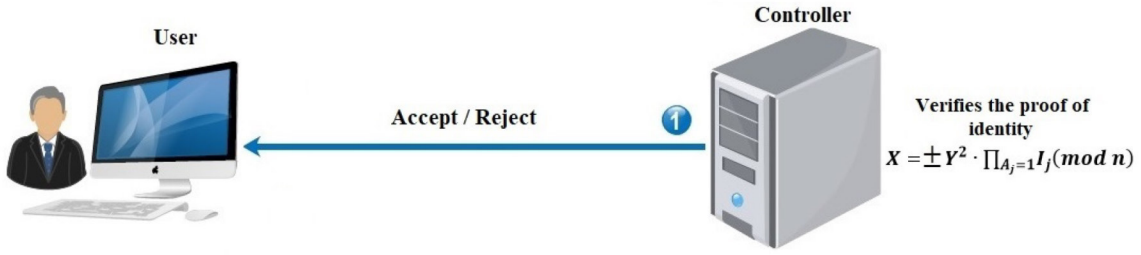


FIG. 4.3. *Identification phase*

5. Result Discussion. In this section, the following parameters are used to evaluate the efficiency of the proposed scheme.

5.1. Computation Cost, Communication Cost and Storage Overhead Analysis. The primary goal of the proposed scheme is to reduce the overall computation cost, communication cost and storage overhead associated with the prover (user) and verifier (controller). To make the proposed scheme in practical use, the Table 5.2 along with notations Table 5.1, show that this scheme is an efficient scheme in term of computation cost communication cost and storage overhead for SDN environment

TABLE 5.1
Notations of Table 5.2

Symbol	Description
e	Exponent
m	Multiplication
mod	Modulus

TABLE 5.2
Computation cost, communication cost and storage overhead of login and identification phases

Phase	Computation Cost	Communication Cost/bit	Storage Overhead/bit
Login phase	$1e + 2m + 2mod$	2689	2560
Identification phase	$3e + 2m + 1mod$		

Many aspects must be taken into consideration to identify the efficiency of any identification scheme such as computation cost, communication cost and storage overhead. We mainly concentrate on the login and identification phases. These two phases are considered the major parts of the identification mechanism in this proposed scheme, and they are frequently executed as compared to other phases. Without loss of generality, the parameters *(R,I,S,A)* are considered to be 128-bit while the n is 1024-bit long. The computation cost is calculated according to how many mathematical operations occurred between user and controller sides in both login and identification phases. The communication cost focuses on the number of the bits are being sent between the user and controller in both login and identification phases. The calculation of the storage overhead occurs according to how many bits are the user and controller store in their devices. The parameters*(R,I,S,n)* are stored in the user side device, where*(A,n)* are stored in the controller side. The calculation of computational cost, communication cost and storage overhead in Table 5.2 is considered for one-time identification iteration.

5.2. Security Analysis. The proposed scheme is efficient to avoid various types of attacks as host impersonate attack, DoS attack and Man-in-the-middle attack due to its robust security mechanism of identifying the real users and invisibility of user credentials as well.

5.2.1. Host Impersonation Attack and Man-in-the-middle Attack. The proposed scheme can prevent the host impersonation attack as well as the man-in-the-middle attack by allowing the user to send his credential $ID_u//\text{h(PW)}//IP_u//I_j$ to the controller only once and that is in registration face. In the login phase,

the user will send $X = \pm R^2 \ (mod\ n)$ to the controller where$X$ changes in every attempt of login according to the secret n sent by the authentication server to both user and controller. In this case, the attacker cannot hack a consistent$X$ to impersonate the host to communicate the controller for subversive reasons.

5.2.2. DoS Attack. In the case of a DoS attack, the hacker will try to gain control over the controller to suspend some service or make the whole network unavailable. The proposed scheme prevents the issues of DoS attack by instructing the controller to verify whether the user knows the secret or not. If the user proofs that he knows the secret multiple times (which is a guarantee that the user is not a hacker) as mentioned in identification phase as $X = \pm Y^2 \prod (A_{j=1}) I_j \ (mod\ n)$, then the user is able to communicate the controller.

6. Conclusion. The security enhancement of SDN has been taken seriously by researchers and enterprise networks in recent years especially the security between control and data planes. The users in the data plane have to prove their identity to the controller to gain access to it and get networking policies. In this paper, we presented the security features of the ZKP protocol and how the user identity proceed and proved. Then we proposed an efficient ZKP based identification scheme for securing SDN. This scheme guarantees that only real users can connect to the SDN controller. We also discussed the computation cost communication cost and storage overhead analysis to validate the efficiency of our proposed scheme along with security analysis.

REFERENCES

[1] M. Casado, T. Koponen, D. Moon, S. Shenker, *Rethinking Packet Forwarding Hardware,*In Proc. OfHotNets, 2008.
[2] ONF,*Software-Defined Networking: The New Norm forNetworks,*white paper, https://www.opennetworking.org.
[3] Xia, Wenfeng, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie, *A survey on software-defined networking.,*IEEE Communications Surveys & Tutorials 17.1 (2015): 27-51.
[4] Zhang, Yuan, Lin Cui, Wei Wang, and Yuxiang Zhang, *A survey on software defined networking with multiple controllers,* Journal of Network and Computer Applications (2017).
[5] Kreutz, Diego, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig, *"Software-defined networking: A comprehensive survey"* , Proceedings of the IEEE 103.1 (2015): 14-76.
[6] Bozakov, Zdravko, and Amr Rizk, *"Taming SDN controllers in heterogeneous hardware environments."* Software Defined Networks (EWSDN), 2013 Second European Workshop on. IEEE, 2013.
[7] Feamster, Nick, Jennifer Rexford, and Ellen Zegura. *"The road to SDN: an intellectual history of programmable networks."*ACM SIGCOMM Computer Communication Review 44.2 (2014): 87-98.
[8] Nunes, Bruno Astuto A., Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti., *"A survey of software-defined networking: Past, present, and future of programmable networks"* IEEE Communications Surveys & Tutorials 16.3 (2014): 1617-1634.
[9] OpenFlow Specification 1.0 http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf., (Accessed: 2017/04/05) (2009). pp. 9196.
[10] OpenFlow Specification 1.1.0., http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf., (Accessed: 2017/04/05) (2011).
[11] OpenFlowSpecification1.3.0., https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.0.pdf. (Accessed: 2017/04/05) (2012).
[12] OpenFlow Specification 1.3.0., https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.0.pdf.,(Accessed: 2017/04/05) (2012).
[13] OpenFlow Specification 1.4.0., https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf, (Accessed: 2017/04/07) (2013).
[14] OpenFlow Specification 1.5.0., https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf, (Accessed:2017/04/07) (2014).
[15] OpenFlow-Configuration Protocol 1.2, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf, (Accessed:2017/04/07) (2014).pp. 101109.
[16] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., et al *"OpenFlow: enabling innovation in campus networks."* ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.
[17] Abdullaziz, Osamah Ibrahiem, Yu-Jia Chen, and Li-Chun Wang. *"Lightweight authentication mechanism for software defined network using information hiding."*Global Communications Conference (GLOBECOM), 2016 IEEE. IEEE, 2016.
[18] Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., & Sherwood, R. *"On Controller Performance in Software-Defined Networks."*Hot-ICE 12 (2012): 1-6.
[19] Wang, M., Liu, J., Chen, J., Liu, X., & Mao*"Perm-guard: Authenticating the validity of flow rules in software defined networking."* Hot-ICE 12 (2012): 1-6.
[20] Oktian, Y. E., Lee, S., Lee, H., & Lam, J*"Secure your northbound SDN API."*Hot-ICE 12 (2012): 1-6.
[21] Jones, Michael, and Dick Hardt *The oauth 2.0 authorization framework: Bearer token usage.* No. RFC 6750. 2012.

[22] Mattos, Diogo Menezes Ferrazani, and Otto Carlos Muniz Bandeira Duarte. *"AuthFlow: authentication and access control mechanism for software defined networking."*Annals of Telecommunications 71.11-12 (2016): 607-615.
[23] Mutaher, Hamza, Pradeep Kumar, and Abdul Wahid. *"OPENFLOW CONTROLLER-BASED SDN: SECURITY ISSUES AND COUNTERMEASURES."* International Journal of Advanced Research in Computer Science 9.1 (2018).
[24] Hong, S., Xu, L., Wang, H., & Gu, G *"Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures."* NDSS. Vol. 15. 2015.
[25] Mininet. Rapid prototyping for software defined networks.http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/.
[26] Goldwasser, Shafi, Silvio Micali, and Charles Rackoff. *"The knowledge complexity of interactive proof systems."* SIAM Journal on computing 18.1 (1989): 186-208.
[27] Feige, Uriel, Amos Fiat, and Adi Shamir *"Zero-knowledge proofs of identity."* Journal of cryptology 1.2 (1988): 77-94.

Edited by: Khaleel Ahmad
Received: Nov 24, 2018
Accepted: Feb 11, 2019

AIMS AND SCOPE

The area of scalable computing has matured and reached a point where new issues and trends require a professional forum. SCPE will provide this avenue by publishing original refereed papers that address the present as well as the future of parallel and distributed computing. The journal will focus on algorithm development, implementation and execution on real-world parallel architectures, and application of parallel and distributed computing to the solution of real-life problems. Of particular interest are:

Expressiveness:

- high level languages,
- object oriented techniques,
- compiler technology for parallel computing,
- implementation techniques and their efficiency.

System engineering:

- programming environments,
- debugging tools,
- software libraries.

Performance:

- performance measurement: metrics, evaluation, visualization,
- performance improvement: resource allocation and scheduling, I/O, network throughput.

Applications:

- database,
- control systems,
- embedded systems,
- fault tolerance,
- industrial and business,
- real-time,
- scientific computing,
- visualization.

Future:

- limitations of current approaches,
- engineering trends and their consequences,
- novel parallel architectures.

Taking into account the extremely rapid pace of changes in the field SCPE is committed to fast turnaround of papers and a short publication time of accepted papers.

INSTRUCTIONS FOR CONTRIBUTORS

Proposals of Special Issues should be submitted to the editor-in-chief.

The language of the journal is English. SCPE publishes three categories of papers: overview papers, research papers and short communications. Electronic submissions are preferred. Overview papers and short communications should be submitted to the editor-in-chief. Research papers should be submitted to the editor whose research interests match the subject of the paper most closely. The list of editors' research interests can be found at the journal WWW site (`http://www.scpe.org`). Each paper appropriate to the journal will be refereed by a minimum of two referees.

There is no a priori limit on the length of overview papers. Research papers should be limited to approximately 20 pages, while short communications should not exceed 5 pages. A 50–100 word abstract should be included.

Upon acceptance the authors will be asked to transfer copyright of the article to the publisher. The authors will be required to prepare the text in LaTeX 2ε using the journal document class file (based on the SIAM's `siamltex.clo` document class, available at the journal WWW site). Figures must be prepared in encapsulated PostScript and appropriately incorporated into the text. The bibliography should be formatted using the SIAM convention. Detailed instructions for the Authors are available on the SCPE WWW site at `http://www.scpe.org`.

Contributions are accepted for review on the understanding that the same work has not been published and that it is not being considered for publication elsewhere. Technical reports can be submitted. Substantially revised versions of papers published in not easily accessible conference proceedings can also be submitted. The editor-in-chief should be notified at the time of submission and the author is responsible for obtaining the necessary copyright releases for all copyrighted material.